Algorithms for Discovering Collections of High-Quality and Diverse Solutions, With Applications to Bayesian Non-Negative Matrix Factorization and Reinforcement Learning

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

Citation

Citable link
http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029756

Terms of Use
This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA
Algorithms for discovering collections of high-quality and diverse solutions, with applications to Bayesian non-negative matrix factorization and reinforcement learning

A DISSERTATION PRESENTED
by
MUHAMMAD ARJUMAND MASOOD
TO
The John A. Paulson School of Engineering and Applied Sciences

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy
IN THE SUBJECT OF
Applied Mathematics

Harvard University
Cambridge, Massachusetts
April 2019
To Ambo Khala,
For all the teachings, inspiration & love
Algorithms for discovering collections of high-quality and diverse solutions, with applications to Bayesian non-negative matrix factorization and reinforcement learning

Abstract

Machine Learning problems often admit a solution space that is not unique. When multiple feasible solutions exist, picking from a diverse, representative set may lead to better generalization and task-specific performance. While the emphasis of much of the literature has been on directly finding the 'best' solution, we show that often a diverse set of near optimal solutions can be found which may be useful to practitioners and experts using machine learning models in decision making. This thesis investigates methods for obtaining a useful collection of solutions in specific models.

Non-negative Matrix Factorization (NMF) is a popular data exploration tool and its Bayesian formulation is a promising approach for understanding uncertainty within this structure. We demonstrate that current approaches are lacking in the proper characterization of uncertainties and present novel techniques to provide model flexibility and improve the quality and speed of the inference. These techniques are applied to standard benchmark datasets for NMF as well as a curated medical dataset for understanding comorbidities in the Autism Spectrum Disorder (ASD). We show how a distinct collection of NMFs of nearly equal quality give rise to variability in interpretation of features and subsequent predictions.

Finally, we present extensions of our diverse collection-based approach to the on-policy and off-policy Reinforcement Learning setting. Here, a completely new set of technical tools is required. In both on-policy and off-policy variants, we use diversity as a regularization feature in order to obtain a set of high-quality diverse policies. In addition to finding diverse policies in simulate-able multi-goal domains, we find a diverse set of policies designed to aid clinical decision making using ICU data for sepsis and hypotension management.
4.7 Related Work .......................................................... 53
4.8 Conclusion ........................................................... 55

5 Multiple Solutions in NMF: A Particle-Based Approach with Transfer Learning .................................................. 58
  5.1 Problem Description .................................................. 58
  5.2 Inference Setting ...................................................... 61
  5.3 Background .......................................................... 62
  5.4 Approach ............................................................. 64
  5.5 BNMF Models ......................................................... 70
  5.6 Experimental Setup ................................................. 74
  5.7 Results ............................................................... 79
  5.8 Discussion: When is $Q$-Transform successful? .......... 87
  5.9 Related Work ......................................................... 91
  5.10 Conclusion .......................................................... 92

6 Multiple Solutions in RL: Diversity Inducing Policy Gradient for On-Policy RL ........................................ 94
  6.1 Problem Description ................................................ 94
  6.2 Background ........................................................ 95
  6.3 Diversity-Inducing Policy Gradient (DIPG) ................. 97
  6.4 Experimental Setup ................................................. 101
  6.5 Results ............................................................... 103
  6.6 Related Work ......................................................... 106
  6.7 Discussion .......................................................... 107
  6.8 Conclusion .......................................................... 107

7 Multiple Solutions in RL: Diverse, Safe and (High) Quality Policies for Off-Policy RL .................................. 109
  7.1 Problem Description ................................................ 109
  7.2 Background ........................................................ 111
  7.3 Related Work ......................................................... 113
  7.4 Cohort ............................................................... 114
  7.5 Methods ............................................................. 116
  7.6 Experimental Setup and Results ............................... 119
# Listing of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison of max angle and $\ell_1$-min match measures when $|A_1 - A_2|_F = |A_1 - A_3|_F$.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison of minimum covering numbers of samples ($\epsilon = 1$).</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of minimum covering numbers of samples ($\epsilon = 2.5$).</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Comparison of persistent minimum covering numbers ($C_\epsilon$ as a function of $\epsilon$) of samples.</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>The dotted triangles show two rank-3 factorizations of data whose convex hull is shaded in grey.</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>A toy illustration of the extreme rays. The dotted green and red lines are the extreme rays and define the cone where all vectors $q_A$ lie such that $Aq_A \geq 0$.</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>ERMH consistently outperforms the baselines in angular distance. Samples in the ERMH chains are significantly different in the angle sense. The difference seems to be maintained as $N$ increases from 500 to 5000.</td>
<td>31</td>
</tr>
<tr>
<td>3.4</td>
<td>ERMH consistently outperforms the baselines on integrated autocorrelation time. In fact, we notice that as the number of observed data points increases, the Gibbs sampler and RMH tend to get stuck even more in the same optima therefore increasing their integrated autocorrelation time measurements. ERMH remains constant across $N$.</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>Per-iteration run time for the three approaches on the example from Laurberg et al. (2008); our making the proposals for our ERMH approach requires more time than just Gibbs sampling but less than a proposal that instantiates all the elements of $A$ and $W$ from the prior (RMH).</td>
<td>32</td>
</tr>
<tr>
<td>3.6</td>
<td>We show the clusterings obtained from two distinct factorizations from our chain. The basis matrices are separated by 35 degree angle. We notice that the basic rank 2 classification stays the same between the two factorizations, but we get a different sub-clustering.</td>
<td>33</td>
</tr>
<tr>
<td>3.7</td>
<td>Different top-ten lists on four topics from ERMH chain. The four topics appear to be politics, technology, sports and economy.</td>
<td>33</td>
</tr>
</tbody>
</table>
3.8 Comparing samples in our chains to a fixed randomly generated matrix to see how angles change across the chains. It appears that RMH and Gibbs have both found (slightly different) modes where they remain.

3.9 We show two distinct factorizations from our chain. These are separated by 19 degree angle and have been permuted so that they align according to minimum average angle. The image on the top-left is the one that corresponds to the maximum angle.

3.10 It appears that the Gibbs sampler entry is being perturbed around one place whereas the ERMA entry is showing clear upward and downward movements. These movements would show up as high autocorrelations.

4.1 A 2-D projection of Laurberg Data and NVI components. The different components are not distinguishable.

4.2 A 2-D projection of Laurberg Data and the components accepted in the HMC + ONVI algorithm. These components are all coming from the same mode, which overlaps with the analytical solution (initialization).

4.3 A 2-D projection of Laurberg Data and the components accepted in the RRT + ONVI algorithm. These components are distinct from the analytical solution (initialization). These diverse components continue to explain the data under the Uniform noise model as the data lies in the convex hull of these RRT-based factorizations.

4.4 Laurberg Data: A persistence plot of samples from the variational distribution obtained from the different algorithms. The RRT + ONVI is the only algorithm for which minimum cover numbers are larger than one for angles greater than 0.01. The other algorithms are unable to escape the modes that they were initialized in.

4.5 20-Newsgroups data: Persistence plot of the components shows that RRT + ONVI components are more spread out than baselines.

4.6 BBC data: Persistence plot of the components shows that RRT + ONVI components are more spread out than baselines.

4.7 Reuters data: Persistence plot of the components shows that RRT + ONVI components are more spread out than baselines.

4.8 AML/ALL data: The RRT + ONVI components are more spread out than baselines but it is a much smaller spread in terms of angle. For angles greater than 0.1 the minimum covering number is one for all algorithms.

4.9 Faces data: The RRT + ONVI components are spread out more than baselines.

4.10 Hubble data: The RRT + ONVI components show the largest posterior coverage.
5.1 A schematic of the transfer learning procedure for NMF: A small dataset is used to learn transformation matrices $Q_A, Q_W$. We then apply these transformation matrices to multiple larger datasets (with any number of dimensions or observations) using its SVD to obtain a transfer-based initialization.

5.2 A comparison of SILF loss and commonly used $l_1, l_2$ loss functions. The SILF insensitivity parameter $\epsilon$ is set to 0.5, and the smooth transition factor $\beta$ is varied. Small values of $\beta$ lead to sharp transition in the SILF loss profile whereas the transition is less abrupt for large values of $\beta$. In contrast, other popular loss functions such as $l_1$ or $l_2$ do not have insensitive regions, and in the case of NMF, treat the objective function as the sole guide for factorization quality.

5.3 For each dataset we show the quality of the BNMF approximate posterior ($M = 5$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to the exponential-Gaussian BNMF (lowest Stein discrepancy) are produced either using the $Q$-Transform initializations (in red) for the Gibbs sampler (in yellow).

5.4 For each dataset we show the quality of the BNMF approximate posterior ($M = 5$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).

5.5 The top 15 words for topic A (computers/electronics) and topic B (space) shows that different factorizations provide an emphasis on different terms. In topic A, the top word from factorization 1 and 2 is ‘card’, but it does not appear in the top 15 words of factorizations 3. Instead a similar term ‘chip’ is emphasized in Factorization 3. In topic B, the terms ‘space’ and ‘nasa’ appear in all three factorizations but factorization 2 is the only one with digital terms like ‘ftp’, ‘server’, ‘site’ and ‘faq’. In contrast factorization 1 and 3 both contain more physical terms like ‘sun’, ‘moon’, ‘launch’.

5.6 Classifiers trained on feature vectors from different factorizations yield variability in prediction performance (as measured by AUC). The dotted lines show the factorization that produces the best performing classifier for each category. The factorization (blue dotted line) that predicts the “Talk” category best is actually one of the worst performing factorizations for the “Science” category. This variability in performance demonstrates that no single factorization gives the best latent representation for the overall prediction task.
5.7 We explore top words in the topics relating to key terms of interest to clinicians and discover that different NMFs place varying amount of emphasis on different terms. Such variability is of interest to clinicians who may be trying to interpret topics to understand patterns in ASD.

5.8 Classifiers trained on weights matrix of $M_{max} = 100$ different factorizations to predict the presence of certain medical codes in a patient’s trajectory exhibit significant variability in prediction on a test set (as measured by AUC). Different factorizations lead to top predictors for the onset of different medical issues.

5.9 Under different percentages of missingness in the Olivetti Faces dataset (10%, 30%, 50%), the quality of the BNMF approximate posterior and the corresponding runtime of $Q$-Transform and the other baselines is shown. The best discrete posterior approximations to BNMF are produced using the $Q$-Transform initializations (in red).

5.10 Sample factorizations from the variational posterior using $Q$-Transforms show that a diverse range of basis elements can be use to approximate the data. However, HMC samples seem to be identical indicating that HMC was only exploring a very small region of the posterior space.

5.11 For different synthetic data $X_S$ generating procedures, we show the initialization quality obtained via the $Q$-Transform matrices on a target data $X_T$. Dirichlet, Uniform, Gaussian have significantly superior performance compared to Sub-sample, Column-projection and Uniform data. For comparison, we show the quality of NMF solutions (solid line) and random initializations (dashed line).

5.12 Using Gaussian factors for the synthetic data generation process with different ranks does not appear to change the quality of the $Q$-Transform initialization quality on the target data $X_T$. This indicates that this generating procedure is not sensitive to the rank in order to produce high quality (close to true NMF solution) initializations using $Q$-Transform. For comparison, we show the quality of NMF solutions (solid line) and random initializations (dashed line).

5.13 In the low-noise regime, the reconstruction error of $Q$-Transform initializations is significantly less than random restart initializations. This relative advantage gets smaller as the noise level increases. Similarly, the time taken to converge is significantly shorter than the random restart approach under the low noise scenario and continues to increase with noise. As expected, at high noise levels there exists no additional advantage to the $Q$-Transform approach (the optimization time ratio approaches 1).
5.14 On a synthetic target dataset \((D = N = 500, K_{\text{NMF}} = 10)\), we apply \(Q\)-Transform initializations using varying transfer ranks and SVD ranks \(R_T = R_{\text{SVD}} = \{1, 2, \ldots, 10\}\). We see that for a range of low rank values, the \(Q\)-Transform initializations are high quality, but at larger values the quality of initializations gets worse. The dotted line shows the quality of random initializations and the solid line shows the quality NMF solutions. The reconstruction errors are normalized by the norm of the data.

6.1 A comparison of trajectories in the 2-D navigation tasks shows that DIPG-PPO (with \(N = 4\) for Multi-Goal and \(N = 2\) for Asymmetric Goals) produces near-optimal trajectories to reach each goal. Soft-Q takes sub-optimal paths to the goal regions and SVPG generally fails to solve the environments.

6.2 A comparison of the kernel-based similarity between trajectories and the average return across different algorithms reveals that DIPG-PPO and RR-PPO consistently give the largest returns while achieving low similarity. The colors are consistent with those in figure 6.1.

6.3 We show the probability of taking actions throughout the duration of a single ICU stay. While these agents all have roughly equal value, they exhibit different emphasis on treatment styles as can be seen by the variability in action probabilities.

7.1 Reward function used in our analysis. The reward depends primarily on MAP, with inflection points at 55, 60, and 65. Values 65 and above are considered optimal.

7.2 Two synthetic patient trajectories along with the recommended actions by the two agents in one run of our algorithm (DSQRL). In the left pane the two agents suggest different treatment strategies that ultimately yield the same return of 0.60 for one type of patient (with disease A), while in the right pane the two agents agree on an identical unique optimal treatment for this type of patient (with disease B).

7.3 Overall action probabilities. Each column corresponds to one of the nine actions in our action space. The top row shows the physician behavior probabilities aggregated across all patients in the test set. The bottom four rows show probabilities from the four different agents from one run of our algorithm.
B.1 SVGD escape

A collection of 100 particles is unable to escape the central mode. A large collection has a higher repulsive force and is able to escape into the two other modes.

B.3 An empirical verification of the repulsive force for particles localized in a single mode. We see an increase in the maximum distance between SVGD particles is observed as the total size of the approximating set increases.

B.4 Using weights on subsets of a 200 particle selection allows us to use fewer particles to achieve smaller Stein discrepancy

C.1 The joint likelihood of factorizations shows that SVGD and DSGD generally produce the worst quality factorizations. The remaining algorithms produce higher quality factorizations.

C.2 The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that meet the threshold quality (black line) for useful factorizations. The other approaches consistently produce factorizations that meet this minimum quality requirement.

C.3 The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that result in low error on heldout data. The other approaches consistently produce factorizations which generalize better and give low error on heldout data.

C.4 The joint likelihood of factorizations shows that SVGD and DSGD generally produce the worst quality factorizations. HMC, NNDSVDar, Random and $Q$-Transform produce high quality factorizations.

C.5 The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that meet the threshold quality (black line) for useful factorizations. The other approaches consistently produce factorizations that meet this minimum quality requirement.

C.6 The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that result in low error on heldout data. The other approaches consistently produce factorizations which generalize better and give low error on heldout data. The matrix completion error on the missing data variant of the Olivetti faces dataset also gives similar results.
C.7 The kernel similarity indicates that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. $Q$-Transform and Random are the only algorithms that produce factorizations of high quality that are not similar.

C.8 The pairwise distance between basis matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar, $Q$-SVGD, DSGD and SVGD factorizations are also very similar. $Q$-Transform and Random produce basis matrices that are more distinct than the Gibbs sampler.

C.9 The pairwise distance between weights matrices shows that factorization collections obtained by DSGD, SVGD and $Q$-SVGD lead to diverse weights matrices. The remaining methods lead to similar levels of diversity within the factorization collections.

C.10 The kernel similarity indicates that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar.

C.11 The pairwise distance between basis matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. $Q$-Transform and Random are the only algorithms that produce factorizations of high quality that are different.

C.12 The pairwise distance between weights matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. $Q$-Transform and Random are the only algorithms that produce factorizations of high quality that are different.

C.13 Variability in the prediction task on the Autism dataset shows that SVGD factorizations yield poor prediction (that are slightly improved upon by $Q$-SVGD), HMC and NNDSVDar factorizations predictions are high but not diverse, and Gibbs and random restarts yields performance similar to $Q$-Transform.

C.14 Variability in the prediction task on the 20 Newsgroups dataset shows that SVGD factorizations yield poor prediction (that is drastically improved upon by $Q$-SVGD), HMC factorizations are not diverse, NNDSVDar, Gibbs and random restarts yields some variability in prediction that is similar to $Q$-Transform.
C.15 For each dataset we show the quality of the BNMF approximate posterior ($M = 25$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).

C.16 For each dataset we show the quality of the BNMF approximate posterior ($M = 50$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).

C.17 For each dataset we show the quality of the BNMF approximate posterior ($M = 25$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).

C.18 For each dataset we show the quality of the BNMF approximate posterior ($M = 50$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).

C.19 The runtime for calculating optimal weights (left: low-quality random parameters, right: “good” parameters obtained from NMF solvers) does not depend on the parameters; both have the same runtime costs and depend only on collection size.
Acknowledgments

I want to thank several important people without whom this thesis would truly not have been possible.

First and foremost, I want to thank Finale. Finale is an exceptional individual who inspires everyone not only for her intellect but also her humanity. Writing this thesis was an exercise in resilience and I have benefited greatly from Finale’s belief in me, encouragements and constant insights. As with all research projects, ideas fail and most times failed ideas are incredibly disheartening but Finale’s optimism and direction served as crucial drivers in those times. Finale, I truly cannot thank you enough for an amazing educational experience, professional growth opportunity and a very memorable four years.

Next, I want to thank my committee members David Parkes and Yaron Singer as well as other mentors and supporters at SEAS. David always knew what was going on with my projects – perhaps even better than me sometimes – and always provided an avenue for thoughtful discussions! Yaron, always had his door open for me to ask any questions about ideas that I was developing. Thank you also to Vahid Tarokh for looking out for me and always giving me sincere advice. I am also grateful to John Girash for supporting me through challenging times. Helping a graduate student with the generosity that these individuals showed with their time is something that I understand to be both rare and yet so crucial to this PhD. Another thanks to Eli Tziperman for making it all possible in the first place.

I would also like to extend my gratitude to my lab- DTAK. You are all individuals that have been so inspirational to me for the integrity of your research, the generosity of your spirit and the kindness that you all bring to our shared academic experience. No unicorn is as magical as all of you!

There are many other people I wish to thank: teachers, coaches, extended family members, professors and friends – you know who you are! In addition, I want to thank Lowell House for all the love and craziness, my yoga teachers for showing me a path to self-fulfillment and to my squash buddies for all the intense rallies and pure fun!

Finally, I want to thank my family - the people who have stood by me, made me, and in many ways made all this possible. Let me begin by thanking my Mama and Baba. I am so grateful for all the sacrifices you have made to give us the educational experience and the opportunities that we had. Your faith in my abilities and my decisions, your energy and effort into raising me to have the values
and skills that I have and your endless love have always spurred me on. I owe this all to you! A big thank you to my brother who was a confidante, a role model and a support system. My brother and sister-in-law are people I can turn to with all kinds of news and be assured of an open ear, a kind word and a shoulder to lean on. As you can imagine, that is crucial to a PhD. And of course a big thank you to my wife, Mahnum, who has helped name many of the proposed algorithms in this work and, well, written the first draft of these acknowledgements. I admire you for your work ethic and intellect, and am honored and privileged to be in a partnership that has given meaning to it all.
1 Introduction

1.1 Motivation

Machine Learning techniques are rapidly gaining popularity. Accompanying this rapid growth are new challenges for the research community. Amodei et al. (2016) identify misspecification of the objective function as one such challenge in the development of safe AI. In some cases, it is unclear how to quantify the true objective function of interest and in other cases the true objective might not be feasible to optimize so practitioners often choose a misspecified one instead to make the problem more suitable to apply learning algorithms. Roberts et al. (2016) point out that in a sense all objective functions we optimize over are misspecified since optimizing in-sample quality measures is often not the end goal. They argue that objective functions only serve as a guide to quality and cannot be used to distinguish between solutions of near-equal value in terms of the objective.

One way we propose to address issues related to misspecification of the objective is to design machine learning algorithms to find multiple solutions that are representative of solutions considered feasible under the objective and present them to practitioners so they can use a richer set of tools to evaluate and choose the solution best suited to their end goal.
1.2 Summary of Contributions

In this work, we explore techniques for identifying and discovering meaningful diversity in the solution space of a popular unsupervised learning model; Non-negative Matrix Factorization (NMF) as well as an extension to on-policy and off-policy Reinforcement learning (RL).

The work on NMF (Chapters 2, 3, 4, 5) starts with Chapter 2, where we present an empirical investigation that identifies the limitations of current evaluation methods for quantifying diversity, proposes alternate metrics and demonstrates how existing approaches fail to discover some valid regions of the NMF solution space. We use this as motivation to develop a faster-mixing Bayesian NMF (BNMF) sampler by interweaving a traditional Gibbs sampler with a novel geometric Metropolis-Hastings proposal (Chapter 3). In Chapter 4, we apply a non-parametric variational inference technique to BNMF in combination with a novel geometric search method for NMF solutions. Finally, in Chapter 5, we introduce an algorithm for particle-based inference along with a novel (more flexible) BNMF model. Unlike the work in Chapters 3 and 4 which directly leverages geometric structure of NMF, we introduce a technique for indirectly leveraging the geometry of NMF through transfer learning using synthetically generated data. The combination of our particle based inference algorithm, flexible modeling framework and this transfer learning technique leads to our discovery of meaningfully distinct factorizations in our BNMF posteriors on many benchmark NMF datasets as well as in a clinical application area for NMF.

Motivated by the presence and discovery of multiple distinct solutions in the single-step (unsupervised) NMF case, we extend our investigation to multi-step sequential decision-making problems (Reinforcement Learning). The goal in Reinforcement Learning is to learn a policy (action selection criteria based on the state of an environment) that typically maximizes average discounted reward (feedback from the environment). In Chapter 6, we introduce a method for augmenting existing state of the art policy gradient methods for RL with a diversity-inducing term in order to sequentially find a set of diverse policies that perform roughly equally well. This technique requires direct access to the environment (on-policy RL setting) in order to simulate the performance of the policy being optimized. In Chapter 7, we introduce a joint-optimization approach for finding a diverse set of policies in the off-policy setting i.e. when we only have a batch of data available and the environment cannot be directly accessed. This off-policy setting corresponds closely to the situation in some critical domains such as healthcare and we use our algorithm on a batch of clinical data to find a collection of diverse policies for the management of hypotension in the ICU.
Multiple Solutions in NMF: An Empirical Investigation

This chapter introduces Non-Negative Matrix Factorization (NMF): a popular model for data-exploration which is known to be non-identifiable, thus giving rise to a solution space that encompasses diverse factorizations. Bayesian approaches to NMF (BNMF) promise to characterize uncertainty in this model, however, we demonstrate using various (established and novel) metrics that current BNMF methods are lacking in many respects.

2.1 Problem Description

Bayesian approaches to machine learning begin by positing that the data $X$ can be explained by some probabilistic model $p(X|\theta)$, where $\theta$ is a set of parameters. Rather than finding a point estimate for $\theta$ that maximizes the likelihood $p(X|\theta)$, Bayesian approaches place a prior distribution over the parameters $p(\theta)$ and compute the posterior $p(\theta|X)$. The posterior $p(\theta|X)$ captures uncertainty in the parameters $\theta$.

For most models, the posterior $p(\theta|X)$ does not have an analytic form. In this situation, a popular approach is to approximate the posterior $p(\theta|X)$ through a set of samples $\{\theta_1, \ldots, \theta_N\}$. Approaches for generating these samples include importance and rejection sampling (Liu, 1996), se-
quential Monte Carlo (Halton, 1962), and Markov Chain Monte Carlo (Gilks, 2005).

In this chapter, we empirically explore the question: how can we assess the quality of these samples? We assume that the samples are provided by some valid Monte Carlo procedure, so we are guaranteed that the collection of samples will asymptotically approximate the true posterior $p(\theta|X)$. Most current evaluation approaches focus on two questions: (1) Has the chain mixed, that is, is it sampling from the posterior $p(\theta|X)$? and (2) How independent are the samples (as MCMC procedures produce correlated samples)? Focusing on the case of Bayesian nonnegative matrix factorization, we empirically evaluate standard metrics of sampler quality as well as propose new metrics to capture aspects that these measures fail to expose. The aspect of sampling that is of particular interest to us is the ability (or inability) of sampling methods to move between multiple optima in NMF problems. As a proxy, we propose and study a number of metrics that might quantify the diversity of a set of NMF factorizations obtained by a sampler through quantifying the coverage of the posterior distribution. We compare the performance of a number of standard sampling methods for NMF in terms of these new metrics.

2.2 Background

2.2.1 Bayesian Non-negative Matrix Factorization

In the case study below, we will evaluate several metrics of sample quality in the context of Bayesian nonnegative matrix factorization. We choose this example because it is one of the simplest and popular data exploration techniques—NMF has been used to in wide-ranging applications ranging from understanding protein-protein interactions (Greene et al., 2008), finding topics in large text corpora (Roberts et al., 2016), and discovering molecular pathways from genomic samples (Brunet et al., 2004)—with a myriad of efficient algorithms for solving it (Paisley et al., 2015; Schmidt et al., 2009; Moussaoui et al., 2006; Lin, 2007; Lee and Seung, 2001; Recht et al., 2012). However, NMF still suffers from non-identifiability: even in the exact case, there can be multiple solutions. Thus, it serves a good test case for measuring how well current Bayesian approaches can describe this uncertainty.

*There are other approaches for approximating posterior distributions, such as variational methods (Wainwright and Jordan, 2008; Tzikas et al., 2008; Opper and Archambeau, 2009); we focus on sampling-based methods here but the ideas are generally applicable.
Nonnegative Matrix Factorization and Identifiability

Given an $D \times N$ nonnegative matrix $X$ and desired rank $R$, the nonnegative matrix factorization (NMF) problem involves finding an $R \times N$ nonnegative weight matrix $W$, and an $D \times R$ nonnegative basis matrix $A$, such that $X \approx AW$. The ease of interpreting the weights $W$ and bases $A$ (due to the nonnegativity constraints), and the myriad of efficient algorithms for solving NMF (Paisley et al., 2015; Schmidt et al., 2009; Moussaoui et al., 2006; Lin, 2007; Lee and Seung, 2001; Recht et al., 2012), has made NMF—and related models, such as topic models—a popular approach to data exploration in many fields.

NMF has been used to understand protein-protein interactions (Greene et al., 2008), find topics in large text corpora (Roberts et al., 2016), and discover molecular pathways from genomic samples (Brunet et al., 2004).

However, in many cases NMF is not identifiable: there may be very different pairs $(A, W)$ and $(A', W')$ that might explain the data $X$ (perhaps almost) as well. In the following, we briefly recall some relevant terminology and properties of NMF related to the notion of identifiability.

If $X = AW$, we call the pair $(A, W)$ an exact NMF; the minimum rank $R$ such that $X$ admits an exact NMF is called the nonnegative rank of $X$ and is denoted $\text{rank}_+(X)$. A nonnegative matrix factorization, $X = AW$, can be considered trivially non-unique. Given any permutation matrix $P$ and diagonal matrix $D$ with positive entries, we obtain an alternate factorization of $X$, namely, $X = (AP^T D^{-1})(DPW)$. Since the factorization $(AP^T D^{-1}, DPW)$ differs from $(A, W)$ by scaling and relabeling of the column vectors in $A$, we consider them equivalent. Thus, we call an NMF unique if all solutions can be represented as $AQ Q^{-1}W$, where $Q$ is a monomial matrix (i.e. a product of some $P$ and some $D$).

Awareness and concerns of non-identifiability has been gaining attention among practitioners. For example, Greene et al. (2008) use ensembles of NMF solutions to model chemical interactions, while Roberts et al. (2016) conduct a detailed empirical study of multiple optima in the context of extracting topics from large corpora. These approaches use random restarts to find multiple optima.

Bayesian Nonnegative Matrix Factorization

Bayesian NMF approaches (Schmidt et al., 2009; Moussaoui et al., 2006) promise to characterize parameter uncertainty in a principled manner by solving for the posterior $p(A, W|X)$ given priors $p(A)$ and $p(W)$. Having such a representation of uncertainty in the bases and weights can further assist with the proper interpretation of the factors: we may place more confidence in subspace directions with low uncertainty, while subspace directions with more uncertainty may require further exploration. Unfortunately, in practice, the uncertainty estimates are often of limited use: sampling-based approaches (Schmidt et al., 2009;
Moussaoui et al., 2006) rarely switch between multiple modes.

In this case study, we will use the generative model of Schmidt et al. in (Schmidt et al., 2009). In this case, we place exponential priors \( p(A) \) and \( p(W) \) on \( A \) and \( W \) and choose a Gaussian likelihood. Thus, the elements of \( W, A \) are sampled from a rectified normal distributions \( R(x; \mu, \sigma^2, \lambda) \), which is proportional to the product of a Gaussian and an exponential \( N(x; \mu, \sigma^2)\text{Exp}(x; \lambda) \). The full condition for the entries in \( A|Q \) is given by

\[
p(A_{d,r}|X, A_{\setminus(d,r)}, W, \sigma^2) = R(A_{d,r}; \mu_{A_{d,r}}, \sigma^2_{A_{d,r}}, \lambda_{A_{d,r}})
\]

\[
\mu_{A_{d,r}} = \frac{\sum_n (X_{d,n} - \sum_{r' \neq r} A_{d,r'} W_{r',n})}{\sum_n W_{r,n}^2}, \quad \sigma^2_{A_{d,r}} = \frac{\sigma^2}{\sum_n W_{r,n}^2}
\]

with a symmetric update for \( W \).

2.2.2 Current Measures of Sampling Quality

Measures of Mixing. While it is practically impossible to assess whether a chain has mixed, some popular approaches include sample paths, cumulative sums, autocorrelation plots, batch means, AR and spectral analysis estimators (Johnson, 1996; Cowles and Carlin, 1996; Flegal et al., 2010).

Measures of Sample Independence. Most current approaches to measuring sampling quality focus on measuring the independence the samples from the chain. Since sample independence is hard to assess, most measures focus on correlation between samples. These include: effective sample size, autocorrelation plots, cross-correlation, integrated autocorrelation time, Hairiness Index (Cowles and Carlin, 1996; Brooks, 1998).

Other Measures In the theoretical literature, other popular measures include cover time, hitting time, etc (Aldous and Fill; Lee et al., 2015). However, these are impractical in large scenarios or when the modes of the posterior distribution are unknown.

2.3 Additional Measures of Sample Quality: Notions of Coverage

The measures in Section 2.2.2 largely focus on the independence of samples. However, even in relatively simple models, such as Bayesian NMF, it is possible for a chain to mix quickly within a single
mode and never reach an alternate mode or region. What is often missing in our discussion of practical MCMC approaches is a notion of coverage: In addition to moving in “independent” ways, how much of the posterior space does a finite-length chain explore? (Obviously given infinite time, every correct MCMC procedure will find all the modes.) In this Section, we describe several easy-to-compute and principled measures of coverage that can be applied to any set of samples, whether or not they come from a Markov Chain.

2.3.1 Measures of Similarity

To quantify the “diversity” of a set of samples, we first need a notion of distance or similarity between a pair of samples. Below, we describe two matrix similarity measures, which can be meaningfully interpreted in the context of Bayesian NMF. Later, we show that the choice of one similarity measure may be more appropriate than another depending on the NMF model and the application.

Recall from Section 2.2.1 that we consider two basis matrices $A$ and $A'$ to be equivalent (defining the same factorization of $X$) when $A' = AQ$ for some monomial matrix $Q$. Thus, we need to ensure that each similarity measure we construct is defined on equivalence classes of matrices; that is, the disimilarity of two matrices in the same class should be zero. To do this, we scale each column in our matrices to be unit in some norm and we use permutation invariant representations of matrices (e.g. matrices as unordered collections of column vectors).

**Minimum Matching Distance** For a fixed metric $m$ on $\mathbb{R}^D$, the minimum matching distance is a metric supported on sets of vectors in $\mathbb{R}^D$ (Walters, 2011). Given two subsets of $\mathbb{R}^D$, $A = \{A_1, \ldots, A_R\}$ and $A' = \{A'_1, \ldots, A'_R\}$, their minimum matching distance is defined as

$$d_{MM}(A, A') = \min_{\sigma \in S_R} \sum_{r=1}^{R} d(A_{\sigma(r)}, A'_{\sigma(r)})$$

(2.1)

where $S_R$ is the set of permutations of the index set $\{1, \ldots, R\}$. Intuitively, the minimum matching distance measures the total distance of the corresponding vectors of $A$ in $A'$, minimized over all bipartite matchings of the vectors. The minimum matching distance can be efficiently computed using the Kuhn-Munkres algorithm (Kriegel et al., 2003; Brecheisen et al.).
\[ d_{\text{MM}_1}(A, A') = \min_{P \in \mathcal{P}} \|AP - A'\|_1, \quad (2.2) \]

where \( \mathcal{P} \) is the set of \( R \times R \) permutation matrices. Note that since the columns of \( A \) and \( A' \) are unit \( \ell_1 \)-norm, \( d_{\text{MM}_1}(A, A') \) is bounded between 0 and \( R \).

The \( \ell_1 \) minimum matching distance measurement is closely related to the total variation distance for comparing discrete probability distributions. In the case of topic modeling, each column of \( A \) or \( A' \) corresponds to a “topic”, which can be interpreted as a probability distribution over words in a dictionary. The \( \ell_1 \) minimum matching distance is (up to scale) the total variation distance after pairwise matching topics in \( A \) and \( A' \) based on similarity.

**Maximum Angle Similarity**  
Masood and Doshi-Velez (2016) defines an angle-based similarity measurement that aims to capture the permutation ambiguity as well as the essence of a diverse factorization. Given \( A, A' \in \mathbb{R}^{R \times D} \), let \( \tilde{\sigma} \) be a permutation of the columns of \( A' \) that minimizes the average angle between corresponding columns,

\[ \tilde{\sigma} = \arg \min_{\sigma \in S_R} \frac{1}{R} \sum_{r \in R} \cos^{-1} \left( \frac{A_r \cdot A'_{\sigma(r)}}{\|A_r\|\|A'_{\sigma(r)}\|} \right) \quad (2.3) \]

The maximum angle similarity of \( A \) and \( A' \) is defined largest angle between corresponding columns, under the permutation \( \tilde{\sigma} \),

\[ d_{\text{Angle}}(A, A') = \max_{r \in R} \cos^{-1} \left( \frac{A_r \cdot A'_{\tilde{\sigma}(r)}}{\|A_r\|\|A'_{\tilde{\sigma}(r)}\|} \right) \]

As the entries in \( A \) and \( A' \) are non-negative, the above dot product must always be non-negative. Thus, \( d_{\text{Angle}}(A, A') \) is bounded between 0 and \( \pi/2 \).

Since the maximum angle similarity is measurement of orientation, we can apply the same measurement to basis elements of different dimensions and interpret the results in a similar manner. By focusing only on measuring the maximum angle, we allow for an extreme case where two factorizations only differ by one basis element. This may well be the case in some scenario so we incorporate it as a feature of this distance measurement.
In addition, by considering the maximum angle, our similarity measure gains a certain degree of robustness to column perturbations. For example, a similarity measurement based on the sum of angles between basis elements would not be successful in differentiating between a case where one basis element is significantly different versus if all basis elements were just perturbed a little bit. We want our understanding of diversity to be robust to perturbation of existing basis columns.

**Notes and Observations about similarity in NMF basis factors:** Even when we are not dealing with factors in the same equivalence class, these measures of similarity capture the sense of diversity in solutions that is of interest to us. Consider as a baseline, measuring the similarity as the Frobenius error of the difference i.e. \( \| A - A' \|_F \). Figure 2.1 shows two pairs of matrices \((A_1, A_2)\) and \((A_1, A_3)\) that have the same difference in the naive Frobenius sense (\(d_F = 0.122\)) but different maximum angle and \(l_1\)-minimum matching similarity scores. From figure 2.1 it is also clear that the \((A_1, A_3)\) pair is the more dissimilar.

Finally, we point out that the permutation computed in the minimum matching distance, which minimizing the total distance between matrices, is not necessarily the same as the permutation, which minimizes the average angle between corresponding columns. In practice, this means that permutations used to compute the minimum matching distance cannot be used to compute the maximum angle distance (and vice versa). It also raises an interesting question about what is means in empirical work to ‘correct’ for the permutation ambiguity since we’ve observed that this correction is metric-dependent.

\[
\begin{align*}
(a) \quad d_{MM_{l_1}}(A_1, A_2) &= 0.137, d_{\text{Angle}}(A_1, A_2) = 6.38 \\
(b) \quad d_{MM_{l_1}}(A_1, A_3) &= 0.400, d_{\text{Angle}}(A_1, A_3) = 20.44
\end{align*}
\]

**Figure 2.1:** Comparison of max angle and \(l_1\)-min match measures when \(\| A_1 - A_2 \|_F = \| A_1 - A_3 \|_F\).
2.3.2 Measures of Coverage

Given a set, $S$, of samples from some parameter space explored by a sampler, and given a similarity measure $m$ for pairs of samples, we introduce three measurements of the diversity of the samples contained in $S$.

**Maximum Pairwise Distance** We define a notion of diversity for a set $S$ based on the “diameter” of $S$, that is, we compute the maximum distance between pairs of points in $S$. The maximum pairwise distance of $S$ is defined as

$$\text{MaxDist}(S) = \max_{A,A' \in S} d(A, A).$$

(2.4)

**Mean Pairwise Distance** We can alternatively quantify the diversity of $S$ by approximating the “density” of points in $S$. Motivated by this intuition, we define the mean pairwise distance of $S$ as

$$\text{MeanDist}(S) = \frac{1}{|A|^2} \sum_{A,A' \in S} d(A, A).$$

(2.5)

**Covering Number** We quantify the amount of the parameter space explored by the sampler, by approximating a notion of “volume” for a set of samples $S$. For each $\epsilon > 0$, we define the minimum covering number of $S$, denoted $C_\epsilon(S)$, as the cardinality of the smallest subset $S' \subset S$ such that $\bigcup_{s \in S'} B(s, \epsilon)$ covers $S$, where $B(s, \epsilon)$ is the $\epsilon$ ball centered at $s$ with respect to some metric or similarity measure. Our minimum covering number can be stated in terms of the covering number of graphs of the $\epsilon$-neighbor graph of the points in $S$. We note that the covering number is a frequently studied property of graphs in literature (Abbott and Liu, 1979; Chepoi et al., 2007).

Clearly, $C_\epsilon(S)$ depends on the choice of $\epsilon$. When $\epsilon = 0$, the minimum covering number is equal to the cardinality of $S$; for sufficiently large $\epsilon$, the minimum covering number is 1, since an $\epsilon$-ball centered at any element will contain the entire set $S$. It is also straightforward to see that $C_\epsilon$ is a monotone increasing as a function of $\epsilon$.

Generally speaking, for a fixed $\epsilon$, the larger the minimum covering number, the more of the parameter space covered by the sampler. However, cases may arise, for $\epsilon_1 < \epsilon_2$ and sets $S_1, S_2$, where $C_{\epsilon_1}(S_1) > C_{\epsilon_1}(S_2)$ but $C_{\epsilon_2}(S_1) < C_{\epsilon_2}(S_2)$. In Figure 2.2 and Figure 2.3, we see samples $S_1, S_2$ from a mixture of two Gaussians, where $S_1$ is concentrated in one mode and $S_2$ is distributed
amongst both modes. The latter is demonstrated by the fact that $C_{\epsilon_2}(S_2) > C_{\epsilon_2}(S_1)$ for sufficiently large choices of $\epsilon$.

To avoid arbitrariness in selecting a single value for $\epsilon$, we compute the minimum covering number for a range for values between $\epsilon = 0$ and $\epsilon = t$, for $t$ is sufficiently large that each $\epsilon$-ball covers $S$. We interpret covering numbers which “persists” for significant intervals of $\epsilon$ to be revealing the diversity of the sample set and covering numbers which appear for small intervals of $\epsilon$ to be negligible (due to small variations in the set). The motivation for our interpretation is based in the body of work on persistent topology (Edelsbrunner and Harer, 2008; Carlsson, 2009; Ghrist, 2008), in which topological features of manifolds are deduced from an approximation given by a set of sample points by interpreting the features which persist in the reconstruction across a number of resolutions.

We call the collection of the covering numbers which persists for large intervals of $\epsilon$, or, in a slight abuse of language, the collection of covering numbers for all $\epsilon$, the persistent minimum covering numbers. Figure 2.4 shows the plots of the persistent minimum covering numbers of the sets $S_1$ and $S_2$, from Figure 2.2, as a function of $\epsilon$. These plots intuitively demonstrate the greater diversity of $S_2$, as $C_\epsilon(S_2)$ persists above $t$ for a greater interval of $\epsilon$-values.

(a) Sample $S_1$ lie in a single mode. The minimum covering number for $\epsilon = 1$ is $C_\epsilon(S_1) = 3$.

(b) Sample $S_2$ lie in two modes. The minimum covering number for $\epsilon = 1$ is $C_\epsilon(S_2) = 2$.

Figure 2.2: Comparison of minimum covering numbers of samples ($\epsilon = 1$)
2.4 Case Study

In the following, we consider three different synthetic data sets with qualitatively different posterior structure: one with a single mode, one with two modes, and one with an infinite number of modes (or rather, a connected space of equally good solutions). We consider both a tiny version of the NMF problem, with only 4 or 5 observations and 6 or 7 dimensions, as well as larger one with 50 observations and 500 dimensions. We evaluate a number of sampling algorithms with various measures of “diversity”, running each algorithm for 10 repetitions of 10,000 samples. To avoid questions of whether samplers like Gibbs and HMC have mixed, we initialize them at one of the modes/maximum likelihood solutions.
2.4.1 Synthetic Data Sets

**Unique Solution** Laurberg et al. (2008) gives the following example, $X = WA$, where the NMF solution is a unique equivalence class of factorizations, for the value $a = 0.3$.

$$W = \begin{pmatrix} a & 1 & 1 & a & 0 & 0 \\ 1 & a & 0 & 0 & a & 1 \\ 0 & 0 & a & 1 & 1 & a \end{pmatrix}, \quad A = W^T. \quad (2.6)$$

Under our Bayesian model, the posterior space of this NMF is unimodal.

**Two Solutions** Simply by setting the value of $a$ to be 0.5 in the previous example, Laurberg et al. (2008) shows that $X$ will have two distinct solutions. In particular, these solutions are related by a change of basis matrix $Q$, so the two factorization cones are in the same subspace.

$$W = \begin{pmatrix} 0.5 & 1 & 1 & 0.5 & 0 & 0 \\ 1 & 0.5 & 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0.5 & 1 & 1 & 0.5 \end{pmatrix}, \quad A = W^T; \quad Q = Q^{-1} = \frac{1}{3} \begin{pmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{pmatrix}. \quad (2.7)$$

Under our Bayesian model, the posterior space of this NMF is bimodal.

**Infinite Solutions** Finally, for the following matrix,

$$X = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (2.8)$$
there are an infinite number of nonnegative factorizations. For any $\delta \in [0, 1]$, we have $X = A_\delta W$, where

$$A_\delta = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 - \delta & 1 - \delta & 1 - \delta & \delta & \delta & \delta \end{pmatrix}, \quad W = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$ (2.9)

Under our Bayesian model, the posterior space of this NMF contains an infinite number of modes that form a connected subset of the parameter space.

**Larger Data Sets**

The above examples of data sets are valuable performing diagnostics on sampling algorithms and comparing diversity measures, because we know ground truth about the solution space for each example. The larger data sets we work with are simply high dimensional embeddings of these small data sets, chosen such that the desirable properties (such as the number of solutions) of the original data are preserved. To generate the larger data sets, we take a small $D \times N$ data $X$ and transform it using non-negative matrices $B_1$ and $B_2$ such that $X_{\text{large}} = B_1 X B_2$. For an NMF data set, the factors transform in a simple manner: $A_{\text{large}} = B_1 A$ and $W_{\text{large}} B_2$. In our experiments, the larger data sets have dimensions $D = 500$ and $N = 50$ (while relatively small, we show that even with data sets of this size samplers rarely move very far).

**2.4.2 Inference Approaches**

In constructing our chain of factorization matrices, we add Gaussian noise, with standard deviation $\sigma$, to the data matrix $X$, for which we know the exact factorizations. We call the noisy data $\tilde{X}$.

**Lower Bound: Single Mode** To provide a reasonable lower bound on coverage, we establish the single-mode baseline, wherein we generate a set of samples that only explore one mode or region. For this baseline, we fix one known exact solution, $X = W A$; to generate samples consistent with
the Gaussian noise model, we initialize an NMF algorithm at $(W, A)$ and iteratively generate factorizations for the data $X$ with added Gaussian noise. The procedure is as follows: we initialize the multiplicative update algorithm of Lee and Seung (1999) with $(W, A)$; we then run the algorithm for the data $\hat{X} + E_i$, where $E_i$ is the noise matrix for the additional Gaussian noise added at the $i$th iteration. In this fashion, we obtain variation in the entries of our factorization matrices (depending on the noise-level), but we know these variation is of a limited scale and that the chain contain samples from only one mode.

**Upper Bound: All Modes**  To provide a reasonable approximation of ideal coverage, we establish the all-modes baseline, wherein generates a set of samples that covers the entire solution space. At each iteration in the chain, we randomly pick one known exact solution. When the solutions are discrete, we uniformly sample them. In the infinite solutions case, we pick $\delta_i$ independently from the uniform distribution over $[0,1]$. We then apply the same perturb-and-factor methodology as in the single mode baseline to produce a set of factors that approximate the modes of the parameter space.

Gibbs Schmidt et al. (2009) propose a Bayesian approach to NMF in which each basis element $A_{d,r}$ and each weight $W_{r,n}$ is sampled independently from an exponential prior. Combined with a Gaussian likelihood, the updates for each element of $A$ and $W$ can be sampled element-wise from a rectified normal distribution, which is proportional to the product of a Gaussian and an exponential.

We initialize with a sample from the All Modes baseline so that no burn-in needed. We fix $\sigma$, the noise parameter in the Gaussian noise, as the same value used in generating $\hat{X}$.

**Hamiltonian Monte Carlo**  Hamiltonian Monte Carlo (HMC) is a Markov Chain Monte Carlo (MCMC) procedure that simulates Hamiltonian dynamics over the target distribution state space. The time-reversibility and volume preserving properties of the evolution of Hamiltonian systems ensure detailed balance (Neal et al., 2011). By incorporating gradient information, HMC suppresses the random walk behavior which contributes to the inefficiency of many MCMC methods. Given a target density $\pi(q)$, we define a Hamiltonian of the form

$$H(q, p) = -\log \pi_H(q) + \frac{1}{2} p^\top p$$  \hspace{1cm} (2.10)
where $-\log \pi(q)$ is called the potential energy and the velocity, $p$, is given by a certain tangent vector at $q$. We simulate the dynamics with a discretized integrator, called the leap-frog integrator. Given an initial state $(q_0, p_0)$, a proposal state $(q_n, p_n)$ is reached by a series of steps in the direction of $\nabla_q \log \pi_H(q)$ (Byrne and Girolami, 2013).

In the case of NMF, our target density, $\pi(W, A)$, is the log posterior distribution corresponding to our generative model. Since HMC have been demonstrated to be more successful than other MCMC methods in crossing multiple modes, for low-dimensional posterior spaces, one might expect HMC to out-perform Gibbs in finding multiple optima for NMF problems as well.

For our experiments, we implement HMC with adaptive step size and a fixed 100 leap count. We perform the HMC on the entry-wise log of the factor matrices in order to keep our solutions feasible. We initialize with a sample from the All Modes baseline and allow an additional burn-in period of 200 iterations so that we can adaptively find a reasonable step-size.

**Non-MCMC Baseline: (Filtered) Random Restarts** We run Lin’s projected gradient NMF algorithm (Lin, 2007) with random initializations. We expect this algorithm to explore multiple solutions when they exist because it is initialized at a random factor matrix each time. There are no asymptotic guarantees for exploring all solutions nor is there an underlying probabilistic framework. However, random restarts provides a baseline of how much coverage we might achieve in practice if we did not have access to the true solution but were not constrained to a probabilistic MCMC framework.

In practice, deterministic optimization algorithms such as the projected gradient algorithm are prone to get caught in very poor local optima (Lin, 2007). Thus, we report metrics for all random restart outputs as well as the subset of random restarts results that achieve likelihoods comparable with the other samplers (that is, we exclude random restarts results from local optima corresponding to very poor solutions). We call this second baseline filtered random restarts. In our experiments, we allow for reconstruction errors in the filtered samples to be up to ten times the maximum reconstruction error in the Gibbs chain.

### 2.4.3 Evaluation Metrics

For our experiments, we report evaluation metrics measuring three key aspects of sampling: quality of fit, sample independence and coverage of the parameter space.
Quality of Fit  We measure the quality of NMF solutions contained in a set of samples by computing the mean likelihood of the samples. Because methods like random restarts can produce samples from low-likelihood local optima that are widely distributed in the posterior space, the mean likelihood can be used to distinguish samples from multiple modes from those from multiple local optima.

Sample Independence  We measure the independence of the samples in a set using integrated autocorrelation time. Although sample independence can be assessed in a number of different ways, we choose to use only integrated autocorrelation time since previous work (Cowles and Carlin, 1996; Brooks and Roberts, 1998a; Kass et al., 1998; Brooks and Roberts, 1998b) have shown IAT to be a reasonable representative of this class of metrics.

Coverage  To assess the amount of coverage achieved by a set of samples, we compute the maximum pairwise distance, the mean pairwise distance, and the persistent minimum covering numbers. Furthermore, each metric of coverage is calculated with respect to both maximum angle similarity and $\ell_1$-minimum matching distance.

The persistent minimum covering numbers are computed over a range of 100 $\epsilon$ values and for the first 1000 elements of each chain. For each $\epsilon$ value, the minimum covering number we record is the mean of $C_\epsilon$ over 10 repetitions of the same experiment. As the set cover problem is NP-hard, we employ a greedy algorithm to approximate the coverage (Chvatal, 1979).

2.5 Results

Quality of Fit: Log-Likelihoods  By design, the one mode and all modes sampling algorithms fit the data well (see Table A.1 in Appendix A and Table A.5 in Appendix A). Since the Gibbs sampler the HMC sampler are initialized at a known mode, they also fit the data well while exploring the posterior space around that mode.

Random Restarts, on the other hand, has much lower likelihoods than the samplers. This result is expected (although perhaps not at this scale). It appears that, even on small toy data sets, Random Restarts is prone to being trapped in local optima around low-quality solutions. The presence of such poor factorizations in Random Restarts motivates the construction of the Filtered Random Restarts baseline, which contains the subset of Random Restart samples whose reconstruction error in the Frobenius norm $\|X - AW\|_F$ falls within a fixed threshold. In constructing the filtered sam-
samples, we note that a large proportion of the Random Restarts samples from the toy-sized data sets fall within our allowable reconstruction criterion, while significantly fewer samples are allowed to filter through when we consider the larger $500 \times 50$ data sets. In particular, the worst performance of Random Restarts is observed in the enlarged version of the bimodal data, wherein, within ten repetitions, the number of sample satisfying our filtering criteria ranged between 5 and 309 out of a total of 10,000 samples.

**Sample Independence: IAT** The MCMC algorithms show high degrees of autocorrelation rendering the effective sample size of the 10,000 chain to be orders of magnitude smaller (Table A.2 in Appendix A and Table A.6 in Appendix A). Even on toy data sets with infinite number of connected solutions, the samples are highly correlated. As expected, the non-MCMC samples are effectively independent.

**Coverage: Pairwise Distance and Minimum Covering Numbers** We note again that the likelihood and sample independence does not necessarily provide information about the region of the posterior that is being explored in these chains. For example, in the infinite solutions case, one can imagine a trajectory through the line of solutions parameterized by $\delta$ that could generate a set of samples with high autocorrelation but traverse along a large portion of the posterior. On the other hand, samples could appear uncorrelated when a chain is only making local but independent moves around a single mode. To assess coverage, we study the maximum/mean pairwise distance and persistent minimum covering numbers.

Both the pairwise distance measures (Table A.3, Table A.7, Table A.4, Table A.8) and the persistent covering numbers (Figures A.1 - A.6) indicate that Random Restarts obtains the widest range of solutions. However, because some (or many, depending on the data set) of the elements correspond to poor factorizations, they do not represent samples from multiple modes of the posterior space.† Thus, the Filtered Random Restart samples reveals a better picture of the movement of this sampling algorithm through the posterior. The coverage metrics indicate that the Filtered Random Restart samples still explore significantly more of the posterior than the MCMC methods. In particular, in the toy sized bimodal data set, the Filtered Random Restart samples appears to sample from both modes.

†Occasionally, Random Restarts yields a solution with a column of zeros in the basis matrix $A$, this sort of degenerate sample gives rise to a maximum pairwise angle distance of 90 degree for the entire samples (Table A.3).
The persistence covering number plots of the All Modes samples establish clear visual baselines for distinguishing sets containing samples from multiple modes in the posterior space. Comparing the plots of the Gibbs sampler and HMC against those of All Modes, we see these asymptotically correct sampling mechanisms tend to explore only a single mode (Figures A.1 - A.6).

2.6 Conclusion

We conclude from our experiments that the coverage metrics we introduce yield useful information regarding sampling behaviors that cannot be assessed using traditional MCMC diagnostics, such as the ability of the sampler to cover a large portion of the posterior space. We are able to evaluate and compare the exploratory behavior of chains through these metrics. Within the sampling algorithms we study, it appears that a filtered version of Random Restarts would show the best exploratory behavior while maintaining some quality of the factorization. In the data sets explored, the quality of the factorization deteriorates for Random Restarts as the scale of the data grows.

Both MCMC approaches, Gibbs and HMC, produce factorizations of excellent quality but IAT shows that effective sample size is very small. Furthermore, our coverage analysis reveals that neither Gibbs nor HMC are able to explore the multiple modes in very small toy data sets. As the scale of the data set grows, we expect the Gibbs sampler to be more vulnerable to becoming trapped in a single mode due of the local nature of its updates.

For the practitioner of NMF, these results first demonstrate that random restarts of some sort—such as running multiple chains—can still be important for covering the posterior space, even for very small problems. More importantly, we argue for adding coverage analysis of solutions to the standard diagnostic process evaluating sampling-based approaches, since solution set diversity is an unavoidable questing arising from the identifiability of NMF and other problems. These multiple solutions can lead to very different interpretations of the data and effect subsequent modeling decisions. We hope that the metrics introduced in this chapter to quantify the notion of sample diversity begins to fill the gap left by current performance measurement of sampling algorithms.
Multiple Solutions in NMF: A Geometric Approach to Faster-Mixing MCMC

In Chapter 2, we saw that current Bayesian Non-negative Matrix Factorization (BNMF) algorithms tend to fail in appropriate coverage of the posterior distribution as they typically explore a single mode only. In this Chapter, we attempt to improve upon a Bayesian NMF Gibbs sampler using a Metropolis-Hastings (MH) proposal mechanism that is inspired by the geometric structure of NMF. The aim of this MH proposal is to aid in faster mixing of the MCMC chain and better coverage of the posterior.

3.1 Problem Description

Nonnegative matrix factorization (Lee and Seung, 1999), has received much attention for its ability to extract interpretable decompositions in a variety of applications, including image processing (Yang et al., 2007), topic modeling (Arora et al., 2012a), music analysis (Févotte et al., 2009) and blind source separation (Cichocki et al., 2006). Given a $D \times N$ data matrix $X$ consisting of $N$ observations of dimension $D$, the non-negative matrix factorization problem is to find non-negative factors $A \geq 0, W \geq 0$ such that $X \approx AW$. Here, the basis $A$ is a $D \times R$ matrix that represents the cone within the positive orthant in which the data $X$ live, and the weights $W$ is $R \times N$ which represents the convex combinations of the basis elements needed to reconstruct the data.
Bayesian approaches to NMF (Schmidt et al., 2009; Moussaoui et al., 2006) place exponential priors $E(\lambda)$ over the basis matrix $A$ and the weight matrix $W$.

$$p(A) = \prod_{d,r} E(A_{d,r}; \lambda_{A_{d,r}})$$

$$p(W) = \prod_{r,n} E(W_{r,n}; \lambda_{W_{r,n}})$$

and then solve for the posterior $P(A, W | X)$. Solving for a posterior distribution, rather than a point estimate, provides a representation of uncertainty in the bases and weights, which can further assist with the proper interpretation of the factors: we may place more confidence in subspace directions with low uncertainty, while subspace directions with more uncertainty may be areas requiring greater exploration.

While providing measures of uncertainty is cited as one of the key advantages of Bayesian approaches to NMF, in practice, the uncertainty estimates are often of limited use: variational approaches to Bayesian NMF (Paisley et al.) find only the uncertainty around a single posterior mode, and sampling-based approaches (Schmidt et al., 2009), (Moussaoui et al., 2006) also rarely switch between multiple modes. Thus, these approaches will typically miss situations in which there is a completely different cone that could have explained the data (perhaps almost) as well. When NMF factors are interpreted to have meaning (e.g. topic modeling, feature extraction from images etc), then characterizing this form of uncertainty is critical. Even if we are only interested in dimensionality reduction, knowing that there are multiple similar quality factorizations may allow the domain expert to choose a cone that will generalize to unobserved data.

In this Chapter, we develop a novel reformulation of the NMF problem that naturally leads to an MH step that results in faster mixing chains. The correctness of our approach is guaranteed by standard MCMC results; we show that our steps create chains with lower integrated autocorrelation times and more varied samples than the standard Gibbs sampling approach.

### 3.2 Background

One standard definition of the NMF problem is to find factors $A \geq 0$ and $W \geq 0$ of rank $R$ that minimize the Frobenius error of the reconstruction:

$$\arg\min_{A,W}(\|X - AW\|) \quad s.t. \quad A \geq 0, W \geq 0.$$
The objective in equation 3.2 is also equivalent to a Gaussian likelihood

\[ p(X|A, W) = \prod_{d,n} \mathcal{N}(X_{d,n}, (AW)_{d,n}, \sigma^2) \]  

(3.3)

We can view the columns of the factor matrix \( A \) as vectors that define a cone within which most of the data lie. The corresponding \( W \) matrix is then viewed as a weights matrix with respect to the basis of this cone. Geometrically, the NMF problem can be stated as one of finding a cone of minimal order that is a subset of the positive orthant but a superset of the cone defined by the extreme points (columns) of the data matrix \( X \) (Gillis, 2012).

The NMF objective in equation 3.2 is ill-posed in the sense that there is always a permutation and scaling ambiguity. Given a permutation matrix \( P \) and a diagonal matrix \( D \) with positive entries, if \( X = AW \), then \( X = ADP P^T D^{-1} W = A_2 W_2 \) where \( A_2 = ADP \) and \( W_2 = P^T D^{-1} W \). Placing priors on \( A \) and \( W \) can reduce the scale ambiguity, but more generally scale and permutation ambiguity are not interesting or concerning from an interpretation perspective.

![Figure 3.1](image.png)

**Figure 3.1:** The dotted triangles show two rank-3 factorizations of data whose convex hull is shaded in grey.

However, there are situations in which a factorization \( A, W \) is not unique in nontrivial ways. For example, in figure 3.2, we show two factorizations represented by triangles whose vertices are the basis elements of the factorizations. Many papers have established conditions under which the factorization \( A, W \) is unique up to scale and permutation (Donoho and Stodden, 2004; Laurberg et al., 2008; Gillis, 2012; Arora et al., 2012). For example, Donoho and Stodden (2004) defines separable factorial articulation families of data that are generated through systematic articulations of parts that each data point consists of. They show that when the parts of the data appear in all possible articulations, the cone defined by the data gives rise to a unique low-dimensional cone so that the NMF is unique.
Laurberg et al. (2008) starts with the observation that if the positive rank is the rank of the matrix, that is rank\( (X) = \text{rank}^+(X) \), and we are given some factorization \( X = AW \), then any alternative factorization \( A'W' \) must be of the form \( AQQ^{-1}W = A'W' \) and go on to show that NMF is unique when the change of basis \( Q \) is limited to only monomial matrices.

Unfortunately, in many cases we may not know whether the assumptions required for uniqueness are met, and then there may exist fundamentally different modes. In the following, we use the geometric insights from Laurberg et al. (2008) to create a novel reformulation of the NMF problem that naturally suggests a faster-mixing MCMC procedure for Bayesian NMF.

Bayesian NMF (Schmidt et al., 2009) propose a Bayesian approach to NMF with independent exponential priors on each element of the factors \( A \) and \( W \) (equation 3.1). Assuming independent Gaussian-distributed noise (equation 3.3), the updates for each element of \( A \) and \( W \) can be efficiently sampled from a rectified normal distribution \( R(x; \mu, \sigma^2, \lambda) \) which is proportional to the product of a Gaussian and an exponential \( N(x; \mu, \sigma^2)E(x; \lambda) \).

\[
p(A_{d,r} | X, A_{(d,r)}, W, \sigma^2) = R(A_{d,r}; \mu_{A_{d,r}}, \sigma^2_{A_{d,r}} \lambda_{A_{d,r}})
\]

\[
\mu_{A_{d,r}} = \frac{\sum_n (X_{d,n} - \sum_{r' \neq r} A_{d,r'} W_{r',n})}{\sum_n W_{r,n}^2}
\]

\[
\sigma^2_{A_{d,r}} = \frac{\sigma^2}{\sum_n W_{r,n}^2}
\]

(3.4)

A symmetrical update equation can be derived for elements of \( W \). Since the elements in each column of \( A \) and row of \( W \) are conditionally independent, in practical implementations, we sample columns of \( A \) and rows of \( W \) at a time. However, an alternating sampling procedure that samples only one column of \( A \) and one row of \( W \) at a time can make it hard for the sampler to make large moves in the space of factors \( A \); in practice the sampler tends to explore only one mode.

3.3 Reformulating the non-negativity constraint with Extreme Rays

As noted in Laurberg et al. (2008), if rank\( (X) = \text{rank}^+(X) \), and we are given some factorization \( X = AW \), then any alternative factorization \((A', W')\) must be of the form \( AQQ^{-1}W = A'W' \). The matrix \( Q \) can be thought of a change-of-basis matrix within the same subspace spanned by the basis matrix \( A \). In this Section, we use the vertex enumeration problem for polyhedra (Motzkin
et al., 1953) to further characterize the space of change-of-basis matrices $Q$; in Section 3.4 we will augment the standard Gibbs sampler (equations in Schmidt et al., 2009) with MH steps that propose change-of-bases $Q$ to allow for global changes to the factors $A$ and $W$.

For a change-of-basis matrix $Q$ to be a valid proposal, it must satisfy the following constraints:

$$AQ \geq 0 \quad Q^{-1}W \geq 0 \quad \det Q \neq 0$$

(3.5)

In addition, we are specifically interested in $Q$ matrices that satisfy the above constraints but cannot be written as a product of diagonal and permutation matrices (i.e. monomial matrices).

Below, we describe a relaxation that finds two matrices $Q_A$ and $Q_W$ such that $AQ_A \geq 0$, $Q_W W \geq 0$, and $Q_A Q_W \approx I$. The inequality constraint $AQ_A \geq 0$ implies that $Q_A$ is a column of the $Q_A$ matrix. This linear inequality $AQ_A \geq 0$ is a half-space intersection description of a polyhedron. In the case of non-negative matrices, the solution space is necessarily non-empty as all vectors in $q_A$ in the positive orthant satisfy the conditions. We also know that this polyhedron is necessarily unbounded with a point at the origin, making it a pointed cone.

A convex polyhedron can be described as a collection of overlapping half-spaces or a collection of vectors known as extreme rays (Motzkin et al., 1953) (Fukuda and Prodon, 1996). As an example, figure 3.2 illustrates the half-spaces implied by the inequality constraint $AQ_A \geq 0$ for a toy example involving a rank-2 factorization of a 3-dimensional problem (so $A$ is $3 \times 2$). Each row in the matrix is shown as a colored vector in the positive orthant and the shaded region of the same color represents the space where the dot product is negative. There exist many redundant constraints; the minimal set of constraints (shown as dotted lines in figure 3.2) are the extreme rays of the pointed cone that contains solutions satisfying $AQ_A \geq 0$. For example, in figure 3.2, the rays corresponding to the red and green vectors are the ones that contribute to finding the cone where all vectors $q_A$ lie such that $AQ_A \geq 0$. Fukuda and Prodon (1996) provide efficient algorithms using computational geometry to find the extreme rays. We also note that the extreme rays can contain negative values; in the separable case the extreme rays will be standard basis elements (and in that case, the only solutions will be cones that are subsets of the initial factorization $A$). If $A$ is a minimum volume factorization, then the only $Q_A$ possible will be monomial matrices Laurberg et al. (2008)).

The extreme rays allow us to compactly describe the space of feasible solutions to $AQ_A \geq 0$. Let the matrix $R_A$ be an $R \times n_A$ matrix with each of the extreme rays of $A$ in its $n_A$ columns. Then the linear inequality of the columns $q_A$,

$$AQ_A \geq 0$$
Figure 3.2: A toy illustration of the extreme rays. The dotted green and red lines are the extreme rays and define the cone where all vectors $q_A$ lie such that $Aq_A \geq 0$.

can be equivalently written as

$$q_A = \{ R_A y_A \mid y_A \geq 0 \}$$

Here, the vector $y_A$ represents the non-negative combination of weights of the extreme rays needed to construct a point in the cone. Thus, the problem of finding a matrix $Q_A$ that satisfies the constraint that $Aq_A \geq 0$ reduces to choosing a nonnegative $n_A \times R$ matrix $Y_A$; our new $A' = AR_A Y_A$. Geometrically, different $Y_A$ explore the $R$ dimensional subspace defined by the $A$ matrix. We can view the $K \times n_A$ matrix $R_A$ as a cone of order $n_A$ in $\mathbb{R}^K$ that is a superset of of the non-negative orthant; the non-negative orthant corresponds to the original factorization $A$. In the following, we will use the observation that very different $Y_A$ matrices, by utilizing different extreme rays, will result in large, global changes to the basis matrix $A$.

3.4 Extreme Ray Metropolis Hastings

In this Section, we introduce our novel MH scheme that leverages the geometric insights from Section 3.3. In the following, we will refer to our approach as Extreme-Ray Metropolis Hastings (ERMH).

Once the matrix $A$ is fixed and the extreme ray matrix $R_A$ is found, there exists a one to one relationship between a matrix of convex combinations $Y_A$ and change of basis matrices $Q$. To avoid
the scaling ambiguity that exists in these factor matrices, we choose the convention of having column stochastic $Y_A$ matrices. Specifically, in our MH proposal, we sample columns of the $Y_A$ matrix from a Dirichlet distribution $D(\alpha_0)$ with a uniform concentration parameter $\alpha_0$. After generating a proposal $A_p = A R_A Y_A$, we solve for the best weights matrix corresponding to that basis, and add noise from a truncated Gaussian in order to propose the corresponding weights matrix in the MH proposal $W_p$. The full algorithm is described in algorithm 3.1.

**Algorithm 3.1: Generating Proposal Factorizations**

Input: Baseline Factors $A_b, W_b$, Proposal noise $\sigma_p$, Dirichlet parameter $\alpha_0$

for $i = 1$ to $r$ do
    Sample $(Y_A)_r$ from Dirichlet $\sim D((Y_A)_r, \alpha_0)$
end for

Compute change of basis matrix $Q = R_A Y_A$
Set new proposal factor $A_p = A_b Q$
Compute associated optimal weights $W^\ast$ via non-negative least squares
Sample $W'$ from a truncated Gaussian $N_T(W^\ast, \sigma_p)$
Output: Proposal Factors $A', W'$

Given the current factors $(A, W)$ and the proposed factors $(A', W')$, the acceptance probability is given by

$$
\alpha = \min \left( 1, \frac{p(X|A', W') p(A') p(W') q(A) q(W|X, A)}{p(X|A, W) p(A) p(W) q(A') q(W'|X, A')} \right)
$$

where the probability of the proposed $(A', W')$ is independent of the current sample $(A, W)$; the values of the new basis $A'$ only depend on the extreme rays $R_A$ of the reference sample $A_b$ and the sampled columns of $Y_A$:

$$
q(A) = \prod_r D((Y_A)_r, \alpha_0)
$$

Similarly, the values of $q(W')$ depends only on the corresponding $A'$ that we have sampled and the data $X$. Specifically, let $W^\ast$ be the best weights matrix for the data $X$ given a basis factorization $A'$. We add truncated Gaussian noise to $W^\ast$ to generate a proposed $W'$ with positive elements:

$$
q(W|X, A) = N_T(W; W^\ast, \sigma_p)
$$

where $\sigma_p$ is the variance of the Gaussian noise.
Finally, the likelihood terms are straightforward to compute:

\[
p(X|A, W) = \prod_{d,n} \mathcal{N}(X_{d,n}, (AW)_{d,n}, \sigma^2)
\]
\[
p(A) = \prod_{d,r} \mathcal{E}(A_{d,r}; \lambda_{A_{d,r}})
\]
\[
p(W) = \prod_{r,n} \mathcal{E}(A_{r,n}; \lambda_{W_{r,n}})
\]

**Choice of concentration parameter \(\alpha_0\)** In initial explorations, we observed that the choice of weights \(Y_A\) that covers that original factorization, that is, \(R_A Y_A = I\), tends to be sparse. More generally, sparse weight matrices \(Y_A\) result in a proposal \(A'\) that lies close to the lower-dimensional facets of the extreme ray cone. Since the extreme ray cone is a superset of the non-negative orthant, sparse weight matrices \(Y_A\) will also tend to give factorizations covering large hypervolumes. In contrast, a dense matrix \(Y_A\) will encourage the cone of the new factorization to be well inside the non-negative orthant, making it less less likely to be able to explain data that are spread in the non-negative orthant of \(\mathbb{R}^K\). Thus we use a small concentration parameter \(\alpha_0\) for our proposals.

**Computational Considerations and Clustering of Extreme Rays** The algorithm to find the extreme rays is an incremental method and its run time is highly sensitive to the ordering of the rows of the matrix. Heuristics to find good orderings are employed by extreme ray algorithms. In our experiments, we found that while the algorithm is designed to eliminate redundant or degenerate inequalities, we find some extreme rays to be directionally very close to each other. This may happen for inequalities that are nearly degenerate. At runtime, each proposal requires solving a system with \(K \times n_A\) variables. In the worst case, there may be combinatorial growth of extreme rays with number of rows of \(A\) (McMullen and Shephard, 1971). To reduce the computational burden, we employ a clustering approach using DBSCAN (Ester et al., 1996) and a directional distance measurement to pick a subset of extreme rays. Note that while we may need all the extreme rays for a perfect reconstruction, we can pick a subset of the extreme rays and still have a valid sampler.

### 3.5 Evaluation Approach

Both our proposed sampler and the Gibbs sampler satisfy detailed balance, and thus both approaches will, asymptotically, approximate the true posterior distribution. However, more variation among
the samples in the chain reflects faster exploration of the posterior space. In our evaluations in Section 3.6, we shall consider the following two metrics to evaluate the rate of exploration of the samplers.

**Autocorrelation** The first is simply the average autocorrelation at lag 1 between corresponding elements in samples of the factor matrix $A$. The autocorrelation of a sequence $\{x\}_i$ at lag $l$ is given by:

$$\rho(l) = \frac{1}{N} \sum_{i=1}^{N} x(i)x(i-l)$$

A faster drop-off in autocorrelation implies greater independence between the samples and a larger effective sample size.

**Integrated Autocorrelation time** The autocorrelation time is a direct measure of the number of evaluations of the posterior PDF required to produce independent samples of the target density (Foreman-Mackey et al., 2013). Shorter autocorrelation times correspond to faster mixing chains. The autocorrelation time for a sequence of samples $x$ is given as:

$$\tau = 1 + 2 \sum_{T=1}^{\infty} \frac{C(T)}{C(0)}$$

$$C(T) = \lim_{t \to \infty} \text{cov}(x(t+T), x(t))$$

We use the python module *PyMC* (Patil et al., 2010) for estimating autocorrelation time from our samples.

**Maximum Angular Distance** We note that autocorrelation can be small even when a sampler is simply exploring a single mode if the hops around that mode are sufficiently independent. To compare more explicitly the variation in the chains proposed by our sampler as opposed to a standard Gibbs-sampling scheme, we propose an angle distance measure that finds the maximum angle between corresponding angles after finding the permutation that minimizes the average angle between corresponding columns:

$$d_A(A, A') = \max_{r \in R} \cos^{-1} \left( \frac{A_r \cdot A'_p(r)}{\|A_r\| \|A'_p(r)\|} \right)$$

(3.7)
where \( p^* \) is a permutation of the columns of \( A' \) that minimizes the average angle between the columns \( \theta_p \).

\[
\theta_p = \frac{1}{R} \sum_{r \in R} \cos^{-1} \left( \frac{A_r \cdot A_{p(r)}'}{\|A_r\| \|A_{p(r)}'\|} \right)
\]  

(3.8)

We can find \( p^* \) in polynomial time by solving the associated bipartite matching problem.

The largest value that this distance measurement can take is \( 90^\circ \). The criterion in equation 3.8 has several desirable properties. First, it is invariant to multiplication by a positive monomial matrix (i.e. \( d_A(A, APD) = 0 \)), where \( P \) is a permutation matrix and \( D \) is a positive diagonal matrix. Since the diagonal matrix \( D \) only rescales the columns of \( A \), we get that the angle measurements will remain unchanged.

The distance measure is also robust to many small perturbations. Unlike the Frobenius norm, the angle distance in equation 3.8 cannot be made large by simply perturbing all of the basis elements by a small amount. Taking the maximum of all the angles, rather than the sum, also prevents a large value that is simply the sum of many small angular perturbations. This property is helpful when columns of \( A \) are being interpreted (e.g. top ten lists in topic modeling (Arora et al., 2012c)). Knowing that a pair of columns has some minimum degree of separation gives us evidence that we may have found a qualitatively different solution.

In our experiments (Section 3.6), we choose one sample basis \( A_{ref} \) as our ‘reference example’ and compute \( d(A_{ref}, A) \) for each sample basis \( A \) in the chain. We experiment with choosing a known factor matrix or a random sample from our chain to be \( A_{ref} \) as well as generating a random matrix to serve as a reference. We look for the variance in the angle measurements with respect to a reference matrix over the samples in the chain. Higher variance suggests that the chain was able to explore a larger space.

### 3.6 Results

In the experiments below, we interleaved steps of the blocked Gibbs sampler of Schmidt et al. (2009) with different kinds of MH steps. In particular, we considered the following three scenarios:

- **Gibbs.** We ran the blocked Gibbs sampler of Schmidt et al. (2009) without any adjustments.

- **Gibbs with Random MH Proposals (RMH).** We interleaved the Gibbs sampling with MH steps in which both the basis matrix \( A \) and the weights \( W \) were drawn from their priors (exponential distributions with parameters \( \lambda_A \) and \( \lambda_W \)). Proposals from the prior are likely
to be very distinct from the current proposal, but unlikely to be also of high quality. We include this baseline to show that simply trying to insert moves far from the current location is not good enough to create a faster-mixing sampler.

- Gibbs with Extreme-Ray MH Proposals (ERMH). We interleaved the Gibbs sampling with our novel ERMH steps. These steps are designed to both move us far from our current location and likely to be of high quality.

For each of these approaches, we computed the running time, the integrated autocorrelation time, and the angle distance measure in Section 3.5 over multiple runs of the chain. The values for $A$ and $W$ were randomly initialized from the prior, and for our ERMH approach we took the extreme rays from a $K + 1$ rank factorization of the data $X$ using the (non-Bayesian) factorization algorithm of Lin (2007). A larger rank factorization was used so that the extreme rays could span a larger subspace.

### 3.6.1 Synthetic Data

Laurberg et al. (2008) gives an example of $6 \times 6$ data matrix that has two distinct, non-trivial NMFs (also shown in figure 3.2 that are 36 degrees apart by our angle measurement. We used this example to generate several $6 \times N$ data matrices by generating additional points in the convex hull of this toy data set for $N = [100, 500, 1000, 5000]$. Each of these data sets still had the same two distinct factorizations, but mixing would obviously be more challenging as the number of weights to be inferred increased. In the results below, we ran 10 independent chains of 10,000 iterations for each data set size $N$.

Figure 3.3 shows our angular distance metric (standard deviations) for these synthetic data as the number of observations is increased. Our ERMH approach has consistently higher angular distances between samples and a reference sample, and lower integrated autocorrelation time (See figure 3.4). In particular, our integrated autocorrelation time does not increase with the number of observations $N$ because we are still able to propose large moves in the lower dimensional space of extreme rays, which does not change in size as $N$ increases. Finally, figure 3.5 shows the running time per iteration of each of the algorithms (note that all are fast). Our ERMH approach requires less time than sampling proposals from the prior (RMH) because we operate in a lower dimensional space.
Figure 3.3: ERMH consistently outperforms the baselines in angular distance. Samples in the ERMH chains are significantly different in the angle sense. The difference seems to be maintained as \( N \) increases from 500 to 5000.

Figure 3.4: ERMH consistently outperforms the baselines on integrated autocorrelation time. In fact, we notice that as the number of observed data points increases, the Gibbs sampler and RMH tend to get stuck even more in the same optima therefore increasing their integrated autocorrelation time measurements. ERMH remains constant across \( N \).
3.6.2 Real Data

AML

The Leukemia dataset (from (Žitnik and Zupan, 2012)) was used in (Brunet et al., 2004) where NMF was performed for a classification task to distinguish between AML and ALL samples. The authors found that higher rank factorizations revealed sub-structure in the classes. We use our methods to illustrate that we can find factorizations that yield different block diagonal patterns in reordered consensus matrices. We run 10 independent chains of length 1000 for this dataset.

Figure 3.6 shows two distinct sub-clusterings that are obtained from factors in the ERMH chain. In particular, these two sub-clusterings arise from the weights matrix of our sample \( W_s \) and accepted proposal \( W_p \). This example also illustrates the dramatic changes that ERMH proposals involve.

BBC

We tested our proposed algorithm on a collection of 1162 BBC news articles (Greene and Cunningham, 2006) and were able to successfully obtain factorizations that gave rise to different top ten lists for the same topics. We run 5 independent chains of length 1000 for this data set.
Figure 3.6: We show the clusterings obtained from two distinct factorizations from our chain. The basis matrices are separated by 35 degree angle. We notice that the basic rank 2 classification stays the same between the two factorizations, but we get a different sub-clustering.

Figure 3.7 shows two different top-ten lists that we obtain. This type of analysis can be used to better understand the importance of key terms in a corpus.

**Faces**

The Olivetti (now AT&T) faces dataset consists of 400 images of 40 distinct subjects with 4096 grayscale pixel densities. NMF is a popular tool for feature extraction on such datasets. We obtained the data through Scikit-learn (Pedregosa et al., 2011) and applied our techniques, setting the rank to 4. We run 5 independent chains of length 1000 for this data set.

Figure 3.8 shows a histogram of the angle measurements made against a fixed matrix. We note that Gibbs and RMH appear to be in different modes with very little angle deviation across the
samples. ERMH samples show a lot of movement across angles. In Figure 3.9 we show two sample factorizations from the ERMH chain. We note that the factorizations correspond to different sets of faces.

![Figure 3.8: Comparing samples in our chains to a fixed randomly generated matrix to see how angles change across the chains. It appears that RMH and Gibbs have both found (slightly different) modes where they remain.](image)

![Figure 3.9: We show two distinct factorizations from our chain. These are separated by 19 degree angle and have been permuted so that they align according to minimum average angle. The image on the top-left is the one that corresponds to the maximum angle.](image)

**Real Data Summary**

In this Section we summarize the runtimes (Table 3.1), angle deviations (Table 3.2) and autocorrelations (Table 3.3) observed in our experiments on real data. We also provide 95% confidence intervals.
for these measurements. We note that ERMH and RMH take more time than Gibbs. Since it is very
difficult to explore multiple modes using Gibbs, we realize that the additional computational cost
we pay is at most a 5-fold increase in computation time for the examples explored here. We find that
ERMH is far superior than the baselines in terms of angle deviation measurements. We noted higher
autocorrelation values in ERMH (particularly for the faces data). This was not expected because fig-
ure 3.8 clearly indicates that ERMH is the only algorithm exploring the space of matrices of different
angles. In figure 3.10 we show the time evolution of a single entry of our factor matrix. The plots
shown clearly indicate better exploration by ERMH but the slow movement (possibly due to low
acceptance rate) indicates high autocorrelation at lag 1.

Table 3.1: Time per sample (in seconds)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AML (10^{-2})</th>
<th>BBC (10^{-1})</th>
<th>Faces (10^{-1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>0.66 (0.66, 0.67)</td>
<td>1.1 (1.1, 1.1)</td>
<td>0.57 (0.56, 0.57)</td>
</tr>
<tr>
<td>RMH</td>
<td>10 (10, 11)</td>
<td>3.3 (3.2, 3.4)</td>
<td>1.7 (1.6, 1.8)</td>
</tr>
<tr>
<td>ERMH</td>
<td>1.9 (1.8, 1.9)</td>
<td>5.4 (5.2, 5.6)</td>
<td>1.6 (1.5, 1.7)</td>
</tr>
</tbody>
</table>

Table 3.2: Maximum Angle Standard Deviation(in degrees)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AML</th>
<th>BBC</th>
<th>Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>0.6 (0.2, 1.0)</td>
<td>0.49 (0.48, 0.51)</td>
<td>1.09 (0.47, 1.71)</td>
</tr>
<tr>
<td>RMH</td>
<td>0.7 (0.2, 1.2)</td>
<td>0.49 (0.47, 0.51)</td>
<td>1.37 (0.66, 2.07)</td>
</tr>
<tr>
<td>ERMH</td>
<td>2.4 (2.1, 2.8)</td>
<td>3.28 (3.17, 3.40)</td>
<td>6.59 (5.78, 7.40)</td>
</tr>
</tbody>
</table>
Table 3.3: Autocorrelation at lag 1

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AML</th>
<th>BBC</th>
<th>Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>0.74</td>
<td>0.06</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>(0.73, 0.74)</td>
<td>(0.060, 0.061)</td>
<td>(0.19, 0.33)</td>
</tr>
<tr>
<td>RMH</td>
<td>0.73</td>
<td>0.06</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>(0.73, 0.74)</td>
<td>(0.06, 0.061)</td>
<td>(0.20, 0.34)</td>
</tr>
<tr>
<td>ERMH</td>
<td>0.75</td>
<td>0.12</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>(0.71, 0.78)</td>
<td>(0.119, 0.120)</td>
<td>(0.91, 0.93)</td>
</tr>
</tbody>
</table>

Figure 3.10: It appears that the Gibbs sampler entry is being perturbed around one place whereas the ERMH entry is showing clear upward and downward movements. These movements would show up as high autocorrelations.

3.7 Conclusion

In this Chapter, we presented a reformulation of the NMF problem that naturally resulted in novel, faster mixing MH sampling scheme. Specifically, our approach leverages the geometry of the NMF problem to propose large moves in the space of feasible NMF solutions. Interleaved with Gibbs sampling steps, there we show that our sampler finds multiple interesting modes on several interesting problems. Our approach complements current sampling-based inference approaches to NMF (e.g. (Schmidt et al., 2009; Moussaoui et al., 2006)).

3.7.1 Moves between Subspaces

An important and natural extension of this work would be to consider situations in which we move between subspaces. Specifically, we assumed that given a reference factorization \((A, W)\), the alter-
nate factorization would be of the form \((AQ_A, Q_W W)\)—which implies that the factorizations are in the same subspace. However, it is possible for there to exist solutions not in the same subspace. In practice, we found that our Gibbs sampler usually stayed inside the same subspace (perhaps another sign of slow mixing), but there is no guarantee that this fact will generally be true. We also mitigated this issue by using a higher-rank \(K + 1\) factorization to generate extreme rays in a larger subspace. More principled methods to propose moves across subspaces is an interesting direction for future work.

**Future Directions for Direct Optimization** For the sampler, we only took advantage of writing reformulating the proposed \(A'\) as \(AQ_A = AR_A Y_A\). However, we can similarly describe the space of non-negative transformations of the weight matrix \(W\) by considering the cone within which the rows of the \(Q_W\) matrix must live to get an expression \(W' = Y_W R_W W\) in which \(R_w\) contain the extreme rays of the weights \(W\) and the only constraint on the matrix \(Y_W \geq 0\). Then, the quest for alternative solutions \((A', W')\) reduces to finding \(Y_A \geq 0\) and \(Y_W \geq 0\) such that \(R_A Y_A Y_W R_W \approx I\); this space may be easier to directly search for alternative solutions rather than the original space of factors \(A\) and \(W\). Such solutions may make useful initializations for optimization-based approaches to inference such as variational methods (e.g. \((\text{Paisley et al.})\)).
Multiple Solutions in NMF: Variational Inference using RRTs

The presence of multiple distinct solutions to the NMF problem and the difficulties in discovering them led to the development of the algorithm in Chapter 3. Increasing computational cost of computing the extreme rays with dimensionality of the matrix data makes the approach unscalable and the difficulty in hyper-parameter tuning makes it challenging to apply to new datasets. In this Chapter, we present a variational inference approach that is non-parametric, allowing for the discovery of as many distinct modes as necessary to sufficiently cover the BNMF posterior. In addition, we employ a search strategy for discovering new, distinct factorizations using Rapidly exploring Random Trees (RRTs) in a latent space for transformations of NMF factors.

4.1 Problem Description

Non-negative Matrix Factorization (NMF) is a popular model for understanding structure in data, with applications such as understanding protein-protein interactions Greene et al. (2008), topic modeling Roberts et al. (2016), and discovering molecular pathways from genomic samples (Brunet et al., 2004). The goal is simple: given a $D \times N$ data matrix $X$ and desired rank $R$, the nonnegative matrix factorization (NMF) problem involves finding an $R \times N$ nonnegative weight matrix $W$, and an $D \times R$ nonnegative basis matrix $A$, such that $X \approx AW$. Applied work has benefited from the
myriad of efficient algorithms for solving different versions of the NMF objective (Paisley et al., 2015; Schmidt et al., 2009; Moussaoui et al., 2006; Lin, 2007; Lee and Seung, 2001; Recht et al., 2012).

However, in many cases NMF is not identifiable: there may be different pairs \((A, W)\) and \((A', W')\) that might explain the data \(X\) (almost) as well. Non-identifiability in NMF has been studied in detail in the theoretical literature (Pan and Doshi-Velez, 2016; Donoho and Stodden, 2003; Arora et al., 2012b; Ge and Zou, 2015b; Bhattacharyya et al., 2016), but it is of practical concern as well. For example, Greene et al. (2008) use ensembles of NMF solutions to model chemical interactions, while Roberts et al. (2016) conduct a detailed empirical study of multiple optima in the context of extracting topics from large corpora. Both works use random restarts to find multiple optima.

Bayesian approaches to NMF (Schmidt et al., 2009; Moussaoui et al., 2006) promise to characterize parameter uncertainty in a principled manner by solving for the posterior \(p(A, W|X)\) given priors \(p(A)\) and \(p(W)\). Having such a representation of uncertainty in the bases and weights can further assist with the proper interpretation of the factors: we may place more confidence in subspace directions with low uncertainty, while subspace directions with more uncertainty may require further exploration. Unfortunately, in practice, these uncertainty estimates are often of limited use: variational approaches (e.g. Paisley et al. (2015); Hoffman and Blei (2015)) typically underestimate uncertainty and fit to a single mode; sampling-based approaches (e.g. Schmidt et al. (2009); Moussaoui et al. (2006)) also rarely switch between multiple modes.

In this Chapter, we take steps to a more complete characterization of uncertainty in NMF. First, we use recent insights (Pan and Doshi-Velez, 2016) about how different NMF solutions of similar quality may be related to each other to limit the spaces that one must explore to find alternate solutions. Second, we use rapidly-exploring random trees (RRTs) with an online nonparametric variational inference framework to rapidly cover the space of probable solutions. The first part is specific to NMF; the second is broadly applicable to other statistical models as well. We demonstrate that our approach achieves not only better posterior coverage (as measured by the covering number metric of Masood et al. (2016a)), but often better ELBO values as well.

4.2 Background

Nonparametric Variational Inference (NVI) Variational methods approximate a desired distribution \(p(\theta|x)\) with a distribution \(q(\theta)\) by maximizing the evidence lower bound (equivalent to minimizing \(KL(q||p)\)):

\[
\mathcal{F}[q] = \mathcal{H}[q] + \mathbb{E}_q[\log p(\theta, x)].
\]
To model complex patterns in $p(\theta|x)$, Gershman et al. (2012) suggest using a mixture of uniformly weighted $M$ Gaussians with isotropic co-variance for the variational family $q(\theta)$:

$$
q(\theta) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(\theta; \mu_m, \sigma_m^2 I)
$$

The entropy term $\mathcal{H}[q]$ encourages diversity in the set of components. The expectation term $\mathbb{E}_q[\log p(\theta, x)]$ encourages the components to be placed in regions of high joint likelihood. This can be viewed as a quality component of the ELBO. The authors note that the entropy term $\mathcal{H}[q]$ in equation 4.1 can be lower bounded by

$$
\mathcal{H}[q] \geq -\frac{1}{M} \sum_{m=1}^{M} \log \left( \frac{1}{M} \sum_{j=1}^{M} \mathcal{N}(\mu_m; \mu_j, (\sigma_m^2 + \sigma_j^2) I) \right)
$$

and provide a second-order approximation for the likelihood term $\mathbb{E}_q[\log p(\theta, x)]$, where we use $f(\theta) = \log p(\theta, x)$:

$$
\mathbb{E}_q[f(\theta)] \approx \frac{1}{M} \sum_{m=1}^{M} f(\mu_m) + \frac{\sigma_m^2}{2} \text{Tr}(H_m)
$$

where $H_m$ is the Hessian $\nabla^2_\theta f(\theta)$. While many other approaches could be used to optimize this term (e.g. BBVI Ranganath et al. (2014)), we will use the approximation from equation 4.3 as it permits certain aspects of our optimization to be computed analytically.

Rapidly-Exploring Random Trees    Rapidly-Exploring Random Trees (RRTs) and their variants have been extremely successful for solving high-dimensional path-planning problems in robotics and other domains LaValle (1998). RRTs take as input a configuration space $C$—for example, all possible parameter settings—which may have some obstacles $O \subset C$ or invalid parameter settings. Given a metric $\rho(c, c')$ on the configuration space $C$, a method for uniformly sampling within $C$, a means to determine if a specific point $c \in O$, and a current set of nodes $\{c\}$ is in the tree, the RRT rapidly expands across the free space via the following algorithm:

1. Generate a sample $c^* \in C$ uniformly
2. Find the node $c \in \{c\}$ that is nearest to $c^*$ according to the metric $\rho(c, c^*)$. 


3. Add the node $c'$ that is obtained by moving a step-size $\epsilon$ from $c$ in the direction of $c^*$ if $c' \notin \mathcal{O}$.

This procedure implicitly expands nodes in proportion to the volume of the Voronoi region associated with each node in the configuration space $\mathcal{C}$, making the RRT seek new regions. In the context of NMF posterior characterization, we will use geometric insights to first reduce the size of the configuration space to be explored, and then apply RRTs to rapidly explore this reduced space. The resulting samples will be used as centers for a NVI approximation of the posterior.

**Covering Number as a Metric for Posterior Coverage**

Even for relatively small amounts of data (e.g. 100 observations), sampling-based procedures for Bayesian NMF can be mixed within a mode—as measured by traditional mixing metrics such as autocorrelations times—but slow to explore even a single mode. What is missing is a notion of coverage: In addition to moving in “independent” ways, how much of the posterior space does a finite-length chain explore? Masood et al. (2016a) advocate for the use of the minimum covering number to quantify how much of the posterior space a finite-length chain explores. Given a similarity measure on the parameter space, the **covering number** is the minimum number of $\epsilon$ balls needed to cover all the samples. We approximate the covering number by limiting our centers to existing samples and greedily adding centers until all the points are covered. Larger covering numbers suggest larger exploration of the posterior space. Finally, a **persistence plot** shows how the covering number changes with $\epsilon$; for points that are spread far apart, the covering number will remain large for many $\epsilon$.

For the similarity measure, we use weighted angular distance (WAD; an extension of the maximum angle similarity proposed in Masood et al. (2016a)). To compute the WAD, we first compute an angle difference between columns of the basis matrix after adjusting for the permutation ambiguity in NMF. Next we consider the contribution of each vector in the cone in reconstructing the data (weights). Given two pairs of factorizations, $(A, W)$ and $(A', W')$, let $\hat{\rho}$ be a permutation of the columns of $A'$ that minimizes the average angle between corresponding columns,

$$\hat{\rho} = \arg \min_{\rho \in S_R} \frac{1}{R} \sum_{r \in R} \cos^{-1} \left( \frac{A_r \cdot A'_{\rho(r)}}{\|A_r\| \|A'_{\rho(r)}\|} \right)$$

Let $\alpha$ be the vector of angle differences between corresponding columns of $A$ and $A'$. In order for the measurement to be consistent across different scalings of the factorizations, we fix the basis matrices to have the scaling that makes them column stochastic and now consider the corresponding
weights matrices $W$ and $W'$. We sum the rows of the two weights matrices and normalize them to add to 1. We denote these weights vectors as $w$ and $w'$. The WAD is then given by

$$WAD(A, W, A', W') = \alpha^r \frac{(w + w')}{2}.$$ 

The WAD is bounded in the interval $[0, 90]$ degrees since all the columns of $A$ and $A'$ lie in the positive orthant. WAD is also invariant to the scaling and permutation ambiguity in NMF.

4.3 Methods

We now detail our approach, which leverages geometric insights from the NMF problem to use RRTs to rapidly explore nodes to be incorporated into an online NVI framework.

4.3.1 Online Nonparametric Variational Inference

This work is built upon the work in (Gershman et al., 2012). We use the a more general variational family by allowing weights $w_m$ to be non-uniform. We also modify the algorithm to an on-line one which can be used to decide whether a new mixture component should be included in the variational distribution or not. Our Online-NVI (ONVI) takes in a candidate component center $\mu_{m+1}$ and does the following:

1. Find the optimal member of the variational family with the added component (optimizing for $\sigma^2_{m+1}, w_{m+1}$).
2. Decide whether to add the new component.
3. If we accept a new component, decide whether to remove previous components.

In particular, for the first step, we assume that the previous centers $\mu_1, \ldots, \mu_m$ and variances $\sigma^2_1, \ldots, \sigma^2_m$ stay fixed, and that previous weights $w_1, \ldots, w_m$ are simply scaled with the addition of a new weight $w_{m+1}$. This restriction is reasonable because of mode-hugging property of variational inference: each component will want to find a mode and will tend to under-estimate the variance around that mode; thus, it is reasonable to expect that adding a new component will not cause global changes to the NVI solution.

For the second and third steps, we note that adding new components can only make the ELBO increase as we have just created a more flexible variational family—if the proposed candidate mean
\( \mu_{m+1} \) is of low quality, we will simply set its weight \( w_{m+1} \) to zero. Similarly, if a new component is added with significantly higher quality than the existing components, then it is possible that all previous weights \( w_1, \ldots, w_m \) will be scaled down to close to zero. For both cases, we use a minimum change criterion: a minimum ELBO improvement criterion to accept a new component, and a minimum ELBO loss to reject an existing component.

### 4.3.2 NMF exploration using RRTs

**Framework** While there exist many RRT variations, we use the following framework:

1. **Base Nodes**: A set of nodes that always remain part of the RRT.
2. **Temporary Nodes**: A set of nodes temporarily used in the expansion of the RRT. They are deleted once a certain expansion criteria is met to avoid slowing down future nearest neighbor searches.
3. **Feasibility Test**: A way of checking if a proposed node is in our configuration space (that is, our region of interest).
4. **Stepping Method**: A way to move from one node in the direction of another.

**Configuration Space** The space of all the parameters needed for \( A \) and \( W \) is large—of dimension \( R(D + N) \). In this Section, we use the geometry of the NMF problem to limit the space in which our RRT will have to search. Let \( A_{\text{SVD}} \in \mathbb{R}^{D \times R}, W_{\text{SVD}} \in \mathbb{R}^{R \times N} \) be the SVD of \( X \). The SVD factorization is exact when the data \( X \) has rank \( R \). We note that when the rank \( R \) of a matrix is equal to its positive rank \( R^+ \), Laurberg et al. (2008) show that two exact factorizations can only differ by a subspace transformation \( Q \) of the form

\[
X = AW = (AQ)(Q^{-1}W) = A_{\text{SVD}}W_{\text{SVD}}
\]

where the last equality must be true because \( A_{\text{SVD}} \) defines the unique subspace in which the data live. That is, the NMF problem can be viewed as a change of basis from the subspace spanned by the SVD factors. While the unique-subspace assumption is not necessarily true for the approximate case, it does give us a way to rapidly explore the best subspace of rank \( R \). \( Q \) has only \( R^2 \) parameters (we note that (Ročková and George, 2015) use a similar insight within an auxiliary variable approach to find sparse matrix factorizations).
Two concerns remain: First, the space of $Q$, while smaller, still contains many trivial transformations: permutations and changes of scale. Second, the change of basis $A_{\text{SVD}}Q$ and new weights $Q^{-1}W_{\text{SVD}}$ may not produce nonnegative factors. To address the first concern, we limit our change of basis matrices $Q$ to the oblique manifold which reduces the redundancies from infinite (permutations and scalings) to a finite set of permutations. A rank-$R$ oblique manifold is defined as

$$\mathcal{O}(R) \overset{\text{def}}{=} \{ Q \in \mathbb{R}^{R \times R} : \text{diag}(Q^\top Q) = I_R, \det(Q) \neq 0 \}.$$ 

In words, $\mathcal{O}(R)$ is the space of invertible $R \times R$ matrices with columns that are unit Euclidean norm. In fact, the oblique manifold, $\mathcal{O}(R)$ can be treated as the product of $R$ unit-spheres in $\mathbb{R}^R$; the only difference being that these spheres should all be linearly independent in the Oblique manifold. The only remaining NMF inherent ambiguity is that of the permutation of columns but that is only finite. For a rank $R$ problem, given a $Q$, we have $R!$ equivalent factorizations that can be obtained through considering different permutations of the columns of the matrix $Q$—a measure zero set of points.

Next, we address the fact that not all matrices $Q$ will produce a nonnegative solution $A_{\text{SVD}}Q$ and $Q^{-1}W_{\text{SVD}}$. We project the products to positive values. The $\lfloor \rfloor$ operation stands for setting negative values to zero:

$$A = \lfloor A_{\text{SVD}}Q \rfloor \quad W = \lfloor Q^{-1}W_{\text{SVD}} \rfloor$$

(Note that in the case of exact NMF, there exists a $Q$ for which $A = A_{\text{SVD}}Q$ and $W = Q^{-1}W_{\text{SVD}}$, where there are no negative values in the products so $A, W$.)

**Initial Base Nodes**  We initialize our RRT with a set of base nodes $Q_m$ corresponding to each random restart $(A_m, W_m)$ as the change of basis from the SVD: $Q_m = \arg\min_Q \| A_m - A_{\text{SVD}}Q \|$. As the tree is expanded, we only retain nodes that are sufficiently far from existing nodes; these criteria are detailed in Section 4.4.3.

**Feasible Regions**  We compute feasible regions by thresholding based on the joint probability of data and factorizations. We also adaptively adjust the feasibility criteria via a minimum angle threshold in order to avoid finding new points that are too close to existing ones.

**Stepping Method**  Movement from one node $Q_1$ to another $Q_2$ is made by stepping in the direction $Q_d = Q_2 - Q_1$. Each step consists of moving along the tangent space and then retracting
the result onto the oblique manifold (e.g. as in (Absil et al., 2009)). Following the RRT-Extend algorithm of Kuffner and LaValle (2000), we continue until the feasibility criteria fails or a maximum number of extended steps are tried.

Aside: Scale Optimization for Gaussian Likelihoods While the likelihood may be invariant to the scale of $A$ and $W$, the Hessian term in equation 4.3 is not—intuitively, the Hessian term gives local curvature information about the posterior and encourages placing components in regions of high probability volume. For different scales of $A$ and $W$, the isotropic variance $\sigma_m^2$ may result in different relative volumes covered.

The RRT nodes have much little scale flexibility as the entries of the SVD factors are fixed and for every change of basis matrix $Q$ in the oblique manifold, the unit norm columns of the $Q$ matrix determine a scaling as well. Fortunately, we can easily rescale the RRT components to cover an approximate volume.* We use the analytical Hessian approximation for the squared error of the factorization given in Lin (2007). Given a component $(A_m, W_m)$, under the Gaussian Likelihood with variance in the data $X$ given by $\sigma_X^2$, the Hessian term is given by

$$\text{Tr}(H_m) = -\frac{1}{\sigma_X^2} \left( N \times \text{Tr}(A^T A) + D \times \text{Tr}(WW^T) \right)$$

This term depends on the $l_2$-norm of the columns of $A$ and the rows of $W$. Let $S$ be the diagonal matrix that scales the basis matrix to unit column norm so that $\text{Tr}((AS)^T(AS)) = K$. We restrict our optimization to the components where elements of the basis matrix have the same squared column sum (so that column scales are consistent with each other). Each component of this form looks like $(\beta A_m S, \frac{1}{\beta} S^{-1} W_m)$. The Hessian term depends on $\beta$ in the following way:

$$\text{Tr}(H_m)(\beta) = -\frac{1}{\sigma_X^2} \left( \beta NK + \frac{D}{\beta} \text{Tr}((S^{-1}W)(S^{-1}W)^T) \right)$$

We use second order Newton Conjugate Gradient descent to perform an optimization to find $\beta > 0$ that maximizes the Hessian term.

---

*The scale of $A$ and $W$ affects the prior term as well. Empirically, we observed that this rescaling affected the Hessian term more than the prior term, although this may not always be the case.
4.4 Experimental Approach

4.4.1 Problem Formulation: Bayesian NMF

In our experiments, we consider exponential priors for the basis and weights matrices: \( p(A) = \prod_{d,k} \text{Exp}(A_{d,k}; \lambda_{A_{d,k}}) \) and \( p(W) = \prod_{k,n} \text{Exp}(W_{k,n}; \lambda_{W_{k,n}}) \). We consider two different likelihood models:

**Gaussian** The first is a Gaussian model in which Gaussian noise is added to each dimension IID:

\[
p_N(X|A, W) = \prod_{d,n} N(X_{d,n}, (AW)_{d,n}, \sigma_X^2)
\]

As derived in Schmidt et al. (2009), the combination of exponential priors and Gaussian likelihoods results in a relatively straightforward Gibbs sampling update.

**Uniform** The light tails of the Gaussian distribution imply that the samples far from the mode are quickly discounted. In many applications, a Uniform noise model may be more appropriate, especially if the model is misspecified and we desire factorizations that “roughly” model the data (that is, the difference between a perfect factorization and one with some noise may not matter):

\[
p_U(X|A, W) = \prod_{d,n} U(-\epsilon, \epsilon)(X_{d,n} - (AW)_{d,n})
\]

4.4.2 Baselines

We compare to the following baselines:

1. Standard Nonparametric Variational Inference (NVI) as described in Gershman et al. (2012) with \( M = 4 \) and \( M = 10 \) components.

2. HMC + ONVI (HMC): We run Hamiltonian Monte Carlo with adaptive step size (Neal et al., 2011) for 10,000 samples; each sample is proposed to the ONVI.

3. Gibbs + ONVI (Gibbs): For the Gaussian noise model only, we run the Gibbs sampler of Schmidt et al. (2009) for 10,000 samples and propose each sample to the ONVI. (There is no conjugate Gibbs sampler for the Uniform likelihood.)
The NVI components are initialized with a random solutions. In the Uniform likelihood case, we initialize to a single feasible solution as there is no gradient information otherwise. The Gibbs and HMC chains are initialized via first running Lin’s algorithm (Lin, 2007) on a random starting point to avoid the need for burn-in period. We repeat each experiment 10 times and report the mean as well as 25th and 75th percentile of our obtained statistics.

4.4.3 Implementation Details

ONVI Implementation  We use second order Newton Conjugate Gradient descent to perform the optimization. Gradients and Hessian-vector products of our objective are computed using Autograd (Maclaurin et al.).

For the Gaussian likelihood, we use the second order approximation for the likelihood term (equation 4.3) to optimize both the variance $\sigma_m^2$ of the new component as well as its weight $w_m$. Following optimization, if the new component improves the ELBO by less than the same absolute change stopping criteria of $10e^{-4}$ from Gershman et al. (2012), we reject that component. Similarly, if removing the previous components decreases the ELBO by less than the absolute change criteria, we remove them.

The Uniform likelihood is flat within its feasible region and the exponential prior also has no second order information. Thus, we can no longer rely on the Hessian term in equation 4.3 to control the variance $\sigma_m^2$. Instead, we fix the variance $\sigma_m^2$ to the variance found by running standard variational inference (via NVI with M=1) and optimize over the weights $w_m$. We also use a smarter stopping criteria corresponding to the increase in entropy by a prospective component perturbed by $\sigma_m$ in every dimension from an existing component.

RRT Implementation  The framework of the RRT is very flexible and we customize it depending on the likelihood model. Below we describe the specifics of our ‘Base Nodes’, ‘Temporary Nodes’ and feasibility criteria.

In Gaussian likelihood model the quality term dominates in the ELBO, so we design the RRT to look for better quality components than already present in its set of nodes. We initialize the RRT with 50 solutions of Lin’s algorithm. We also pick one of these and optimize it under the ELBO approximation of equation 4.3. We set the maximum number of ‘Temporary Nodes’ to 100 and allow the RRT to expand until it reaches that maximum. For subsequent feasible components the RRT finds, we only replace them with the lowest quality node in the tree. We initially set the feasibility
threshold based on the poorest quality node in the tree. Once the maximum number of ‘Temporary Nodes’ is reached, we increase the threshold of the feasible region to be equal to that of the highest quality component. There are no fixed ‘Base Nodes.’

In the Uniform likelihood model, the entropy term dominates in the ELBO, so we design the RRT to look for components more diverse than those in its existing set of nodes. We fix ten ‘Base Nodes’ from random restarts of Lin’s algorithm and use these to repeatedly grow the RRT. For feasibility, we set the minimum angle condition to 0.01 degrees initially. We allow for 90 ‘Temporary Nodes’ for the RRT’s expansion. Once the maximum number of nodes is reached, we re-start the RRT with the ‘Base Nodes’ and increase the minimum angle criteria by 0.5. For any component that gets added to the ONVI, we add the corresponding node to our set of ‘Base Nodes’ as well.

We set the minimum step-size of the RRT in the oblique manifold to be $s_0 = 0.01$. This step-size grows by 10% (for a maximum of 50 steps) during the expansion of the RRT as long as we are in a feasible region. In both noise models, we terminate the RRT+ONVI algorithm once the ONVI has processed 5000 components or if the RRT fails to find a new feasible node after 10,000 attempts at expansion.

4.5 Results

4.5.1 Demonstration on Synthetic Data

We embed a toy data matrix of rank 3 known to have two exact NMFs (Laurberg et al., 2008) into $\mathbb{R}^{500 \times 500}$ and add Uniform noise. This toy data set has an interesting property that only slightly changing the data changes the nature of solutions from two to infinite. Thus, we expect that adding the noise will result in multiple different solutions to exist within a certain threshold. We make the job of the RRT more difficult by only initializing it at one (known) analytical solution.

Figures 4.1, 4.2, 4.3 show a two dimensional projection of the rank-3 factorizations found by NVI ($M = 10$), ONVI+HMC, and ONVI+RRT. Even when there are many factorizations, both NVI and HMC have very limited exploration in comparison to the RRT. The persistence plots in figure 4.4 echo this property: HMC and NVI lines fall to 1 at an angle less than 0.01 while the RRT persists for much longer.
Figure 4.1: A 2-D projection of Laurberg Data and NVI components. The different components are not distinguishable.

Figure 4.2: A 2-D projection of Laurberg Data and the components accepted in the HMC + ONVI algorithm. These components are all coming from the same mode, which overlaps with the analytical solution (initialization).

Figure 4.3: A 2-D projection of Laurberg Data and the components accepted in the RRT + ONVI algorithm. These components are distinct from the analytical solution (initialization). These diverse components continue to explain the data under the Uniform noise model as the data lies in the convex hull of these RRT-based factorizations.

Figure 4.4: Laurberg Data: A persistence plot of samples from the variational distribution obtained from the different algorithms. The RRT + ONVI is the only algorithm for which minimum cover numbers are larger than one for angles greater than 0.01. The other algorithms are unable to escape the modes that they were initialized in.

4.5.2 Application to Real Data Sets

We apply our approach to several datasets commonly used for NMF: 20-Newsgroups (Mitchell, 1997) (D=813, N=2034, categories = R = 4); Reuters articles (Lewis, 1987) (D = 1540, N = 2362, categories = R = 4); BBC articles (Greene and Cunningham, 2006) (D = 6045, N = 1162, categories = R = 4); AML/ALL cancer cell data (Žitnik and Zupan, 2012) (D = 5000, N = 38, R = 3); Olivetti Faces

†In the Reuters, and 20-Newsgroups data sets, we only took documents from the top 4 categories.
‡The AML/ALL dataset actually has two categories. We chose the factorization rank to be 3 because it reveals useful sub-structure in the data (Brunet et al., 2004).
RRT+ONVI give comparable or better ELBO values. Tables 4.1 and 4.2 show the ELBO values for our approach compared to the baselines with the noise $\sigma^2_X$ set to the empirical noise (found by first fitting a solution via Lin’s algorithm and measuring the squared error, before any additional experiments) and 10x this empirical noise. The latter case was designed to simulate a situation where we might expect to see greater variation in solutions.

In both cases, the RRT+ONVI has the highest ELBO across all data sets. However, due to the light tails of the Gaussian distribution, the expectation term (equation 4.3) dominated the (coverage-encouraging) entropy term (equation 4.2); all the ONVI variants consistently retained only one highest quality component.

Table 4.3 shows the results for the Uniform likelihood (similar to before, the noise $\epsilon$ was set based on the absolute deviation from fitting a solution via Lin’s algorithm). Here, the differences between solutions are smaller because they come primarily from differences in the entropy term; thus we show the deviation from the mean ELBO for each data set. As before, the RRT+ONVI finds the variational distribution with the highest ELBO. The HMC+ONVI fails to move significantly from its starting point and thus only retains one component; in contrast the RRT+ONVI retains between 2 (AML) and 118 (BBC) components depending on the data set.

RRT+ONVI has more posterior coverage. The evaluations above showed that our RRT+ONVI approach achieves comparable or better ELBO values on real datasets. Now we return to our goal: posterior coverage. Figures 4.5, 4.6, 4.7, 4.8, 4.9, and 4.10 plot the minimum covering number of the components of the variational distribution under the WAD similarity measure. While there is a range of variation across the data sets in terms of angle, the RRT components are most distinctly placed.

---

*Figures in Hubble dataset were down-sampled from $N = 16384$)
In particular, the persistence plots that for datasets such as 20-Newsgroups and BBC have a lot of variation in the number of modes found and some variation in how the persistence curve drops to 1 as angle gets larger. This variation is indicative of the random nature of the RRT across different repetitions of the experiment. In the remaining datasets we see much less variation. It appears that the ONVI has consistently accepted components from the random restarts of Lin’s algorithm (in the initialization of the RRT). In the case of AML and Faces datasets, the RRT finds multiple components for the ONVI to consider but the ONVI rejects them based on the entropy-based ELBO improvement criteria. In the Reuters and Hubble datasets, the RRT fails to expand from the initial base nodes.

4.6 Discussion

Our rapid posterior exploration approach involved making many choices, ranging from the approximating variational family to the exploration properties of the RRT. First and foremost, we observed that the choice of the likelihood model plays a crucial role in determining the shape of the posterior—obvious, perhaps, but not necessarily observable if one’s algorithms limit the posterior approximation to small regions. The noise parameters can also have large effects. For consistency, in our experiments we set \( \epsilon \) in the Uniform likelihood as the largest absolute deviation between the data and NMF approximation based on ten random restarts of Lin’s algorithm. However, in other experiments, we found increasing \( \epsilon \)—to \( \epsilon_{\text{new}} = 2 \epsilon \) for Faces, \( \epsilon_{\text{new}} = 8 \epsilon \) for Hubble—resulted in many more components in the posterior constructed by ONVI. It may be more appropriate to specify \( \epsilon_{i,j} \) for each entry \( i, j \) in the data \( X \).

The isotropic variance of our variational family is restrictive. For high-density regions of the Bayesian NMF posterior, we can expect a perturbation of the basis matrix to be correlated to an appropriate change in the weights matrix. A mixture of Gaussians with more general covariances might therefore be most appropriate. For such a model, Huber et al. (2008) describe a generalization of the lower bound of the entropy formula that we used in equation 4.2. The question of determining the entries of the covariance matrices would need some insight. Simply trying to optimize for the entries in the covariance matrix would make the problem high-dimensional again as each step in the ONVI would involve solving for a covariance matrix of a \( R(D + N) \)—dimensional vector.
Figure 4.5: 20-Newsgroups data: Persistence plot of the components shows that RRT + ONVI components are more spread out than baselines.

Figure 4.6: BBC data: Persistence plot of the components shows that RRT + ONVI components are more spread out than baselines.

Figure 4.7: Reuters data: Persistence plot of the components shows that RRT + ONVI components are more spread out than baselines.

Figure 4.8: AML/ALL data: The RRT + ONVI components are more spread out than baselines but it is a much smaller spread in terms of angle. For angles greater than 0.1 the minimum covering number is one for all algorithms.

Figure 4.9: Faces data: The RRT + ONVI components are spread out more than baselines.

Figure 4.10: Hubble data: The RRT + ONVI components show the largest posterior coverage.
The ONVI serves to accept/reject candidate components and to adjust accepted components within the variational posterior in an optimal manner. Before introducing candidate components to the ONVI, one might also imagine using a diversification strategy such as hard-core processes or determinantal point processes Kulesza and Taskar (2012) to cull the set of candidate nodes. Such a step could help eliminate redundant candidate nodes and reduce the number of candidate nodes presented to the ONVI.

Lastly, there were many parameter choices made in the initialization and expansion of the RRT. Changing the RRT parameters can make a difference in the posterior constructed by the ONVI. For example, the RRT fails to expand in the Reuters dataset (under the Uniform likelihood) using our default parameters, but we noticed that increasing the initial step-size from $s_0 = 0.01$ to $s_0 = 0.5$ allows the RRT to expand and contribute new components to the posterior constructed by the ONVI. An interesting future direction is to adapt the RRT to specific noise-models and datasets. To allow for a larger number of temporary and base nodes (to aid the RRT expansion) we suggest using more efficient data structures for finding the nearest point in the tree (Atramentov and LaValle, 2002). For exploring posterior regions that are not conducive to RRT expansion, alternative exploratory techniques such as probabilistic road maps (Kavraki et al., 1996) can be employed.

4.7 Related Work

Outside of the empirical work of Greene et al. (2008) and Roberts et al. (2016), we are not aware of work that has been done to to explicitly consider multiple modes (or connected posterior regions) in NMF. In the Bayesian setting, there have been several general approaches to encourage samplers to explore and mix between modes. Neal et al. (2011) discusses a specific approach to tempering in HMC; Green and Mira (2001) discusses an MH approach that delays rejections to allow a far-reaching proposal to find an alternate mode. Sminchisescu et al. (2007) describe approaches for hopping between modes if the locations of the modes are known to the sampler in advance. Both simulated (Marinari and Parisi, 1992) and parallel (Hansmann, 1997) tempering methods are popular for encouraging samplers to move to new regions of high posterior density. Wang-Landau sampling (Schulz et al., 2003) is a popular MCMC sampling method in statistical physics that involves a random walk in the energy space rather than sampling at a fixed temperature.

Within variational methods, mean-field approaches (Beal, 2003; Jordan et al., 1999) are popular but limited to approximating posteriors with unimodal distributions. The use of mixture distributions in mean-field approaches (Lawrence and Jordan, 1998; Jaakkola and Jordan, 1998; Bouchard
and Zoeter, 2009) adds flexibility to the posterior but the parameter updates are model dependent and may be difficult to derive. We choose a Gaussian mixture with isotropic variance for our variational family. This is a special case of the mixture mean-field approach for which the inference algorithm provided by (Gershman et al., 2012) can be used for a more general class of models as only knowing the joint likelihood and its gradient is required. Flow-based methods of creating flexible posterior families (such as Non-linear Independent Components Estimation (Dinh et al., 2014) and Hamiltonian variational approximation (Salimans et al., 2015) use a series of functions that transform the parameter space. The normalizing flows framework unifies and presents a generalization of these flow-based variational methods for flexible posteriors (Rezende and Mohamed, 2015).

More broadly, the problem of locating multiple modes or a frontier of solutions has been studied in the optimization literature, though not for the particular case of NMF. Many of these approaches are particle or swarm approaches, in which multiple solutions are initialized, adjusted, and killed according to some evolution and fitness function (e.g. Dorsey and Mayer (1995); Brits et al. (2007). In some cases, explicit rules are made for finding solutions that are in alternate modes (Wales and Doye, 1997). Homotopy methods are used to explore the Pareto surface of nonlinear functions (Das and Dennis, 1998).

In some work, NMF is specifically used for clustering data. Greene et al. (2008) and Huang et al. (2011) use ensemble NMF techniques to understand the space of clusterings obtained through distinct NMF solutions (obtained from random restarts of NMF algorithms). More work exists for finding multiple alternate clusterings, which can be viewed as a subset of matrix factorizations in which each observation is associated with only one latent feature. Niu et al. (2010) iteratively find multiple clustering views by taking advantage of relationships between the spectral clustering objective and the Hilbert-Schmidt Independence Criterion. Qi and Davidson (2009) apply constrained optimization that defines trade-offs between alternativeness and clustering quality, also separate into finding a single alternative and multiple alternatives, while Gondek and Hofmann (2007) use a mutual information-based criterion to subtract out the information from existing alternatives to propose new ones. Finally, Grimmer and King (2011) simply run a large number of different clustering algorithms and display the alternatives.

To our knowledge, our application of explicit exploration algorithms for posterior coverage is novel.
4.8 Conclusion

In this Chapter, we leveraged some key geometric insights about NMF to first create a smaller search space, applied RRTs to explore this space, and incorporated these nodes into a flexible, more complete posterior via the nonparametric variational inference framework. Importantly, only the design of the RRT using the oblique manifold is specific to NMF: we expect that this notion of casting posterior inference as an explicit exploration problem will be fruitful for many other models as well.
### Table 4.1: Empirical Gaussian Noise Setting
We show values of the mean ELBO as well as the 25th and 75th percentiles. Across the datasets, the RRT + ONVI procedure finds the best variational fit to the true posterior.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NVI-4</th>
<th>NVI-10</th>
<th>Gibbs</th>
<th>HMC</th>
<th>RRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Newsgroups</td>
<td>3.115e+06</td>
<td>3.116e+06</td>
<td>3.145e+06</td>
<td>3.151e+06</td>
<td>3.191e+06</td>
</tr>
<tr>
<td>2.030e+07</td>
<td>2.030e+07</td>
<td>2.030e+07</td>
<td>2.030e+07</td>
<td>2.030e+07</td>
<td>2.061e+07</td>
</tr>
<tr>
<td>BBC</td>
<td>8.266e+06</td>
<td>8.266e+06</td>
<td>8.239e+06</td>
<td>8.232e+06</td>
<td>8.276e+06</td>
</tr>
<tr>
<td>8.260e+06</td>
<td>8.260e+06</td>
<td>8.230e+06</td>
<td>8.230e+06</td>
<td>8.230e+06</td>
<td>8.230e+06</td>
</tr>
<tr>
<td>Reuters</td>
<td>-1.536e+06</td>
<td>-1.536e+06</td>
<td>-1.571e+06</td>
<td>-1.564e+06</td>
<td>-1.510e+06</td>
</tr>
<tr>
<td>1.292e+06</td>
<td>1.292e+06</td>
<td>1.284e+06</td>
<td>1.284e+06</td>
<td>1.284e+06</td>
<td>1.314e+06</td>
</tr>
<tr>
<td>Faces</td>
<td>-2.593e+07</td>
<td>-8.079e+06</td>
<td>-8.022e+06</td>
<td>-8.179e+07</td>
<td>1.975e+06</td>
</tr>
<tr>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
</tr>
<tr>
<td>Hubble</td>
<td>7.551e+05</td>
<td>9.657e+05</td>
<td>1.126e+06</td>
<td>6.692e+05</td>
<td>1.208e+06</td>
</tr>
<tr>
<td>8.755e+05</td>
<td>9.347e+05</td>
<td>9.347e+05</td>
<td>9.347e+05</td>
<td>9.447e+05</td>
<td>9.447e+05</td>
</tr>
</tbody>
</table>

### Table 4.2: Larger Gaussian Noise Setting
We show values of the mean ELBO as well as the 25th and 75th percentiles. Across the datasets, the RRT + ONVI procedure finds the best variational fit to the true posterior.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NVI-4</th>
<th>NVI-10</th>
<th>Gibbs</th>
<th>HMC</th>
<th>RRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Newsgroups</td>
<td>2.298e+05</td>
<td>2.298e+05</td>
<td>1.968e+05</td>
<td>2.050e+05</td>
<td>2.320e+05</td>
</tr>
<tr>
<td>7.709e+06</td>
<td>7.709e+06</td>
<td>7.752e+06</td>
<td>7.762e+06</td>
<td>7.760e+06</td>
<td>7.844e+06</td>
</tr>
<tr>
<td>BBC</td>
<td>1.767e+06</td>
<td>1.767e+06</td>
<td>1.723e+06</td>
<td>1.730e+06</td>
<td>1.768e+06</td>
</tr>
<tr>
<td>1.767e+06</td>
<td>1.767e+06</td>
<td>1.767e+06</td>
<td>1.767e+06</td>
<td>1.767e+06</td>
<td>1.767e+06</td>
</tr>
<tr>
<td>Reuters</td>
<td>-1.842e+06</td>
<td>-1.842e+06</td>
<td>-1.858e+06</td>
<td>-1.873e+06</td>
<td>-1.818e+06</td>
</tr>
<tr>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
<td>1.988e+05</td>
</tr>
<tr>
<td>AML</td>
<td>-1.644e+06</td>
<td>-1.644e+06</td>
<td>-1.647e+06</td>
<td>-1.643e+06</td>
<td>-1.631e+06</td>
</tr>
<tr>
<td>7.551e+05</td>
<td>9.657e+05</td>
<td>1.126e+06</td>
<td>6.692e+05</td>
<td>1.208e+06</td>
<td></td>
</tr>
<tr>
<td>8.755e+05</td>
<td>9.347e+05</td>
<td>9.347e+05</td>
<td>9.347e+05</td>
<td>9.447e+05</td>
<td>9.447e+05</td>
</tr>
</tbody>
</table>
Table 4.3: Uniform Likelihood: This table shows ELBO deviation from dataset mean and number of RRT components accepted into the ONVI. In all cases, the HMC + ONVI led to unimodal variational distributions. We see that the range of components found are data-set dependent. We provide the mean, first quartile (25) and third quartile (75).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NVI 4</th>
<th>NVI 10</th>
<th>HMC</th>
<th>RRT</th>
<th>Components (RRT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Newsgroups</td>
<td>-3.080e-01</td>
<td>-1.228e+06</td>
<td>-1.499e+00</td>
<td>1.693e+00</td>
<td>14.8</td>
</tr>
<tr>
<td></td>
<td>-3.70e-01,</td>
<td>-3.28e-01,</td>
<td>-3.70e-01,</td>
<td>1.693e+00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.00e-02</td>
<td>3.28e-02</td>
<td>3.70e-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBC</td>
<td>-2.780e-01</td>
<td>-3.481e+06</td>
<td>-3.109e+00</td>
<td>3.040e+00</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>-3.50e-01,</td>
<td>-4.40e-01,</td>
<td>-3.10e-01,</td>
<td>3.040e+00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.07e-01</td>
<td>4.40e-01</td>
<td>3.10e-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuters</td>
<td>-1.840e-01</td>
<td>-2.079e+06</td>
<td>-7.622e-01</td>
<td>1.030e+00</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>-7.65e-01,</td>
<td>-2.07e+06,</td>
<td>-7.65e-01,</td>
<td>1.030e+00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.82e-02</td>
<td>2.07e+06</td>
<td>7.65e-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AML</td>
<td>6.610e-01</td>
<td>-2.216e+06</td>
<td>-2.749e+00</td>
<td>1.430e+00</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>6.81e-01,</td>
<td>-2.07e+06,</td>
<td>-2.74e+00,</td>
<td>1.430e+00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.27e-01</td>
<td>2.07e+06</td>
<td>2.74e+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faces</td>
<td>-4.580e-01</td>
<td>-8.713e+05</td>
<td>-4.580e+01</td>
<td>1.370e+00</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>-8.58e-01,</td>
<td>-8.58e+05,</td>
<td>-8.58e+01,</td>
<td>1.370e+00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.80e-01</td>
<td>8.58e+05</td>
<td>8.58e+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8.58e+01</td>
<td>1.370e+00</td>
<td></td>
</tr>
<tr>
<td>Hubble</td>
<td>-5.389e-01</td>
<td>-1.563e+05</td>
<td>-5.389e-01</td>
<td>1.620e+00</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>-3.17e+00,</td>
<td>-1.56e+05,</td>
<td>-3.17e+00,</td>
<td>1.620e+00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.80e-01</td>
<td>1.56e+05</td>
<td>3.17e+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

57
Multiple Solutions in NMF: A Particle-Based Approach with Transfer Learning

The non-parametric variational inference framework introduced in Chapter 4 leads to improved coverage of the BNMF posterior however concerns remain over the uncertainty defined by the model since isotropic Gaussian noise on the factorization parameters is restrictive and implies probability mass for negative entries in the factorizations. In this Chapter, we seek to further distill the presentation of the posterior by using a particle-based approach designed to construct a discrete collection of factorizations that represents the uncertainty present in the BNMF posterior. We also present a more flexible modeling framework that allows BNMF inference for models corresponding to other widely used NMF objectives. Finally, instead of employing geometric techniques directly to search for diverse factorizations, we employ an implicit approach that leverages the geometric structure of NMF present in synthetic data and transfer it to real datasets.

5.1 Problem Description

The goal of non-negative matrix factorization (NMF) is to find a rank-$R_{NMF}$ factorization for a non-negative data matrix $X$ ($D$ dimensions by $N$ observations) into two non-negative factor matrices $A$ and $W$. Typically, the rank $R_{NMF}$ is much smaller than the dimensions and observations ($R_{NMF} \ll$...
The linear, additive structure of these non-negative factor matrices makes NMF a popular unsupervised learning framework for discovering and interpreting latent structure in data. Each observation in the data $X$ is approximated by an additive combination of the $R_{\text{NMF}}$ columns of $A$ with the combination weights given by the column of $W$ corresponding to that observation. In this way, the basis matrix $A$ provides a part-based representation of the data and the weights matrix $W$ provides an $R_{\text{NMF}}$-dimensional latent representation of the data under this part-based representation.

The ability to easily interpret NMF solutions in this way has made them appealing in many applied areas. A few applications of NMF include understanding protein-protein interactions (Greene et al., 2008), topic modeling (Roberts et al., 2016), hyperspectral unmixing (Bioucas-Dias et al., 2012), polyphonic music transcription (Smaragdis and Brown, 2003), discovering molecular pathways from genomic samples (Brunet et al., 2004), and summarizing activations of a neural network for greater interpretability (Olah et al., 2018).

However, the analysis and interpretation of latent structure in a dataset via NMF is affected by the possibility that several non-trivially different pairs of $A, W$ may reconstruct the data $X$ equally well. This non-identifiability of the NMF solution space has been studied in detail in the theoretical literature (Pan and Doshi-Velez, 2016; Donoho and Stodden, 2003; Arora et al., 2012b; Ge and Zou, 2013b; Bhattacharyya et al., 2016), and domain experts using NMF as a tool have noticed this issue as well. Greene et al. (2008) use ensembles of NMF solutions to model chemical interactions, while Roberts et al. (2016) conduct a detailed empirical study of multiple optima in the context of extracting topics from large corpora.

Bayesian approaches to NMF promise to characterize this parameter uncertainty in a principled manner by solving for the posterior $p(A, W | X)$ given prior $p(A, W)$ and likelihood $p(X | A, W)$ e.g. Schmidt et al. (2009); Moussaoui et al. (2006). Having a representation of uncertainty in the parameters of the factorizations can assist with the proper interpretation of the factors, allowing us to place low or high confidence on parameters of the factorization. However, computational tractability of inference limits the application of the Bayesian approach. Uncertainty estimates obtained from current Bayesian methods are often of limited use: variational approaches (e.g. Paisley et al. (2013); Hoffman and Blei (2013)) typically underestimate uncertainty and fit to a single mode; sampling-based approaches (e.g. Schmidt et al. (2009); Moussaoui et al. (2006)) also rarely switch between multiple modes and often require many thousands of samples for meaningful uncertainty.
estimates.

As a result of the limitations of current Bayesian approaches, domain experts tend to rely on non-Bayesian approaches to characterize uncertainty in NMF parameters. For example, Greene et al. (2008); Roberts et al. (2016); Brunet et al. (2004) all use random restarts to find multiple solutions. Random restarts have no Bayesian interpretation (as they depend on the basins of attraction of each mode), but they do often find multiple optima in the objective that can be used to understand and interpret the data.

Contributions In this work, we present a transfer-learning approach that remains faithful to a principled Bayesian framework and can efficiently identify multiple, disconnected modes for any differentiable prior and likelihood model. Our transfer-learning based approach provides high-quality and diverse NMF initializations to seed a particle-based approximation to the Bayesian NMF (BNMF) posterior. We demonstrate our inference approach on two different BNMF models: first, the common exponential-Gaussian model; second, a novel model that corresponds more closely with the desires of domain experts. Through our experiments, we demonstrate that:

- On a large number of real-world datasets, our particle-based posterior approximations consistently outperform baselines in terms of both posterior quality and computational running time.

- Our approach allows us to produce relatively small (less than 100 NMFs) sets of particles that belong to multiple modes of the posterior landscape, have distinct interpretations, and exhibit variability in performance on downstream tasks—all of which are essential for a domain expert to inspect and understand the full solution space.

- Our novel practitioner-friendly BNMF model involves a new scale-fixing prior that removes many uninteresting multiple optima and captures the kinds of loss-insensitive regions that are important in many applications. Inference in this non-conjugate model is significantly more challenging than with more standard BNMF models, but our approach handles this case with ease.

*Random restarts involve repeating an optimization procedure with different starting points that are independently sampled.
5.2 Inference Setting

The general process of Bayesian modeling consists of three main parts. First, we must select a model (a likelihood and prior). Next, we perform inference on the model given data (under some objective). Finally, we evaluate the quality of the inference. The main innovation in this work is a novel transfer-based approach to the inference phase (section 5.4). Along the way, we also introduce a novel model for BNMF that is more closely aligned to what domain experts desire from NMFs (section 5.5.2).

When performing inference, we must choose how we will approximate the true posterior \( p(A, W | X) \). For notational simplicity, let \( \theta = \{ A, W \} \) represent NMF parameters. We approximate the full BNMF posterior \( p(\theta | X) \) with a discrete variational distribution \( q(\theta | \theta_1:M, w_1:M) \) that has \( M \) different point-masses \( \theta_m \). Each \( \theta_m \) represents a different NMF solution’s full set of parameters: 
\[
\theta_m = \text{vec}[A^T_m, W_m],
\]
and is assigned probability mass \( w_m \). The functional form of the variational distribution is given by:

\[
p(\theta | X) \approx q(\theta | \theta_1:M, w_1:M) = \sum_{m=1}^{M} w_m \delta(\theta - \theta_m) \tag{5.1}
\]

where \( \delta \) is the Dirac delta distribution and \( \Delta^{M-1} \) is the probability simplex in \( \mathbb{R}^M \). Particle-based approximations are attractive to domain experts because each sample represents something that they can inspect and understand.

While there exist many methods for particle-based approximations (Monte Carlo, Sequential Monte Carlo, Markov Chain Monte Carlo), these techniques often only enjoy theoretical guarantees in the limit of infinite or very large samples. Recent work in Stein discrepancy evaluation (Liu and Feng (2016); Chwialkowski et al. (2016); Gretton et al. (2006); Liu et al. (2016); Gorham and Mackey (2015); Ranganath et al. (2016); details in section 5.3) now enables us to measure the quality of an arbitrary finite collection as a posterior approximation.\(^1\) As such, it opens the door to entirely new classes of particle-generation techniques, where traditional conditions, such as detailed balance, are now replaced with minimizing the Stein discrepancy \( S_p(q) \) \(^2\) between the true BNMF posterior

\(^1\)While the popular Kullback-Leibler divergence requires comparing the ratio of probability densities or probability masses, the Stein discrepancy can be used to compare a particle-based collection defined by probability masses with a continuous target distribution.

\(^2\)This notation refers to the Stein discrepancy (a variational objective) between two distributions \( p \) and \( q \).
\(p(\theta|X)\) and the discrete approximation \(q(\theta|\theta_{1:M}, w_{1:M})\):

\[
q^*(\theta|\theta_{1:M}, w_{1:M}) = \arg\min_{q(\theta|\theta_{1:M}, w_{1:M})} S_p(q(\theta|\theta_{1:M}, w_{1:M})) \quad \text{s.t.} \quad w_{1:M} \in \Delta^{M-1}
\]

(5.2)

As with all variational inference problems, the problem of posterior inference is now reduced to the problem of optimization of the objective above; we are free to explore any method for producing settings \(\{\theta_{1:M}, w_{1:M}\}\) to minimize the Stein discrepancy to the true posterior.

In the following, we observe that the task of minimizing this Stein discrepancy often depends on producing high-quality, diverse factorization collections \(\theta_{1:M}\) and determining their associated weights \(w_{1:M}\). In section 5.4, we introduce a transfer-learning based approach to efficiently suggest a diverse collection of particles and optimize their associated weights. We describe a traditional BNMF model along with a novel threshold-based one for NMF and discuss their merits in the context of our approach in section 5.5. Experimental details including parameter choices for our approach as well as description of baselines and evaluation metrics is provided in section 5.6. In section 5.7, we compare our approach to more traditional particle-based approaches (MCMC), more naive ways of generating candidate particle collections, as well as directly attempting to optimize the Stein objective above. We evaluate the quality of different posterior approximations both based on their Stein discrepancies, likelihoods and reconstruction on held-out data.

5.3 Background

Transfer learning The field of transfer learning aims to leverage models and inference applied to one problem to assist in solving related problems. It is of practical value because there may be an abundance of data and computational resources for one problem but not another (see Pan and Yang (2010) for a survey). In this work, we shall use the solutions to BNMF from small, synthetic problems to quickly solve much larger NMF problems.

Stein discrepancy The Stein discrepancy \(S_p(q)\) is a divergence from distributions \(q(\theta)\) to \(p(\theta)\) that only requires sampling from the variational distribution \(q(\theta)\) and evaluating the score function of the target distribution \(p(\theta)\). The Stein discrepancy is computed over some class of test functions \(f \in \mathcal{F}\) and satisfies the closeness property for operator variational inference (Ranganath et al., 2016): it is non-negative in general and zero only for some equivalence class of distributions \(q \in \mathcal{Q}_0\). For a

For a precise definition, see section 5.3
rich enough function class, the only distribution for which the Stein discrepancy is zero is the distribution \( p \) itself. The approximation quality between the discrete distribution \( q(\theta|\theta_{1:M}, w_{1:M}) \) and the posterior distribution \( p(\theta|X) \) of interest can be analytically computed using recent advances in Stein discrepancy evaluation with kernels (Liu and Feng, 2016; Chwialkowski et al., 2016; Gretton et al., 2006; Liu et al., 2016; Gorham and Mackey, 2015). The Stein discrepancy is related to the maximum mean discrepancy (MMD): a discrepancy that measures the worst-case deviation between expectations of functions \( h \in \mathcal{H} \) under \( p \) and \( q \) (Gretton et al., 2006).

\[
\text{MMD}(\mathcal{H}, q, p) := \sup_{h \in \mathcal{H}} \mathbb{E}_{\theta \sim q}[h(\theta)] - \mathbb{E}_{\theta' \sim p}[h(\theta')]
\]

By applying the Stein operator \( T_p \) corresponding to the distribution \( p \), the function space \( \mathcal{H} \) is transformed into another function space \( T_p(\mathcal{H}) = \mathcal{F} \). It is given by:

\[
(T_p h)(x) := \frac{1}{p(x)} \langle \nabla, p(x) h(x) \rangle
\]

Expectations under \( p \) of any \( f \in \mathcal{F} \) are zero, i.e. \( \mathbb{E}_{\theta' \sim p} f(\theta') = 0 \). This property of the Stein operator is of particular interest when the distribution \( p \) is intractable because evaluating the Stein discrepancy does not require expectations over \( p \) and the Stein operator \( T_p \) only depends on the unnormalized distribution via the score function \( \nabla_{\theta} \log p(\theta) \).

\[
\mathbb{S}_p(\mathcal{H}, q) := \sup_{f \in T_p(\mathcal{H})} (\mathbb{E}_{\theta \sim q} f(\theta))^2
\]

In this Chapter, we use a kernelized form of the Stein discrepancy. For every positive definite kernel \( k(\theta, \theta') \), a unique reproducing kernel Hilbert space (RKHS) \( \mathcal{H} \) is defined. Chwialkowski et al. (2016) showed that the Stein operator applied to an RKHS defines a modified positive definite kernel \( \mathcal{K}_p \) given by:

\[
\mathcal{K}_p(\theta, \theta') = \nabla_{\theta} \log p(\theta)^T \nabla_{\theta'} \log p(\theta') k(\theta, \theta') \\
+ \frac{\partial^2 k(\theta, \theta')}{\partial \theta_i \partial \theta'_i}
\]

63
Finally, the Stein discrepancy is simply the expectation of the modified kernel $K_p$ under the joint distribution of two independent variables $\theta, \theta' \sim q$.

$$S_p(q) = \mathbb{E}_{\theta, \theta' \sim q}[K_p(\theta, \theta')]$$

For a discrete distribution over $\theta_1:M$ with probability masses $w_1:M$ (of the form in equation 5.1), this can be evaluated exactly (Liu and Lee, 2016) as:

$$S_p(q) = \sum_{i,j=1}^{M} w_i w_j K_p(\theta_i, \theta_j)$$

$$= w^T Kw \quad (5.4)$$

The (pure) quadratic form $w^T Kw$ is a reformulation where $K \in \mathbb{R}^{M \times M}$ is the (positive definite) pairwise kernel matrix with entries $K_{ij} = K_p(\theta_i, \theta_j)$ and the probability masses $w_1:M$ are embedded into a vector $w \in \mathbb{R}^{M \times 1}$. Our particle-based variational objective (equation 5.2) simplifies to the form in equation 5.4. In Section 5.4, we will provide a method for estimating $\theta_1:M$ and $w_1:M$ for the BNMF problem.

5.4 Approach

In this Section, we describe our transfer-based inference. As noted in Section 5.2, creating a particle-based posterior involves two distinct parts: creating a collection of candidate NMFs $\theta_1:M$, and then optimizing their weights $w_1:M$. We introduce a novel transfer-based approach that uses state-of-the-art (non-Bayesian) algorithms to efficiently generate the candidate NMFs $\theta_1:M$ (Section 5.4.1). Given $\theta_1:M$, we optimize the weights $w_1:M$ via standard convex optimization tools to minimize Stein discrepancy. (See Algorithm 5.1 for the full algorithm.) In Section 5.7, we compare our approach for generating candidate NMFs and weights to other baselines, including those that use traditional methods for particle generation (MCMC), other ways of creating candidate NMFs (and then again using a convex optimization on the weights), and gradient-based optimization of the objective.
5.4.1 Learning factorization parameters $\theta_{1:M}$ via Transfer Learning

A natural approach to finding the factorization parameters $\theta_{1:M}$ is to optimize for them directly via the variational objective (equation 5.4), however, as we shall see in Section 5.7, the direct approach tends to get stuck in poor local optima and is computationally expensive. Since the quality of the variational approximation is determined solely by the value of the variational objective under a given set of parameters $\theta_{1:M}, w_{1:M}$, we are free to employ any technique that produces a suitable collection $\theta_{1:M}$.

We observe that to minimize the Stein discrepancy, we will need solutions that are both high-quality and diverse. Random restarts have been previously used to find multiple solutions in general (Gendreau and Potvin, 2010) and for NMF in particular (Greene et al., 2008; Roberts et al., 2016; Brunet et al., 2004). These restarts can take advantage of specialized (non-Bayesian) optimization algorithms for NMF (Lee and Seung, 2001; Lin, 2007; Hsieh and Dhillon, 2011) that are widely used in applied settings to produce single factorization parameters $\theta_m$ from some initialization; there also exist algorithms to speed up convergence of these methods (Salakhutdinov et al., 2002; Wild et al., 2004; Xue et al., 2008; Boutsidis and Gallopoulos, 2008). However, these random restarts do not take advantage of any of structure of NMF; for each new NMF instance we propose random initializations from scratch. As such, many may converge to the same mode—a waste of computational effort—while missing other modes (especially when the number of restarts is small).

In this Section, we introduce a transfer-based technique (which we will call $Q$-Transform) to speed-up, as compared to random restarts, the process of finding a diverse set of factorizations from high-density regions of the posterior. Our initializations are determined by identifying the low-rank subspace of the data (via singular value decomposition (SVD)) and then transforming it in specific ways. Figure 5.1 shows a schematic illustrating the idea: we generate subspace transformation matrices $Q_A, Q_W$ from a number of small, synthetic datasets and then apply those transformations to the dataset of interest. These transformations serve as more intelligent initializations—compared to random restarts—from which to apply NMF algorithms to obtain a more diverse collection of high-quality NMFs. Because our initializations are almost always already decent NMFs, convergence is also computationally faster. To explain our $Q$-Transform procedure, we first define the subspace transformation matrices, then describe the method for generating transformation matrices $Q_A, Q_W$ using synthetic data, and finally discuss how to apply them to real data sets (transfer learning).
Figure 5.1: A schematic of the transfer learning procedure for NMF: A small dataset is used to learn transformation matrices $Q_A, Q_W$. We then apply these transformation matrices to multiple larger datasets (with any number of dimensions or observations) using its SVD to obtain a transfer-based initialization.

Input: Data \{X\}, Rank \{R_{\text{NMF}}\}, # Factorizations $M$

Step 1: Perform $M$ repetitions of Algorithm 5.2 to get matrices $\{Q^m_A, Q^m_W\}_{m=1}^M$ or re-use them if previously constructed

Step 2: Apply $Q$-Transform (Algorithm 5.3) to get Initializations $\{A^m_0, W^m_0\}_{m=1}^M$

Step 3: Apply NMF algorithm to get Factorizations $\{A^m, W^m\}_{m=1}^M$

Step 4: Apply Algorithm 5.5 using a given BNMF model to get weights $\{w^m\}_{m=1}^M$ for approximate posterior

Output: Discrete NMF Posterior $\{w^m, A^m, W^m\}_{m=1}^M$

Algorithm 5.1: Particle-based Variational Inference for BNMF using $Q$-Transform

Subspace transformations $Q_A, Q_W$ relating SVD and NMF

A low dimensional approximation for the data $X$ can be obtained via the top $R_{\text{SVD}}$ vectors of the SVD $A_{\text{SVD}}, W_{\text{SVD}}$. An NMF $A, W$ of rank $R_T$ (which may be different from $R_{\text{SVD}}$) also leads to an approximation of the data. The NMF factors are interpretable due to the non-negativity constraint whereas the SVD factors typically violate non-negativity. However, both approaches describe low dimensional subspaces that can be used to understand and approximate the data. These subspaces are the same when $R_{\text{SVD}} = R_T$ and the NMF is exact (i.e. $X = AW$; corresponds to Type I non-identifiability in Pan and Doshi-Velez (2016)). Under these conditions, there exist transformation matrices $Q_A, Q_W$ to obtain the non-negative basis and weights exactly in terms of the singular value decomposition matrices:
If \( X = A_{\text{SVD}}W_{\text{SVD}} = AW \) then
\[ A = A_{\text{SVD}}Q_A \quad W = W_{\text{SVD}}Q_W \]

When the data \( X \) is not an exact NMF but rather a perturbation of it (i.e. \( X = AW + \epsilon \)), the singular subspace of the matrix is bounded by Wedin’s theorem (Wedin, 1972). We therefore still expect that there exist transformation matrices \( Q_A \in \mathbb{R}^{R_{\text{SVD}} \times R_A}, Q_W \in \mathbb{R}^{R_T \times R_{\text{SVD}}} \) to yield approximations of the NMF factorizations that can be expressed in terms of the singular value decomposition matrices.

\[ A_Q = A_{\text{SVD}}Q_A \approx A \quad W_Q = Q_WW_{\text{SVD}} \approx W \]

Our transfer-based strategy will involve identifying candidate matrices \( A_Q \in \mathbb{R}^{D \times R_A}, W_Q \in \mathbb{R}^{R_T \times N} \) such that \( A_{\text{SVD}}Q_A \) and \( Q_WW_{\text{SVD}} \) are likely to be good initializations for an NMF of the data \( X \). (Note that we assume that computing the SVD to obtain \( A_{\text{SVD}} \) and \( W_{\text{SVD}} \) from the data \( X \) is straight-forward.) We will describe the details for using these initializations below, but first we describe how we might create a collection of candidate transformation matrices \( Q_A, Q_W \).

**Generating transformations \( Q_A, Q_W \) for NMF initialization.** To generate candidate transformations, we note that if we have already computed an NMF \( A, W \) for a dataset \( X \), the appropriate transforms \( Q_A, Q_W \) can be computed by relating the SVD factors \( A_{\text{SVD}}, W_{\text{SVD}} \) to \( A, W \) (e.g. via linear least squares). We propose to generate candidate transforms by using random restarts on small, synthetic datasets \( X_s \) that follow some generative model for NMF, where we can run (non-Bayesian) NMF algorithms quickly and solve for \( Q_A, Q_W \) (Algorithm 5.2). Multiple pairs of transformation matrices can be obtained by repeating Algorithm 5.2 with different random initializations to compute NMF of the synthetic data \( X_s \), as well as by generating multiple synthetic datasets (see Section 5.8.1 for experiments and discussion of alternate generation procedures). Since the transformations \( Q_A, Q_W \) act on the inner dimensions (columns of \( A_{\text{SVD}} \) and rows of \( W_{\text{SVD}} \)), we emphasize that they can be applied to new datasets with any number of dimensions \( D \) and number of observations \( N \).

**Creating initializations for a new dataset.** Given the top SVD factors of a new dataset \( A_{\text{SVD}}, W_{\text{SVD}} \), we apply the \( Q \)-Transform (Algorithm 5.3) which multiplies SVD factors by the \( Q_A, Q_W \) matrices and adjusts entries of \( A_{\text{SVD}}Q_A \) and \( Q_WW_{\text{SVD}} \) to ensure non-negativity and cor-
Input: Synthetic Data \( \{X_s\} \), SVD Dimension \( \{R_{\text{SVD}}\} \), Transfer Dimension \( \{R_T\} \)

\[ A_{\text{SVD}}, W_{\text{SVD}} \leftarrow \text{Compute top } R_{\text{SVD}} \text{ SVD of } X_s \]

\[ A_{\text{NMF}}, W_{\text{NMF}} \leftarrow \text{Compute rank-} R_T \text{ NMF of } X_s \text{ using random initialization} \]

\[ Q_A = \text{arg min } Q \| A_{\text{NMF}} - A_{\text{SVD}} Q \|_F \text{ via linear least squares} \]

\[ Q_W = \text{arg min } Q \| W_{\text{NMF}} - Q W_{\text{SVD}} \|_F \text{ via linear least squares} \]

Output: \( Q_A, Q_W \)

**Algorithm 5.2: Generate Q-Transform Matrices**

Input: Real Data \( \{X\} \), SVD Rank \( \{R_{\text{SVD}}\} \), NMF Rank \( \{R_{\text{NMF}}\} \), Transformation Matrices \( \{Q_A, Q_W\} \)

\[ A_{\text{SVD}}, W_{\text{SVD}} \leftarrow \text{Compute top } R_{\text{SVD}} \text{ SVD of } X \]

\[ A_0 = A_{\text{SVD}} Q_A, \quad W_0 = Q_W W_{\text{SVD}} \]

\[ A_0, W_0 \leftarrow \text{Apply non-negativity and fix dimensions: Algorithm } 5.4(\tilde{A}_0, \tilde{W}_0, R_{\text{NMF}}) \]

Output: \( A_0, W_0 \)

**Algorithm 5.3: Apply Q-Transform**

rect dimensions using Algorithm 5.4 to obtain initializations \( A_0, W_0 \) that can be used as input for any standard (non-Bayesian) NMF algorithm (e.g. Cichocki and Phan (2009), Févotte and Idier (2011)). Algorithm 5.4 ensures that all values in the initialization \( A_0, W_0 \) are non-negative as well as provides a way to pad the initialization if the size \( R_T \) of the transforms \( Q_A, Q_W \) are smaller than the desired NMF rank \( R_{\text{NMF}} \). The latter is an important point: in Section 5.8.3 we find that it is often the first few dimensions of the transformation that contain transferable information, and the rest provide little benefit. This observation also allows us to use transforms of some rank \( R_T \) on problems with a range of desired NMF ranks \( R_{\text{NMF}} \). Finally, running the algorithm gives us a set of factorization parameters \( \theta_m = \text{vec}[A^{T_m}, W_m] \) that we may (or may not) ultimately decide to keep in our approximation of the true posterior.

In the experiments in Section 5.7, we find that knowledge from these transformations \( Q_A, Q_W \) can be transferred to real datasets by re-using them to relate the top SVD factors of other datasets to high quality, approximately non-negative factorizations.\(^6\)

\(^6\)Code and demonstrations at https://github.com/dtak/Q-Transfer-Demo-public/
Input: Approximation matrices \{A_Q, W_Q\}, NMF Rank \{R_{NMF}\}
\[ \tilde{A}_0 \leftarrow \text{Absolute Value}(A_Q) \]
\[ \tilde{W}_0 \leftarrow \text{Absolute Value}(W_Q) \]
Transfer Rank \(R_T = \# \text{ Columns of } A_Q\)
if NMF Rank \(R_{NMF} > R_T\) then
\[ r = R_{NMF} - R_T \]
Pad \(\tilde{A}_0, \tilde{W}_0\) with matrices \(M_{D \times r}\) and \(M_{N \times r}\) having small random entries so that initializations are the correct dimensions and matrices \(M_{D \times r}, M_{N \times r}\) have little effect of the product \(\tilde{A}_0\tilde{W}_0\).
\[ A_0 \leftarrow [\tilde{A}_0, M_{D \times k}] \]
\[ W_0 \leftarrow [\tilde{W}_0^T, M_{N \times k}]^T \]
else if NMF Rank \(R_{NMF} < R_T\) then
Pick the top \(R_{NMF}\) columns of \(\tilde{A}_0\) and rows of \(\tilde{W}_0\)
\[ A_0 \leftarrow \tilde{A}_0[\cdot, 0 : R_{NMF}] \]
\[ W_0 \leftarrow \tilde{W}_0[0 : R_{NMF}, :] \]
end if

Algorithm 5.4: Initialization Adjustment

Input: Particles \(\theta_{1:M}\), Score \(\nabla_\theta \log p(\theta)\), RKHS \(\mathcal{H}\) defined by kernel \(k\)
Step 1: Compute pairwise kernel matrix \(K_{i,j} = K_p(\theta_i, \theta_j)\) (from equation 5.3)
Step 2: Find probability masses that minimize the Stein discrepancy for the given point-masses:
\[ w^* = \arg \min_w w^T Kw \quad \text{s.t.} \quad w \in \Delta^{M-1} \] via standard convex optimization.

Output: Probability masses \(w^*\)

Algorithm 5.5: Kernelized Stein inference for discrete approximations to posterior

5.4.2 LEARNING WEIGHTS \(w_{1:M}\) GIVEN PARAMETERS \(\theta_{1:M}\)

To infer the weights corresponding to a given factorization collection \(\theta_{1:M}\), we minimize the Stein discrepancy (Algorithm 5.5) subject to the simplex constraint on the weights. This process involves first computing the pairwise kernel matrix\(^4\) \(K\) using the kernel \(K_p\) in equation 5.3. The objective function is convex and can be solved using standard convex optimization solvers. Given point-masses \(\theta_{1:M}\), this framework can be employed to infer weights for discrete approximations to any posterior for which the score function \(\nabla_\theta \log p(\theta)\) can be computed.

\(^4\)As the kernel \(K_p\) is positive definite, \(K\) is also positive definite.
5.5 BNMF Models

Section 5.4 outlined a general procedure for producing a particle-based approximation to the BNMF posterior using transfer learning. In Section 5.7, we compare our approach to other particle-based approaches for BNMF. However, before going to the results, we first describe the two BNMF models (below) as well as our experimental procedure (Section 5.6).

The first model we shall use in our experiments is the commonly-used exponential-Gaussian model. This model is computationally convenient to use (e.g. Schmidt et al. (2009) derive a Gibbs sampler for this model) but the scale-flexible prior allows for multiple optima that are essentially the same factorization, and the Gaussian likelihood severely penalizes solutions of differing quality even when all solutions may be far from perfect reconstructions. These properties make this popular model less desirable from the perspective of a domain expert seeking to understand their data.

The second model is a novel threshold-based model with a scale-fixing prior that at once removes scale ambiguities and allows for the kinds of likelihood ambiguities that practitioners expect—in particular, when the NMF is already an approximation of the data, solutions with different absolute likelihoods but whose relative differences are small compared to the magnitude of the likelihood may be considered similar by a practitioner (Roberts et al., 2016). Our threshold-based likelihood model allows the practitioner to choose what levels of error are, for their purposes, effectively the same.

Before continuing, we emphasize again that our transfer-based inference approach can be applied to any BNMF model; in this paper we demonstrate our approach on the following two models because together they include a standard model often-used in the machine learning community and a novel model of interest to the practitioner community. Importantly, because our inference approach decouples the process of model choice, particle generation, and particle weighting, we use the same particle generation process (non-Bayesian optimization algorithms using the Frobenius objective) for both models. In Section 5.7, we demonstrate empirically that this particle generation process is robust enough; that is, we do not require processes tuned to each model.
5.5.1 Exponential-Gaussian Model for BNMF

The commonly used exponential-Gaussian BNMF model uses a Gaussian likelihood and exponential priors for the basis and weights matrices:

\[
p_{N}(X|A, W) = \prod_{d,n} N(X_{d,n}, (AW)_{d,n}, \sigma_{X}^{2})
\]

\[
p(A) = \prod_{d=1}^{D} \prod_{r=1}^{R} p(A_{d,r}), \quad A_{d,r} \sim \text{Exp}(\lambda_{d,r})
\]

\[
p(W) = \prod_{n=1}^{N} \prod_{r=1}^{R} p(W_{r,n}), \quad W_{r,n} \sim \text{Exp}(\lambda_{r,n})
\]

As derived in Schmidt et al. (2009), the combination of exponential priors and Gaussian likelihoods results in element-wise conjugate parameter updates; in general, this model enjoys relatively straightforward inference approaches.

That said, as noted above, the exponential-Gaussian has several drawbacks from the perspective of a domain expert seeking to interpret their data via NMF. First, especially in settings where the model is misspecified (which will almost always be the case), the reconstruction error of even the best factorization may be relatively large. Even so, the Gaussian likelihood will tend to make the posterior highly peaked around the MAP solution—and exclude factorizations of only slightly worse relative approximation quality with respect to the overall error. However, domain experts may have found those factorizations interesting, as they have about the same relative error. Second, the exponential prior allows for some amount of uncertainty simply due to scale, which is typically uninteresting for domain experts. In the following, we introduce a model that addresses both of these shortcomings; because our transfer-based inference approach does not require conjugacy, we will be able to efficiently compute approximate posteriors for such more complex models.

5.5.2 Threshold-based, Scale-Fixing Model for BNMF

The procedure described in Algorithm 5.1 for finding a discrete approximation to the BNMF posterior does not depend on any special properties (such as conjugacy) and only requires the joint density \( p(X, W, A) \) to be differentiable in order to make inference tractable. Such flexibility is important as different applied domains use different notions of factorization quality: squared Euclidean distance is commonly used in hyperspectral unmixing (Bioucas-Dias et al., 2012), Kullback-Leibler
divergence in image analysis (Lee and Seung, 2001) and Itakura-Saito divergence in music analysis (Févotte et al., 2009).

A common theme in many applied domains is that small differences in factorization quality may not be important if all factorizations have some large level of approximation error. In such cases, domain experts may be interested in all of these solutions (Roberts et al., 2016). At the same time, solutions that are different only in scale are likely uninteresting. Below, we present a novel prior and likelihood that reflect these application-specific preferences of practitioners in a Bayesian framework. In particular, our model class allows domain experts to take any application-specific notion of a high-quality factorization—conjugate or not—and put it into a Bayesian context.

**Likelihood: Soft Insensitive Loss Function (SILF) over NMF objectives** We define a likelihood that is maximum (and flat) in the region of high quality factorizations and decays as factorization quality decreases. To do so, we use the soft insensitive loss function (SILF) (Chu et al., 2004): a loss function defined over the real numbers \( \mathbb{R} \), where the loss is negligible in some region around zero defined by the insensitivity threshold \( \epsilon \), and grows linearly outside that region (see figure 5.2). A quadratic term depending on the smoothness parameter \( \beta \), makes the transition between the two main regions smooth. This transition region has length \( 2\beta \), making smaller values of \( \beta \) correspond to sharper transitions between the flat and linear loss regions. We adapt the SILF from (Chu et al., 2004) to only be defined over the non-negative numbers \( \mathbb{R}_+ \) (as is typical with NMF objectives) and define it as:

\[
\text{SILF}_{\epsilon, \beta}(y) = \begin{cases} 
0 & 0 \leq y \leq (1 - \beta)\epsilon \\
\frac{(y - (1 - \beta)\epsilon)^2}{4\beta \epsilon} & (1 - \beta)\epsilon \leq y \leq (1 + \beta)\epsilon \\
y - \epsilon & y \geq (1 + \beta)\epsilon
\end{cases}
\]

To form the likelihood, we apply the SILF loss to an NMF objective \( f_X(A, W) \) to give:

\[
P(X|W, A) = \frac{1}{Z} e^{-C \times \text{SILF}_{\epsilon, \beta}(f_X(A, W))} \tag{5.5}
\]

We emphasize that the SILF-based likelihood allows the domain expert to use an NMF objective \( f_X(A, W) \) that is best suited to their task and can specify a threshold under that objective for identifying high-quality factorizations. Once an NMF objective is chosen, the domain expert can easily choose appropriate parameters for the SILF-based likelihood since the parameters (insensitivity factor \( \epsilon \) and smooth transition factor \( \beta \)) are interpretable and the likelihood can easily be visually
Figure 5.2: A comparison of SILF loss and commonly used $l_1$, $l_2$ loss functions. The SILF insensitivity parameter $\epsilon$ is set to 0.5, and the smooth transition factor $\beta$ is varied. Small values of $\beta$ lead to sharp transition in the SILF loss profile whereas the transition is less abrupt for large values of $\beta$. In contrast, other popular loss functions such as $l_1$ or $l_2$ do not have insensitive regions, and in the case of NMF, treat the objective function as the sole guide for factorization quality.

Prior: uniform over basis and unambiguous in factorization scaling

Often, domain experts do not have specific notions of what the prior over factorizations should be. However, prior distributions can have a large effect; These effects are undesirable if the prior was chosen for computational convenience rather than some true knowledge about the problem. Another concern is that NMF has a scaling and permutation ambiguity that is uninteresting in practice:

$$AW = ASP(PSP)^{-1}W$$ where $S$ is a positive diagonal matrix, $P$ is a permutation matrix (5.6)

Depending on the priors chosen, this ambiguity can add redundancy to the posterior distribution.

To facilitate exploration of the space of distinct high-quality factorizations, we propose an NMF prior that eliminates redundancy due to scale and is also uniform over the space of factorizations. Specifically, we let each column of the basis matrix $A_r$ be generated by a symmetric Dirichlet distribution with parameter $\alpha = 1$. This prior determines a unique scale of the factorization and is uniform over the basis matrix $A$ for that scaling. For $W$, we use a prior where each entry $W_{r,n}$ is i.i.d from an exponential distribution with parameter $\lambda_{r,n}$. The exponential distribution has support over all $\mathbb{R}_+$. ensuring that any weights matrix $W$ corresponding to a column-stochastic basis matrix.
\( A \) is a valid parameter setting under our model, and that the posterior is proper.

\[
p(A) = \prod_{r=1}^{R} p(A_r), \quad A_r \sim \text{Dir}(1_D)
\]

\[
p(W) = \prod_{n=1}^{N} \prod_{r=1}^{R} p(W_{r,n}), \quad W_{r,n} \sim \text{Exp}(\lambda_{r,n})
\]

### 5.6 Experimental Setup

In this Section, we provide details of our experimental settings and parameter choices, and describe our baseline algorithms and datasets. Our experiments are performed on a wide array of benchmark NMF datasets as well as on Electronic Health Records (EHR) data of patients with Autism Spectrum Disorder (ASD) that is of interest to the medical community (see quantitative and qualitative results in Section 5.7).

#### 5.6.1 Model, Evaluation, and Inference Settings

**Model: exponential-Gaussian model parameters:** We set the standard deviation \( \sigma_X \) to be equal to the empirical standard deviation of a reference NMF. The exponential parameter was set to one for each entry in the basis and weights matrices \( \lambda_{d,r} = \lambda_{r,n} = 1 \).

**Model: SILF model parameters:** While any objective can be put into the SILF likelihood, in the following, we used the squared Frobenius objective \( f_X(A, W) = \|X - AW\|_F^2 \). To set the threshold parameter \( \epsilon \) for each dataset, we use an empirical approach where we find a collection of 50 high-quality factorizations under default settings of scikit-learn (Pedregosa et al., 2011). The objective function is evaluated for each of them \( \{f_i\}_{i=1}^{50} \) and \( \epsilon = 1.2 \max_i f_i \). We set the remaining SILF likelihood sensitivity parameters \( \beta = 0.1, C = 2 \). For the prior, we identically set the exponential parameter for each entry: \( \lambda_{r,n} = 1 \).

**Inference: Generating Q-transform matrices for transfer:** For the \( Q \)-Transform initializations, we set the transfer rank and SVD rank \( R_T = R_{\text{SVD}} = 3 \). We generated twenty sets of synthetic data \( X_s \in \mathbb{R}_{+}^{12 \times 12} \) using non-negative matrices of rank \( R_T \) with truncated Gaussian noise. For each synthetic dataset, we find five pairs of transformation matrices through random
restarts. In all our experiments, the same set of $M_{\text{max}} = 100$ pairs of transformation matrices $\{Q_A^m, Q_W^m\}_{m=1}^{100}$ are applied to each of the real datasets.

**Inference: Solver for inferring weights $w_{1:M}$:** The optimization for the weights $w_{1:M}$ (Step 2 in Algorithm 5.5) is carried out using the Splitting Conic Solver (SCS) in the convex optimization package CVXPY (Diamond and Boyd, 2016).

**Inference and evaluation: Stein discrepancy base RKHS and parameters:** The Stein discrepancy for our variational objective requires a function space to optimize over. This optimization over the function space has an analytical solution when a reproducing kernel Hilbert space (RKHS) is used. Gorham and Mackey (2017) show that the Inverse Multiquadric (IMQ) kernel is a suitable kernel choice for Stein discrepancy calculations as it detects non-convergence to posterior\(^1\) for $c > 0$ and $b \in (-1, 0)$.

$$k_{\text{IMQ}}(\theta_i, \theta_j) = (\|\theta_i - \theta_j\|^2 + c^2)^b$$

Since the length scales of the basis and weights matrix differ, we define a kernel via a linear combination of two IMQ kernels defined separately over the basis $A$ and weights $W$.

$$k([A_1, W_1], [A_2, W_2]) = \frac{1}{2\gamma_A}(\|A_1 - A_2\|^2 + c_A^2)^{b_A} + \frac{1}{2\gamma_W}(\|W_1 - W_2\|^2 + c_W^2)^{b_W} \quad (5.7)$$

Here $\gamma_A = (c_A^2)^{b_A}$ and similarly $\gamma_W = (c_W^2)^{b_W}$ are scaling factors that ensure the kernel takes values between 0 and 1. In general, across our datasets, the Dirichlet prior on the basis matrix induces a small length scale for $A$ and a larger length scale for the weights $W$. We uniformly set $c_A = 1 \times 10^{-2}$, $c_W = 1 \times 10^3$ and $b_A = b_W = -0.5$ across all our datasets.

We note that choosing sensible values for these parameters—and validating them—is important. Kernel parameters that induce length scales that are too small or too large give rise to a similarity measure that either considers all factorizations completely dissimilar or completely similar respectively. In our experiments, our kernel choice gives rise to a similarity measure that distinguishes across collections of factorizations obtained from different algorithms. Our kernel similarity analysis shows agreement with difference between factorizations as measured by the Frobenius distances

\(^1\)Gorham and Mackey (2017) also prove that popular Gaussian and Matern kernels fail to detect non-convergence when the dimensionality of its inputs is greater than 3.
between basis and weights matrices (see figures C.10, C.11, C.12 in Appendix). The range in kernel similarity values and its agreement with alternative measures indicates that our parameter choices for the kernel are reasonable and fairly robust.

**Evaluation: Measuring computational time**  In experiments, we keep track of the time taken (initialization and optimization) to produce each of the $M_{\text{max}} = 100$ factorizations. We sample collections of size $M = \{5, 25, 50\}$ from these factorizations and report the total time taken to produce the factorizations in the collection alongside reporting the Stein Discrepancies for the approximate BNMF posteriors.

For the baselines below, the reported runtimes correspond to time taken to generate NMFs $\{\theta_m\}_{m=1}^M$ in the approximate posterior. For initialization approaches this corresponds to the time taken to generate the initialization and subsequent optimization time. To allow for a transparent comparison of the performance of these initialization approaches with MCMC and gradient-based algorithms, we report runtimes at various points in the duration of the MCMC chain and for the gradient-based algorithms. For more details on measuring computational time, see Appendix C.

### 5.6.2 Baselines

In the previous Section, we described the implementation details for our transfer-based inference approach. In this Section, we describe implementation details for three classes of baselines for our experiments: MCMC, which represents standard practice for generating particle-based posteriors; gradient-based approaches which directly minimize the Stein variational objective, which represent our main competitors; and alternate initialization approaches, which represent simpler ablations on our approach.

**Markov Chain Monte Carlo baselines**  MCMC approaches involve sampling from a Markov Chain whose stationary distribution is the posterior of interest, and are often considered the gold-standard for approximating posterior distributions (as opposed to variational methods). That said, for a finite sample size, MCMC will still be approximate—and thus we must still evaluate its quality with respect to the Stein objective. In this Chapter, we consider two different MCMC baselines:

---

**Factorizations across our different datasets have different scales but the kernel parameters were fixed across all datasets.**
- Hamiltonian Monte Carlo (HMC) Our HMC was initialized with an NMF obtained using the default settings of scikit-learn (Pedregosa et al., 2011) (warm start), and adaptively selects the step-size using the procedure outlined in Neal et al. (2011). We run the chain for a total of 10000 samples and at various intermediate points thin it to \(M = \{5, 25, 50\}\) factorizations and compute the Stein discrepancy using Algorithm 5.5. We repeat this experiment three times to capture variability in the performance of the HMC.

For our scale-fixing prior in Section 5.5.2, we needed to simulate Hamiltonian dynamics as defined on the manifold of the simplex. To do this, we incorporate a reparametrization trick (Betancourt, 2012; Altmann et al., 2014) to sample under the column-stochastic (simplex) constraints of the basis matrix \(A\), and a mirroring trick (Patterson and Teh, 2013) for sampling from the positive orthant for the weights matrix \(W\).

- Gibbs Sampling. Only the exponential-Gaussian model admits a conjugate form for straightforward Gibbs sampling. For experiments using the exponential-Gaussian, we use the same number of samples and thinning factor as with HMC for a Gibbs sampler. Similarly to the HMC baseline, the Gibbs sampler was also initialized with an NMF obtained using the default settings of scikit-learn (Pedregosa et al., 2011) (warm start).

**Gradient-based baselines** Gradient-based baselines optimize the collection of factorizations directly via gradient descent on the Stein variational objective. They represent the class of inference approaches most similar to ours. Gradient-based approaches typically require fixing the size of the collection. In our experiments, we set the size of this collection to be equal to \(M = 5\). Due to the large memory requirement of running this algorithm with automatic differentiation using autograd (Maclaurin et al., 2015), we were unable to run these algorithms for larger \(M\). We impose scaling and non-negativity constraints after every gradient step (for a total of 2000 steps) and keep track of the Stein discrepancy in relation to the algorithm’s runtime. The experiment is repeated three times to capture variability in its performance over multiple iterations. We use the following three algorithms:

- SVGD: Stein Variational Gradient Descent is a functional gradient descent algorithm (Liu and Wang, 2016) that optimizes a collection of particles (factorizations) to approximate the posterior. We replace the RBF kernel from the original work with the more principled IMQ-based kernel defined in equation 5.7.

- SVGD-Q is a variant were we initialize SVGD with the \(Q\)-Transform.
- DSGD: Direct Stein Gradient Descent is a variant where we replace the functional gradient descent of SVGD with the gradient of the Stein discrepancy (using automatic differentiation (Baydin et al., 2015; Maclaurin et al., 2015)).

**Initialization-based baselines**

Our $Q$-transform approach can be thought of as an initialization approach: we provide a way of creating a collection of particles that we believe are likely to be representative of the posterior. Our main algorithm can be run with any process for creating the collection (step 2 of Algorithm 5.1). Our final set of baselines considers other alternatives to creating the collection.

- Random restarts Our random restart initializations for NMF in scikit-learn (Pedregosa et al., 2011) set each entry of the factors $A, W$ as independent, coming from a truncated standard normal distribution. These entries are all scaled by $\eta = \sqrt{\frac{1}{\text{NMF}} \sum_{D,N} X_{d,n}}$ and are given by: $A_{d,k}^0, W_{k,n}^0 \sim \eta \mathcal{N}(0, 1)$.

- NNDSVDar NNDSVDar is a variant of a popular initialization technique called Nonnegative Double Singular Value Decomposition (NNDSVD) which was introduced by Boutsidis and Gallopoulos (2008). It is based on approximating the SVD expansion with non-negative matrices. Since the NNDSVD algorithm is deterministic, this only gives a single initialization. The NNDSVDar variant of this initialization replaces the zeros in the NNDSVD initialization with small random values. We use the scikit-learn initialization for NNDSVDar which uses a randomized SVD algorithm (Halko et al., 2011), and note that it introduces some additional variability in the initializations.

### 5.6.3 Datasets

Our datasets cover a range of different types and can be divided into three main categories (count data, grayscale face images and hyperspectral images). The ranks for hyperspectral data are chosen according to ground truth values. In the 20-Newsgroups data, we select articles from 16 newsgroups (hence the rank 16) and for other datasets we pick a rank that corresponds to explaining at least 70 percent of the variance in the data (as measured by the SVD). Table 5.1 provides a description of each dataset as well as the rank used and a citation. The Autism dataset is of interest to the medical community for understanding disease subtypes in the Autism spectrum and is not publicly available. The remaining datasets are public and are considered standard benchmark datasets for NMF. In our
experiments, we hold out ten percent of the observations and report performance on both provided and held-out observations.

**Table 5.1: Datasets for NMF**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dimension</th>
<th>Observations</th>
<th>Rank</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-Newsgroups</td>
<td>1000</td>
<td>8926</td>
<td>16</td>
<td>Newspaper articles ([20NG, 2013])</td>
</tr>
<tr>
<td>Autism</td>
<td>2862</td>
<td>5848</td>
<td>20</td>
<td>Patient visits ([Doshi-Velez et al., 2014])</td>
</tr>
<tr>
<td>LFW</td>
<td>1850</td>
<td>1288</td>
<td>10</td>
<td>Grayscale Faces Images ([LFW, 2017])</td>
</tr>
<tr>
<td>Olivetti Faces</td>
<td>4096</td>
<td>400</td>
<td>10</td>
<td>Grayscale Faces Images ([Samaria, 1994])</td>
</tr>
<tr>
<td>Faces CBCL</td>
<td>361</td>
<td>2429</td>
<td>10</td>
<td>Grayscale Faces Images ([CBCL, 2000])</td>
</tr>
<tr>
<td>Faces BIO</td>
<td>6816</td>
<td>1514</td>
<td>10</td>
<td>Grayscale Faces Images ([Jesorsky et al., 2001])</td>
</tr>
<tr>
<td>Hubble</td>
<td>100</td>
<td>2046</td>
<td>8</td>
<td>Hyperspectral Image ([Nicolas Gillis, 1987])</td>
</tr>
<tr>
<td>Salinas A</td>
<td>204</td>
<td>7138</td>
<td>6</td>
<td>Hyperspectral Image ([SalinasA, 2015])</td>
</tr>
<tr>
<td>Urban</td>
<td>162</td>
<td>10404</td>
<td>6</td>
<td>Hyperspectral Image ([Zhu et al., 2014])</td>
</tr>
</tbody>
</table>

**5.7 Results**

In this Section, we compare computational time and Stein discrepancy values for variational posteriors obtained through different algorithms. For the exponential-Gaussian model, our approach using $Q$-Transform is either the most-competitive or second in performance to the Gibbs sampler for this model. Under the SILF model, we find that our approach for BNMF posterior approximation using transfer learning ($Q$-Transform) consistently produces the highest quality posterior approximations in the shortest amount of time (see Section 5.6.1 for details on runtime calculation). Inspection of factorization parameters from $Q$-Transform reveals that the parameter uncertainty captured by the BNMF posterior approximation has meaningful consequences for interpreting and utilizing these factorizations.

In appendix C, we provide an in-depth look at our results. We report on quality metrics for both the training data (figures C.4, C.5, C.1 and C.2) as well as held-out data (figures C.6 and C.3); we report on multiple metrics for measuring diversity of factorizations obtained from different algorithms (figures C.10, C.11, C.12, C.7, C.8 and C.9). Overall, these results support the notion that the Stein discrepancy is lowest for algorithms with the most diverse collection of high-quality factorizations.
5.7.1 **Exponential-Gaussian Model Results**

In figure 5.3, we show the performance of our algorithm and other competing baselines across our various datasets. Overall, we note that the best approximate posteriors are produced in the shortest time either by our \( Q \)-Transform algorithm or the Gibbs sampler for this model. Using random restarts for initialization yields approximate posteriors with similar Stein discrepancies to our approach but typically takes more time. The gradient-based approaches (even \( Q \)-SVGD which is initialized with \( Q \)-Transform) rarely do well, often plateauing at much higher discrepancies.

Stein discrepancy over time for exponential-Gaussian BNMF Discrete Posteriors

![Stein discrepancy graphs](image)

**Figure 5.3:** For each dataset we show the quality of the BNMF approximate posterior \( M = 5 \) and the corresponding runtime of \( Q \)-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to the exponential-Gaussian BNMF (lowest Stein discrepancy) are produced either using the \( Q \)-Transform initializations (in red) for the Gibbs sampler (in yellow).

While the likelihood term in this model is invariant to (redundant) scalings\(^\dagger\), a limitation is that

\(^\dagger\)Basis and weights matrices can be multiplied by any positive diagonal matrix and its inverse (respectively)
the prior (chosen for computational convenience) is dependent on the scaling. We find that this is an undesirable feature because the posterior landscape includes infinite redundant scalings and therefore requires greater effort from the inference procedure to find appropriate scalings of factorizations. Another concern is that the likelihood model is not directly expressible in terms whatever properties might be of interest to a practitioner. To address our concerns regarding the exponential-Gaussian model, we focus for the remainder of this work on the threshold-based model with scale-fixing prior.

5.7.2 SILF-based Model Results

In figure 5.4, we show the performance of our algorithm and other competing baselines across our various datasets. Recall that the Stein discrepancy variational objective involves terms that consider both the quality of the factorizations (as given by the score function \( \nabla_{\theta} \log p(\theta) \)) and their similarity (as given by the base RKHS kernel \( k(\theta_i, \theta_j) \)). The NNDSVDa r initializations and thinned HMC samples lead to factorizations that are high-quality but often not diverse (see diversity analysis in supplementary material: figures C.10, C.11, C.12). The SVGD and the DSGD are generally the worst performing algorithms. These methods are often unable to find factorization parameters that meet the quality criteria of the SILF likelihood (see quality analysis in supplementary material: figures C.4 and C.5). This is understandable because even using simple gradient-based approaches to find a single high-quality NMF turns out to be difficult, hence the existence of a literature on specialized algorithms for performing NMF. Our \( Q \)-transform algorithm and random restarts are able to find samples that are both high-quality and diverse, thus achieving the lowest Stein discrepancies; however, our \( Q \)-transform algorithm does so in the shortest time.

Figures C.17 and C.18 in the Appendix C show results for \( M = \{25, 50\} \) where \( Q \)-Transform continues to have a runtime advantage over other baselines. Additionally, for some datasets (Olivetti Faces, LFW and Faces BIO) \( Q \)-Transform also produces higher quality of the posterior approximations. Variational posteriors constructed using thinned samples from HMC significantly lack diversity as the Stein discrepancies for collections of size 5, 25 and 50 are comparable. This indicates that the HMC chain only explores a small region of the posterior distribution and can be confirmed through the diversity analysis in the Appendix C (figures C.10, C.11, C.12). Smimchisescu et al. (2007) notes that in high dimensional spaces, we expect there to be many ridges of probability as there are likely to be some directions in which the posterior density decays sharply. Alternatively, there may to yield ‘new’ factors that identically reconstruct the data but differ in scale.
be several isolated modes with no connecting regions of high probability making it particularly challenging for the HMC chain to avoid getting stuck in a local mode of the BNMF posterior.

Stein discrepancy over time for SILF BNMF Discrete Posteriors

![Stein discrepancy over time for SILF BNMF Discrete Posteriors](image)

Figure 5.4: For each dataset we show the quality of the BNMF approximate posterior ($M = 5$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).

**INTERPRETATION AND UTILIZATION OF POSTERIOR ESTIMATES**

BNMF posteriors can provide insight into the non-identifiability present within a particular dataset. Different factorizations may explain the data as a whole equally well, but do it through dictionary elements that have different interpretations, or can be used to understand specific parts of the data better than other factorizations. We show visual examples of diversity in the top words of the 20 Newsgroups BNMF posteriors and examples of how performance in downstream tasks for the 20
Figure 5.5: The top 15 words for topic A (computers/electronics) and topic B (space) shows that different factorizations provide an emphasis on different terms. In topic A, the top word from factorization 1 and 2 is ‘card’, but it does not appear in the top 15 words of factorization 3. Instead a similar term ‘chip’ is emphasized in Factorization 3. In topic B, the terms ‘space’ and ‘nasa’ appear in all three factorizations but factorization 2 is the only one with digital terms like ‘ftp’, ‘server’, ‘site’ and ‘faq’. In contrast factorization 1 and 3 both contain more physical terms like ‘sun’, ‘moon’, ‘launch’.

Newsgroups and Autism dataset is dependent on the posterior samples. Our analysis yields meaningful insights that could not be gained through a single factorization.

20-Newsgroups Our BNMF of 20-Newsgroups was a rank 16 decomposition of posts from 4 categories. In figure 5.6, we show the held-out AUC of a classifier trained to predict those categories based on the weights matrix $W$ from each factorization in our variational posterior. Even though all of these factorizations have essentially equivalent reconstruction (see figure C.5 in supplementary material), there exists a significant variation in the performance of these NMFs on the prediction tasks. The best performing NMF for one category is generally not the best (or even one of the top performing) NMFs for other categories. This observation may be valuable to a practitioner intending to use the NMF for some downstream task: different samples explain different patterns in the data. In figure 5.5, we see that this is indeed true: even after alignment, distinct NMF factorizations have top words that indicate different emphasis across topics.

Autism Spectrum Disorder (ASD) In addition to core autism symptoms, Doshi-Velez et al. (2014) describe three major subtypes in autism spectrum disorder: those with higher rates of neurological disorder, those with higher rates of autoimmune disorders, and those with higher rates of psychiatric disorders. In figure 5.7, we show the number of topics that contain key terms corresponding to these areas (expressive language disorder, epilepsy, asthma, and attention deficit disorder).

‡‡We compare topics after finding the permutation of columns that best aligns them by solving the bipartite graph matching problem. We minimize the cost given by the angle between topics.
Figure 5.6: Classifiers trained on feature vectors from different factorizations yield variability in prediction performance (as measured by AUC). The dotted lines show the factorization that produces the best performing classifier for each category. The factorization (blue dotted line) that predicts the 'Talk' category best is actually one of the worst performing factorizations for the 'Science' category. This variability in performance demonstrates that no single factorization gives the best latent representation for the overall prediction task.

On the same set of patients, we can also ask whether we can predict the onset of certain medical issues in the subsequent patient trajectory. We train a classifier on the weights of the NMFs to predict the onset of these medical issues. Similar to the category prediction results in 20-Newsgroups, figure 5.8 shows that there is a large variability (around 0.1 in AUC) in the performance of classifiers trained on the weights matrices of different factorizations on the prediction task. No single factorization has the best performance across the different prediction tasks.

5.7.3 Extension: BNMF in the presence of missing data

In the presence of missing data, there is perhaps an even greater need to understand the uncertainty in factorization parameters for NMF. The factorization space of a fully observed dataset forms a subset of the factorization space in the presence of missing data. Our particle-based approach to BNMF posterior approximation can be applied to the missing data setting by making some minor adjustments to the experimental settings.

The multiplicative update algorithm for NMF (Lee and Seung, 2001) can be adjusted so that
Figure 5.7: We explore top words in the topics relating to key terms of interest to clinicians and discover that different NMFs place varying amount of emphasis on different terms. Such variability is of interest to clinicians who may be trying to interpret topics to understand patterns in ASD.

Figure 5.8: Classifiers trained on weights matrix of $M_{max} = 100$ different factorizations to predict the presence of certain medical codes in a patient’s trajectory exhibit significant variability in prediction on a test set (as measured by AUC). Different factorizations lead to top predictors for the onset of different medical issues.
Figure 5.9: Under different percentages of missingness in the Olivetti Faces dataset (10%, 30%, 50%), the quality of the BNMF approximate posterior and the corresponding runtime of $Q$-Transform and the other baselines is shown. The best discrete posterior approximations to BNMF are produced using the $Q$-Transform initializations (in red).

Olivetti Faces (missing 30% data)

Figure 5.10: Sample factorizations from the variational posterior using $Q$-Transforms show that a diverse range of basis elements can be use to approximate the data. However, HMC samples seem to be identical indicating that HMC was only exploring a very small region of the posterior space.

the update equations for factorization parameters only consider the observed data. We use an implementation of this modification to the multiplicative update algorithm to find a completion of the data $X$, compute the SVD subspace and then apply our $Q$-Transform initializations. Figure 5.9 demonstrates that our approach to BNMF can be extended to the case where the data matrix $X$ is partially observed. For the Olivetti Faces dataset with varying degrees of missingness, the $Q$-Transform approach to BNMF consistently finds posterior approximations that are significantly better (as measured by Stein discrepancy) than other baselines whereas for a given $M$, the runtime is second-lowest.

Figure 5.10 shows sample factorizations from the variational posterior using $Q$-Transform and HMC samples. To allow for comparison, we have aligned the positions of the basis (dictionary)
elements to a reference factorization using the bipartite matching algorithm. It is clear from looking at the $Q$-Transform factorizations that a diverse range of dictionaries can be used to approximate the data well whereas the HMC chain only explores one set of dictionary elements. Interestingly, the diversity of solutions obtained using $Q$-Transform have visually interpretable differences, i.e. these are not simply perturbations of some ground truth basis elements. Some of the basis elements look like faces and some of them look like different shadow or lighting configurations. In contrast, the factorization samples from HMC have basis elements that look identical. This indicates that the HMC has explored a limited region of the posterior space.

5.8 Discussion: When is $Q$-Transform successful?

Our ability to extract transferable low-rank transformation matrices from an SVD and an instance of NMF indicates that there exist similarities across different NMF problems. In this Section we seek to develop a better intuition behind the success of the $Q$-Transform initializations at exploiting these similarities. In this Section, we provide discussion and smaller-scale experiments to shed light on when, why, and how our $Q$-transform approach is successful.

5.8.1 $Q$-Transform Generating Process

In our approach, we generated candidate $Q$-Transform matrices (Algorithm 5.2) by applying random restarts to small, synthetic data sets. We focused on this approach because small datasets are much faster to train, and with synthetic data sets, we can know at least one ground truth NMF and level of noise. However, there are obviously a large number of choices for the data used to generate candidate $Q$-Transform matrices.

In figure 5.11, we present results with a variety of different methods for generating candidates. In all cases, the source data was of small dimension ($X_S \in \mathbb{R}^{15\times15}$), and the target data was larger ($X_T \in \mathbb{R}^{500\times500}$). The target data had a true non-negative rank of 10 and factors were generated with i.i.d. entries from a standard normal. In all these experiments we set the transfer rank to be $R_T = R_{SVD} = 3$. We explored six ways of generating candidates from the source data:

- Uniform data: Generating a dataset $X_S$ where each entry is i.i.d. with a uniform distribution in [0,1]; then apply random restarts to find candidate transforms.
- Simple sub-sample data: Generating dataset $X_S$ by uniformly selecting 15 rows and columns of the target data $X_T$; then apply random restarts to find candidate transforms.


- Column-projection data: Generating dataset \( X_S \) by sub-sampling 15 columns of \( X_T \) and applying a random projection into \( \mathbb{R}^{15} \) for each each column; then apply random restarts to find candidate transforms.

- Dirichlet factors: Generating factors \( A, W \) with each column of \( A, W \) from a Dirichlet distribution (with concentration parameter \( \alpha \) set to 1); let \( X_S = AW + \text{Gaussian Noise} \); then apply random restarts to find candidate transforms.

- Uniform factors: Generating factors \( A, W \) with each entry i.i.d. from a uniform distribution in \([0,1] \); let \( X_S = AW + \text{Gaussian Noise} \); then apply random restarts to find candidate transforms.

- Gaussian factors: Generating factors \( A, W \) with each entry i.i.d. from a standard normal distribution; let \( X_S = AW + \text{Gaussian Noise} \); then apply random restarts to find candidate transforms.

The methods that produced the source data from some true NMF factors produced candidate transformations that resulted in the highest quality initializations on the target data (figure 5.11). In settings where a practitioner deals with a collection of similar NMF datasets (e.g. music analysis, hyper spectral images), there may be more clever ways in which the NMF solution spaces corresponding to a real dataset may yield more appropriate \( Q \)-Transforms specific to that type of data. Finally, we find in figure 5.12 that the performance of Gaussian factors synthetic data does not vary with the rank of the synthetic data (the transfer rank is still held fixed).

### 5.8.2 The \( Q \)-Transform Initialization versus Noise

In Section 5.4, we sought high-quality initializations because they generally require less time to converge. On synthetic target data \( X_T = AW + \epsilon N_o \) (\( D = N = 500, R = 20 \)) we explore the effect of increasing noise \( (\epsilon) \) on the quality of our transfer-based NMF initializations and the time taken to converge. Specifically: are there noise regimes in which the \( Q \)-transform method works better, and noise regimes in which it does not?

We normalize the norm of the noise matrix to be equal to the norm of the data \( \| N_o \| = \| AW \| \) so that the contribution of signal \( AW \) and noise \( \epsilon N_o \) to the data is equal when \( \epsilon = 1 \). We continue to use the same 100 pairs of \( Q_A, Q_W \) matrices. We compare the performance of \( Q \)-Transform over random restarts in terms of initialization quality (ratio of the reconstruction error from \( Q \)-Transform to the reconstruction error from random restart) and time to convergence (ratio of time
For different synthetic data $X_S$ generating procedures, we show the initialization quality obtained via the $Q$-Transform matrices on a target data $X_T$. Dirichlet, Uniform, Gaussian have significantly superior performance compared to Sub-sample, Column-projection and Uniform data. For comparison, we show the quality of NMF solutions (solid line) and random initializations (dashed line).

Figure 5.12: Using Gaussian factors for the synthetic data generation process with different ranks does not appear to change the quality of the $Q$-Transform initialization quality on the target data $X_T$. This indicates that this generating procedure is not sensitive to the rank in order to produce high quality (close to true NMF solution) initializations using $Q$-Transform. For comparison, we show the quality of NMF solutions (solid line) and random initializations (dashed line).

taken using $Q$-Transform initialization to time taken using random restart). In both metrics, the $Q$-Transform has an advantage over random restarts for values of the noise $\epsilon$ smaller than 1, and the advantage is greatest for smallest noise. Figure 5.13 shows that the advantage of $Q$-Transform initializations is highest in a low noise regime and decreases as the noise increases. This behavior makes sense because as noise increases, the data is no longer truly low rank.

5.8.3 Selecting ranks

We emphasize that there are two distinct ranks that need to be chosen when applying our technique. The first is the rank of the factorization $R_{NMF}$. There exist multiple approaches for choosing this rank, e.g. Tan and Févotte (2009); Alquier and Guedj (2017), and they can be applied to our approach (as well as any other NMF algorithm).

The second is choosing the transfer rank $R_T$. The transformation dimensions $R_T$ and $R_{SVD}$ determine the dimensions of transformation matrices $Q_A$, $Q_W$ which map basis vectors defining the top SVD subspace of dimension $R_{SVD}$ to a set of $R_T$ non-negative basis vectors that approximate the same subspace. The full initialization for NMF is obtained by either padding the initialization with small entries ($R_T < R_{NMF}$) or removing extra columns and rows of the factor matrices.
Figure 5.13: In the low-noise regime, the reconstruction error of $Q$-Transform initializations is significantly less than random restart initializations. This relative advantage gets smaller as the noise level increases. Similarly, the time taken to converge is significantly shorter than the random restart approach under the low noise scenario and continues to increase with noise. As expected, at high noise levels there exists no additional advantage to the $Q$-Transform approach (the optimization time ratio approaches 1).

$(R_T > R_{NMF})$. (For simplicity, we consider the case where the transfer rank and SVD rank are equal $R_T = R_{SVD}$ and the resulting transformation matrices $Q_A, Q_W$ are square.)

The choice of the transfer rank $R_T$ is specific to our algorithm, and in figure 5.14 we investigate how well our transfer learning performs for different choices of transfer rank $R_T$. In the experiment, we extract a set of 100 transformation matrices $Q_A, Q_W$ for transfer dimensions $R_T = R_{SVD} = \{1, 2, \ldots, 10\}$ using synthetic source data $(D = N = 15)$. Once constructed, we applied the transformation matrices to a $500 \times 500$ target dataset $X_T$ of rank $R = 10$. We find that even though the dataset $X_T$ has rank 10, the rank 10 transformation matrices found using the $15 \times 15$ synthetic source dataset are unable to successfully transfer to this new dataset. We see that the error initially decreases, but then increases as the transfer rank increases. This result suggests that the top directions of variation hold the most transferable information across NMF problems.

5.8.4 Sign Convention for SVD

In considering when $Q$-Transform is successful, we note that there exists an intrinsic ambiguity in the sign of the singular vectors of $X$: changing the sign of any column of $A_{SVD}$ and corresponding row of $W_{SVD}$ gives a valid SVD. For $Q$-Transform to work, we must apply a consistent resolution of the sign ambiguity (e.g. from Bro et al. (2008)). This ensures that learned transformations $Q_A, Q_W$ map in a consistent way to SVD decompositions of new datasets.
On a synthetic target dataset ($D = N = 500, K_{\text{NMF}} = 10$), we apply $Q$-Transform initializations using varying transfer ranks and SVD ranks $R_T = R_{\text{SVD}} = \{1, 2, \ldots, 10\}$. We see that for a range of low rank values, the $Q$-Transform initializations are high quality, but at larger values the quality of initializations gets worse. The dotted line shows the quality of random initializations and the solid line shows the quality NMF solutions. The reconstruction errors are normalized by the norm of the data.

5.9 Related Work

There is a large body of work on inference for BNMF. Sampling-based approaches include Gibbs sampling (Schmidt et al., 2009), Hamiltonian Monte Carlo (Schmidt and Mohamed, 2009), and reversible jump variants (Schmidt and Mørup, 2010). All of these have trouble escaping local modes (Masood et al., 2016b), and are often constrained to a limited class of tractable distributions. Variational approaches to BNMF have successfully yielded interpretable factorizations (Bertin et al., 2009; Cemgil, 2009; Paisley et al.; Hinrich and Mørup, 2018) but also typically only capture one mode and rely on mean-field or other modeling assumptions to make inference tractable. We note that in many cases, priors of convenience—for example, exponential distributions—can induce a single dominant mode, even when that was not the intent of the practitioner.

Closer to the goals of our work, Gershman et al. (2012) develop a non-parametric approach to variational inference that provides flexibility in modeling the number of Gaussian components required to approximate a posterior. However, the isotropic covariance in the model makes it unsuitable for applying it to BNMF. With regard to the inference process, our $Q$-Transform approach to finding multiple optima is most similar to Ročková and George (2016) and Paatero and Tapper (1994), who use rotations to find solutions to a single matrix factorization problem that are sparse and non-negative respectively. In contrast, we use rotations to find multiple non-negative solutions, and also demonstrate how these rotations can be re-used for transfer learning.

More broadly, recent work on NMF has involved theoretical work on non-identifiability with
new algorithms that can provably recover the NMF under certain assumptions (Li and Liang, 2017; Bhattacharya et al., 2016; Ge and Zou, 2015). However, these assumptions are often difficult to check and may indeed be violated in practice; Bayesian methods typically provide more flexibility in modeling and assumptions.

All of the works above typically assume some desired factorization rank. There also exists work on models that automatically detect the rank—through automatic relevance determination for NMF (Tan and Févotte, 2009) or more recently, via a rank-adaptive prior Alquier and Guedj (2017). These works are complementary to ours, in that those techniques could be combined with our transfer-based approach of generating candidates of whatever rank those algorithms determine is appropriate.

The ability of Stein discrepancies to assess the quality of any collection of particles (Gorham and Mackey, 2015) has resulted in large recent interest in other ways to create collections of samples (Oates et al., 2017; Liu and Wang, 2016). Liu et al. (2016) and Chwialkowski et al. (2016) showed that kernelized Stein discrepancy could be computed analytically in reproducing kernel Hilbert spaces (RKHS); Pu et al. (2017) and Feng et al. (2017) use neural networks instead. Ranganath et al. (2016) establish the Stein discrepancy as a valid variational objective. To our knowledge, Stein discrepancy-based posterior approximation has not been applied to NMF, and yet, we see that it allows us to leverage existing non-Bayesian approaches to characterize these multi-modal posteriors. In our work, the Dirichlet prior on the columns of the basis matrix $A$ is important to ensure that we avoid a known saddle point of the zero factorization (from likelihood term) that yields a corresponding zero for the score function.

5.10 Conclusion

In this Chapter, we presented a novel transfer learning-based approach to posterior estimation in BNMF. Simply creating collections of factorizations via random restarts on our $Q$-Transform initializations, and then weighing them, produces diverse collections that approximate the posterior well (the NNDSVDar-based methods fail to produce diverse collections for posterior estimation). In contrast, the functional gradient descent of SVGD and direct gradients of Stein discrepancy (DSGD) perform worse to the collection-based approaches, requiring more time and also limiting the user to specify in advance the number of factorizations. Hamiltonian Monte Carlo also suffers from difficulties in exploring the posterior space, something random initializations are well suited to. Our transfer learning approach consistently produces the highest quality posterior approxima-
Through $Q$-Transform, we introduce a way to speed-up the process of finding multiple diverse NMFs. The discovery that $Q$-Transform matrices can transfer from synthetic to multiple real datasets is exciting and also suggests interesting questions for further research. For example, what is the theoretical nature of the similarities between principal eigenspaces of different non-negative matrices and the relation between their SVD and NMF bases? And, how does the synthetic data generation process used to obtain $Q$-Transform matrices impact the initializations and the effectiveness of the $Q$-Transform algorithm in general?

More broadly, our qualitative results demonstrate that even relatively simple models, such as NMF, can have multiple optima that are comparable under the objective function but have large variation in how well they explain different portions of the data— or how they perform on different downstream tasks. Thus, it is important to be able to compute these posteriors efficiently.
The Bayesian Non-Negative Matrix Factorization (BNMF) investigations (Chapters 2, 3, 4, 5) revealed benchmark and clinical datasets for NMF give rise to interesting forms of diversity in the factorization space, leading to different interpretations or downstream results. In this Chapter, we extend our investigation of multiple solutions to another important class of machine learning problems: sequential decision-making. Reinforcement Learning (RL) algorithms are designed to solve sequential decision-making problems. In this Chapter, we motivate the importance of multiple solutions in RL, demonstrate that existing approaches are lacking, and provide an algorithm that can be used in conjunction with existing state of the art methods to sequentially discover a set of diverse policies that are high-quality.

6.1 Problem Description

Standard reinforcement learning methods find one way to solve a task, even though there may exist multiple near-optimal policies that are distinct in some meaningful way. Identifying this collection of near-optimal policies can allow a domain expert to efficiently explore the space of reasonable solutions. For example, knowing that there exist comparably-performing policies that trade between several small doses of a drug or a single large dose may enable a clinician to identify what might work
best for a patient (e.g. based on whether the patient will remember to take all the small doses).

Unfortunately, existing approaches to find a set of diverse policies involve notions of diversity that are not aligned with the kind of efficient exploration-amongst-reasonable-options setting described above. Liu et al. (2017) characterize the uncertainty over policies via computing a posterior over policy parameters, but differences in policy parameters may not result in qualitatively different behavior (especially in over-parameterized architectures). Haarnoja et al. (2017) encourage diversity via encouraging high entropy distributions over actions (given states), which may result in sub-optimal behavior. Fard and Pineau (2011) seek a single non-deterministic policy that may make multiple decisions at any state, which may be overly restrictive if action choices across states must be correlated to achieve near-optimal performance.

We argue that differences in trajectories (state visits and action choices) better capture the kinds of distinct behavior we are seeking. For example, does one prefer a policy that achieves wellness through a surgery, or via prolonged therapy? More formally, stochasticity in the environment dynamics and the policy will induce a distribution over trajectories. We use the maximum mean discrepancy (MMD) metric to compare these distributions over trajectories under different policies. As noted in Sriperumbudur et al. (2010), the MMD metric has a closed form solution (unlike the Wasserstein and Dudley metrics) and exhibits better convergence behavior than \( \phi \)-divergences such as Kullback-Leibler (KL).

In this Chapter, we first formalize notions of policy diversity via the MMD over their induced trajectories. We show that unbiased gradient estimates of the MMD term can be obtained without knowledge of transition dynamics of the environment, and describe how it can be applied to any policy gradient objective. Across both benchmark domains and a healthcare task, our approach discovers diverse collections of well-performing policies.

6.2 Background

Reinforcement Learning  

We consider Markov decision processes (MDPs) defined by a continuous state space \( \mathcal{S} \), a (discrete or continuous) action space \( \mathcal{A} \), state transition probabilities \( p_T(s, a, s') \), reward function \( r(s, a) \), as well as a discount factor \( \gamma \). A policy \( \pi(s, a) \) indicates the probability of action \( a \) in state \( s \); together with the transition probability \( p_T(s, a, s') \), it induces a distribution \( p_\pi(\tau) \) over trajectories \( \tau = s_0, a_0, \ldots, a_{T-1}, s_T \). Traditionally, the task is to find a
policy that maximizes the long-term expected discounted sum of rewards (return) \( g(\tau) \):

\[
g(\tau) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[ \sum_t \gamma^t r(s_t, a_t) \right]
\]

In this Chapter, we shall seek multiple near-optimal policies.

**Policy Gradient Methods**  
Policy gradient methods use gradients to iteratively optimize a policy \( \pi_\theta(s, a) \) that is parameterized by \( \theta \). The standard objective \( J_{PG}(\theta) \) is the return \( g(\tau) \). An unbiased estimate of the gradient of the objective can be obtained by Monte Carlo rollouts generated by the policy \( \pi_\theta \) using the likelihood ratio trick. For a single rollout \( \{s_t, a_t, r_t\} \), the gradient can be estimated as

\[
\nabla_\theta J_{PG}(\theta) \approx \sum_{t=0}^T \nabla_\theta \log \pi_\theta(s_t, a_t) g_t
\]

where \( g_t = \sum_{t' = t}^T \gamma^{t' - t} r_{t'} \) is the return over the rewards received from time \( t \) onwards. Schulman et al. (2017) introduced Proximal Policy Optimization (PPO); a popular variant that achieves state-of-the-art performance on many benchmark domains and uses trust region updates that are compatible with stochastic gradient descent.

In this Chapter we augment the PPO objective for the policy network with an MMD-based term which encourages policies that lead to different state visitation and/or action choice distribution than previously-identified policies. An iterative use of this diversity inducing variant of policy gradient methods allows us to sequentially obtain distinct policies.

**Off-Policy Evaluation**  
The off-policy RL framework applies when trajectories are collected from a behavior policy that is distinct from the policy being trained, e.g., using observations from clinician behavior to optimize an agent’s behavior. In the batch setting, there are model-based and importance sampling approaches to estimating the value of a policy. Model-based approaches learn a dynamics model for the transitions and use those to simulate outcomes of a policy in order to estimate its value. Importance sampling approaches appropriately re-weight existing batch data to estimate the value for a different policy.

**Maximum Mean Discrepancy**  
We use the MMD metric to measure the difference between two trajectory distributions. The MMD is an integral probability metric (Gretton et al., 2007) that measures the difference between two distributions \( p, q \) using test functions \( h \) from a function space.
It is given by:

\[ \text{MMD}(p, q, \mathcal{H}) = \sup_{h \in \mathcal{H}} (\mathbb{E}_{x \sim p} [h(x)] - \mathbb{E}_{y \sim q} [h(y)]) \]

Computing the MMD is tractable when the function space \( \mathcal{H} \) is a unit-ball in a reproducing kernel Hilbert space (RKHS) defined by a kernel \( k(\cdot, \cdot) \) and is given by:

\[ \text{MMD}^2(p, q, \mathcal{H}) = \mathbb{E}[k(x, x')] - 2\mathbb{E}[k(x, y)] + \mathbb{E}[k(y, y')] \]

where \( x, x' \) i.i.d. \( \sim p \) and \( y, y' \) i.i.d. \( \sim q \). The expectation terms in the analytical expression for the MMD can be approximated using samples. In Section 6.3, we will show that computing the derivative of the MMD metric between trajectories with respect to the policy parameters \( \theta \) is tractable.

### 6.3 Diversity-Inducing Policy Gradient (DIPG)

Our algorithm constructs a set of diverse policies for an MDP by iteratively finding policies that are diverse relative to an existing set of policies. First, we formulate a diversity inducing objective function that regularizes any policy gradient objective. Then, we show that optimizing this objective is tractable using the familiar log-derivative trick. Next, we explain how to iteratively apply this diversity-inducing policy gradient objective in conjunction with an existing algorithm to find a set of distinct policies that solve an MDP. Finally, we introduce an extension to the proposed framework for off-policy batch reinforcement learning.

#### 6.3.1 DIPG Objective

We propose adding a regularization to encourage learning a policy \( \pi_\theta(s, a) \) that induces a distribution over trajectories \( p_\theta(\tau) \) that is distinct from a specified set of distributions over trajectories \( Q = \{q_m(\tau)\}_{m=1}^M \). (Below, these distributions \( q_m(\tau) \) will come from other policies.) Our diversity measure \( D_{\text{MMD}}(p_\theta(\tau), Q) \) is the squared MMD between the distribution of trajectories under the current policy \( \pi_\theta \) and the distribution \( q_*(\tau) \) in \( Q \) that is most similar to it.

\[
D_{\text{MMD}}(p_\theta(\tau), Q) = \min_{m \in \{1, \ldots, M\}} \text{MMD}^2(p_\theta(\tau), q_m(\tau)) = \text{MMD}^2(p_\theta(\tau), q_*(\tau))
\]
We define a kernel over a pair of trajectories \((\tau, \tau')\) using a kernel \(K\) (such as Gaussian kernel) defined over some function \(g\) of a vectorized representation of \(N\) steps of trajectories \((x = \phi_N(\tau), x' = \phi_N(\tau'))\). For trajectories that are not of the same length, we pick \(N\) to be the number of steps corresponding to the shorter of the two trajectories.

\[
k(\tau, \tau') = K(g(\phi_N(\tau)), g(\phi_N(\tau')))
\]

The function \(\phi_N\) is simply a way to stack the states and actions from the first \(N\) steps of a trajectory into a single vector. The function \(g\) gives the flexibility to adjust the focus of the diversity measure for different aspects such as state visits, action choices, or both.

The MMD-based diversity-inducing objective function is given by \(J_{\mathrm{MMD}}(\theta)\):

\[
J_{\mathrm{MMD}}(\theta) = J_{\mathrm{PG}}^*(\theta) + \alpha D_{\mathrm{MMD}}(p_\theta(\tau), Q)
\]

Where \(J_{\mathrm{PG}}^*(\theta)\) is the objective function of a policy gradient algorithm of the user’s choice (e.g. for vanilla policy gradient it would be the expected return) and the second term is proportional to the diversity measure between a policy distribution and the set of specified distributions \(Q\). The parameter \(\alpha\) decides the relative importance of optimality and diversity of the new policy \(\pi_\theta(s, a)\).

To optimize the MMD-based diversity-inducing objective, we need to specify how gradients of the diversity term can be computed with respect to the policy parameters \(\theta\).

### 6.3.2 Optimization via Gradient Ascent

To use gradient ascent on \(J_{\mathrm{MMD}}(\theta)\), what remains to specify is the gradient with respect to \(\theta\) of the diversity inducing term \(D_{\mathrm{MMD}}\). Let \(q_*(\tau)\) be the distribution in \(Q\) that minimizes the MMD between the state trajectory distribution of the policy \(\pi_\theta\) and \(q_m \in Q\). Then, the gradient with respect to the policy parameters \(\theta\) of the diversity term is given by

\[
\nabla_\theta D_{\mathrm{MMD}}(p_\theta(\tau), Q) = \nabla_\theta \mathrm{MMD}^2(p_\theta(\tau), q_*(\tau), H)
\]

\[
= \mathbb{E}[k(\tau_p, \tau'_p)\nabla_\theta \log(p_\theta(\tau_p)p_\theta(\tau'_p))]
- 2\mathbb{E}[k(\tau_p, \tau_q)\nabla_\theta \log(p_\theta(\tau_p)q_*(\tau_q))]
+ \mathbb{E}[k(\tau_q, \tau'_q)\nabla_\theta \log(q_*(\tau_q)q_*(\tau'_q))]
\]

98
Input: Known policies $\mathcal{P}_{\text{known}}$, MDP $\{S, A, p_s, r\}$, policy gradient objective $J^*_\text{PG}$, learning rate $\eta$, diversity parameter $\alpha$

Initialize policy parameters $\theta$

$Q \leftarrow$ Sampled trajectories from policies in $\mathcal{P}_{\text{known}}$

repeat
  Generate an episode $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$, following policy $\pi_\theta$
  for Each step $t = 0, \ldots, T - 1$
    1: Estimate $\nabla_\theta J^*_\text{PG}$ and $\nabla_\theta D_{\text{MMD}}(p_\theta(\tau), Q)$
    3: Update policy parameters via gradient ascent $\theta \leftarrow \theta + \eta(\nabla_\theta J^*_\text{PG} + \alpha \nabla_\theta D_{\text{MMD}}(p_\theta(\tau), Q))$
  end for
until convergence

Output: policy $p_\theta$

Algorithm 6.1: MMD-based Diversity-Inducing Policy

where $\tau_p, \tau'_p$ i.i.d. $\sim p_\theta(\tau)$ and $\tau_q, \tau'_q$ i.i.d. $\sim q_*(\tau)$. The last term only involves the distribution $q_*(\tau) \in Q$ that has no dependence on $\theta$. The gradient term can be estimated by linear combinations of the $\nabla_\theta \log p_\theta(\tau)$ involving the kernel between sample trajectories from the policy $\pi_\theta$ and with a set of specified trajectories from $Q$. It is well-known (Sutton et al., 2000) that the gradient of the score function of the trajectory distribution $\nabla_\theta \log p_\theta(\tau_p)$ does not require the dynamics model and can be expressed in terms of the score function of the policy network $(\nabla_\theta \log p_\theta(\tau) = \sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(a_t|s_t))$.

We now have all the machinery in place to augment any existing policy gradient method with a diversity inducing term. Next, we will specify the basic algorithm for finding a policy that is diverse with respect to some specified set of trajectory distributions and then introduce an algorithm that leverages this to find a set of diverse policies.

### 6.3.3 Finding Multiple Diverse Policies

Our goal is to find a collection of policies that perform optimally or near-optimally and are diverse in terms of the distributions over trajectories that they induce. In Algorithm 6.1, we state how the diversity inducing term $D_{\text{MMD}}$ is used in order to learn a single policy that is distinct with respect to a specified set of distributions over trajectories $Q$. In Algorithm 6.2, we iteratively apply Algorithm 6.1 to find the desired set of $N$ distinct policies (agents): The first policy is learned without any diversity term because there is no set of known policies to begin with. Subsequent policies are

99
Input: Number of policies $N$, MDP $\{S, A, p, r\}$, policy gradient objective $J_{PG}^*$, learning rate $\eta$, diversity parameters $\alpha_1, ..., N$
Collection of known policies $P_{known} = \emptyset$
for $n = 1$ to $N$ do
  1: Find a policy $p_n$ that is distinct from the current set of known policies $P_{known}$:
     $p_n \leftarrow$ Algorithm 6.1($P_{known}$, MDP, $J^*_PG$, $\eta$, $\alpha_n$)
  2: Add $p_n$ to the set of known policies $P_{known}$:
     $P_{known} \leftarrow P_{known} \cup p_n$
end for
Output: Set of policies $P_{known}$

Algorithm 6.2: DIPG

learned by initializing random policy parameters and training with the augmented objective function that encourages diversity with respect to previously discovered policies. The strength of the diversity parameter $\alpha$ can be varied (e.g. to seek diverse policies more aggressively as more policies are discovered).

6.3.4 DIPG Extension: Batch Off-Policy Setting

Batch reinforcement learning ([Lange et al., 2012]) aims to learn policies from a fixed set of previously-collected trajectories. It is common in domains such as medicine, dialogue management, and industrial plant control where logged data are plentiful but exploration is expensive or infeasible.

We now extend our DIPG framework to this batch setting. In the on-policy case, we defined the DIPG diversity term as a kernel over the distribution of trajectories induced by different policies. However, in the batch case, trajectories cannot be simulated from the learned policies. Thus, we instead define the diversity as a kernel over the likelihoods of specific (observed) trajectories in the batch with respect to a policy.

Specifically, let $T = \{\tau_i\}_{i=1}^I$ be a batch of $I$ trajectories. We use $p(T|\pi) \in \mathbb{R}^I$ to indicate a vector where the $i$th coordinate equals the probability of the $i$th trajectory under the policy $\pi$ i.e. $p_i(T|\pi) = p(\tau_i|\pi)$. Now, the diversity term can be defined in an analogous fashion to the on-policy case where we compare our policy being optimized $\pi_\theta$ to the previous policies $\{q_1, ..., q_M\}$:
\[ D^{\text{T}}_{\text{BATCH}}(\pi_{\theta}, Q) = \min_{m \in \{1, \ldots, M\}} k(p(T|\pi_{\theta}), p(T|q_m)) \]
\[ = k(p(T|\pi_{\theta}), p(T|q^*_{m})) \]

We also require a measure of quality for the policies. Levine and Koltun (2013) note that gradient-based optimization of importance sampling estimates is difficult with complex policies and long rollouts. We suggest a surrogate that is equal to the sum of the likelihoods of each trajectory in the batch \( J^{\text{T}}_{\text{Surrogate}}(\theta) = \sum_{i=1}^{I} p(\tau_i|\theta) \). This surrogate is more robust to optimize and encourages ‘safe’ behavior from the agent, a desirable feature in the healthcare setting.

\[ J^{\text{BATCH}}(\theta) = J^{\text{T}}_{\text{Surrogate}}(\theta) + \alpha D^{\text{T}}_{\text{BATCH}}(\pi_{\theta}, Q) \]

While we optimize with respect to this surrogate, in our results, we still report on the value of the policy with respect to a standard importance sampling-based estimator (CWPDIS, from Thomas (2015)).

6.4 Experimental Setup

In the implementation of our on-policy algorithm, we augment Proximal Policy Optimization (PPO) (Schulman et al., 2017) with the diversity inducing term (we refer to it as DIPG-PPO). We set the maximum number of steps taken by all our baseline algorithms to be 1 million and consequently set the maximum number of steps for each of the \( N \) policies in DIPG-PPO to 0.2 million. Since we choose \( N < 5 \) for all our testing environments, the total number of steps to learn all \( N \) policies is necessarily less than 1 million. In these experiments we set \( \alpha_n = 1.0 \) across all environments and iterations \( n^* \). For the MMD kernel, we choose the Gaussian kernel with bandwidth set to 1. Each algorithm is run 3 times to assess variability in performance, except for RR-PPO which does not have any inherent diversity component, so we run it with 10 random restarts to help it make up for this disadvantage.

*The performance of our algorithm may improve when this parameter is tuned based on the environment or which of the \( N \) policies is being learned*
6.4.1 Environments

Synthetic Environments We illustrate qualitative aspects of our approach on 2-dimensional navigational tasks.

*Multi-goal environment* of Haarnoja et al. (2017) is solved by reaching one of four symmetrically placed goals in a continuous 2-D world. An ideal collection of diverse policies would include policies that solve the task via reaching different goals in different parts of the state space.

*Asymmetric goals* environment has one goal that is closer to the starting point and easier to reach than the second. While most agents find the region closest to the initial position, a distinct collection of policies solves the environment by also reaching the goal further away. We use this environment to explore the case where there exist a slightly sub-optimal solution that is clearly distinct from the optimal one.

Benchmark Environments We evaluated the performance of our algorithm on standard benchmark environments:

*Reacher, Ant, Humanoid* and *Humanoid Flag Run* (Schulman et al., 2017; Ellenberge, 2019).

Clinical batch Hypotension batch data is built from a cohort of patients from MIMIC critical care data set (Johnson et al., 2017). In this Chapter, we aim to learn multiple treatment strategies using off-policy methods. In the MDP formulation of this domain, the agent can choose to give a discretized set of 36 actions (corresponding to all combinations of six categories of IV fluid and vasopressor dosage volumes). The reward is based on blood-pressure with linear penalties for blood pressure that is either higher or lower than the target range.

6.4.2 Algorithm and Baselines

We compare our approach [DIPG-PPO] to the following other algorithms that can encourage diverse behavior:

*Random Restarts* [RR-PPO]: Independent runs of PPO without the diversity inducing regularization. Due to variability in initializations and experience, we obtain a collection of policies that are not exactly the same.

*Stein Variational Policy Gradient* [SVPG]: Liu et al. (2017) use functional gradient descent to compute a point-based posterior distribution over the policy parameter space. We report results for a collection of 8 agents, and for comparison also on a single agent’s performance.
Figure 6.1: A comparison of trajectories in the 2-D navigation tasks shows that DIPG-PPO (with $N = 4$ for Multi-Goal and $N = 2$ for Asymmetric Goals) produces near-optimal trajectories to reach each goal. Soft-Q takes sub-optimal paths to the goal regions and SVPG generally fails to solve the environments.

Deep Energy-Based Policy [Soft-Q]: Haarnoja et al. (2017) present a way to train a single stochastic policy that is encouraged to have high entropy on the probability of the action (given state). The higher entropy encourages the agent to choose actions that are diverse. We try Soft-Q learning with the default entropy regularization parameter of 1 as well as variants with 0.5 and 0. We choose the smaller (and zero) values of the regularization parameter to see the effect of the tradeoff between diversity and quality in this algorithm.

6.5 Results

We evaluate the performance based on both quality and similarity scores (figure 6.2). The quality is evaluated by averaging the returns coming from rollouts of the final stochastic policy (or collection of policies) and the similarity score comes from the kernel used to measure similarity between trajectories (smaller kernel values indicate higher diversity). Additional results can be found in our supplement$^\dagger$.

DIPG method does not sacrifice quality Figure 6.2 shows that DIPG-PPO and RR-PPO generally provide the highest quality policies. The baseline algorithms designed for finding a diverse set of policies (SVPG and Soft-Q) have significantly worse quality performance (SVPG in particular) and find policies that lead to high variability in the rewards even though each individual policy in SVPG and Soft-Q is trained for longer (1 million steps) as compared to RR-PPO and DIPG-PPO.

$^\dagger$https://tinyurl.com/ijai2019-DIPG
Figure 6.2: A comparison of the kernel-based similarity between trajectories and the average return across different algorithms reveals that DIPG-PPO and RR-PPO consistently give the largest returns while achieving low similarity. The colors are consistent with those in figure 6.1.

Figure 6.3: We show the probability of taking actions throughout the duration of a single ICU stay. While these agents all have roughly equal value, they exhibit different emphasis on treatment styles as can be seen by the variability in action probabilities.
(0.2 million steps). Such variability in the distribution of returns suggests that the diversity in SVPG and Soft-Q comes not only at the cost of quality, but importantly (especially in the clinical setting) at the cost of consistency of quality returns. It must be emphasized that diversity in policies is only meaningful if they are beyond some threshold of quality and are able to ‘solve’ a domain consistently.

**DIPG method finds meaningfully diverse policies with fewer runs**  The diversity information in figure 6.2 shows that SVPG and Soft-Q algorithms give policies that are quite diverse (comparison of pairwise trajectories shows negligible overlap). Since these collections of stochastic policies fail to provide high rewards in the environments, the diversity in the trajectory distribution that is induced is of little value. The DIPG-PPO ($N = 2$ and $N = 4$) and RR-PPO collections ($N = 10$) are collections of policies that essentially ‘solve’ these environments and discover the set of ways these environments were designed to be solved (see figure 6.1). In comparison to RR-PPO, DIPG-PPO requires many fewer agents to discover an appropriate collection of diverse agents.

**Clinical Example: A collection of DIPG policies reveals different choices of treatment in the ICU.**  We learned four policies through our off-policy extension of DIPG using real data from the ICU and evaluated their performance using the CWPDIS importance sampling estimator (Thomas, 2015). Not only were all four learned policies of slightly higher quality ($\sim 9.33$ on held out test data) than the clinician behavior policy ($9.29^\ddagger$), they are also distinct.

Figure 6.3 plots the probabilities of each action for each policy for a single patient over their ICU stay duration. The discrete actions correspond to different combinations of vasopressor and IV fluid dosages. Agent 1 has a near-deterministic policy with an emphasis on a single treatment that is a combination of a low vasopressor dosage and a medium fluid dosage. Agents 2 and 3 are more aggressive than other agents, giving weights to medium and high vasopressor dosages as well. All agents emphasize taking action early rather than later in the duration (by which time hopefully the patient is in a stable state). Importantly, all of these policies cannot be differentiated by value alone; thus they form a collection to be presented to clinicians for further review of potentially valuable options.

$^\ddagger$The scale of these values can be understood by the reward function which is defined to always be between 0 and 1 (with 1 corresponding to normal blood pressure).
6.6 Related Work

Existing approaches to find a set of diverse policies involve notions of diversity that are not aligned with the kind of efficient exploration-amongst-reasonable-options settings we described earlier. Liu et al. (2017) characterize the uncertainty over policies via computing a particle-based posterior over policy parameters, but differences in policy parameters may not result in qualitatively different behavior (especially in over-parameterized neural architectures). Haarnoja et al. (2017) encourage diversity in a single stochastic policy network via encouraging high entropy distribution of actions (conditioned on state), which may result in sub-optimal behavior. Fard and Pineau (2011) seek a single non-deterministic policy that may make multiple decisions at any state, which may be overly restrictive if action choices across states must be correlated to achieve near-optimal performance. Smith et al. (2018) learn a policy over options and can train multiple options (in an off-policy manner) using a rollout from a single option. The distinct options can give rise to behavior that is diverse, however there is no explicit diversity component in the objective and it is unclear how to summarize the kinds of distinct trajectories that are possible and what combination of options leads to the most interesting policies.

In the graph-based planning literature (which is limited to the discrete setting), there is also an interest in finding diverse plans; Srivastava et al. (2007) seek to do this using domain independent distance measures for evaluating diversity of plans whereas Sohrabi et al. (2016) first generate a large set of high quality plans and then use clustering to identify a diverse set of representative plans that can be used for further analysis. Our motivation for seeking diverse policies in the reinforcement learning setting is aligned most closely with the end goal of Sohrabi et al. (2016): presenting a diverse set of representative solutions as a tool for hypothesis generation and to discover specific directions of interest for further inquiry. In our case, we believe that multiple policies learned using data from the healthcare domain can serve to inform clinicians of multiple hypotheses for treatment strategies.

There exists a literature on Bayesian methods for reinforcement learning (Ghavamzadeh et al., 2015). These approaches constitute an area of active research as they promise to characterize uncertainty and incorporate prior/expert knowledge. However, these approaches do not systematically attempt to identify distinct policies.
6.7 Discussion

Our DIPG algorithm successfully finds multiple goals in an optimal or near-optimal manner whereas baseline approaches are either unable to reach multiple goals or do so sub-optimally. The poor task performance of SVPG could be due to the difficulties of performing functional gradient descent over a high-dimensional parameter space. However, even if those difficulties were to be overcome, the search for diversity in the space of neural network parameters does not correspond directly to any meaningful notion of diversity in the trajectories. The Soft-Q algorithm exhibits some meaningful diversity in the policies it finds (e.g. reaching different goals in the multi-goal environment), however, there is unnecessary stochasticity in the actions, leading to sub-optimal policies. Unlike the baselines, random restarts do result in near-optimal in-task performance. However, the diversity of the policies is at the mercy of the basins of attraction of each local optima (based on the initialization and subsequent experience). In contrast, DIPG-PPO successfully identifies a collection of distinct policies consistently and efficiently (that is, with few runs).

While we have focused on certain notions of distinctness here (state visits, action counts), our approach extends easily to other measures of distinctness as measured by alternative kernels (see Chen et al. (2018) and Gretton et al. (2012) for novel kernels and discussion). Whereas the full trajectory may be needed for training the RL agent, we can capture (MMD-based) distinctness more generally using an arbitrary function of the trajectory. For example, we could group patients into clinically meaningful clusters which can be used to define a kernel for measuring diversity. There are also opportunities for incorporating more efficient search algorithms than gradient descent (e.g. Toussaint and Lopes (2017)).

While we have focused on the task of returning policies as possible options to a human user, another use-case could be for situations in which we have a cheap, low fidelity simulator and a more expensive, high-fidelity simulator. In this case, the distinct trajectories from the low-fidelity simulator could be used as seeds for deep exploration in the more expensive simulator. Diverse policies can help manage the exploration-exploitation trade-off (Bellemare et al., 2016; Fortunato et al., 2017; Fu et al., 2017).

6.8 Conclusion

We presented an approach for identifying a collection of near-optimal policies with significantly different distributions of trajectories. Our MMD-based regularizer can be applied to the distribution
of any statistic of the trajectories—state visits, action counts, state-action combinations—and can also be easily incorporated into any policy-gradient method. We applied these to several standard benchmarks and developed an off-policy extension to identify meaningfully different treatment options from observational clinical data. The ability to find these diverse policies may be useful when the agent does not have complete information about the task, and for presenting a set of potentially reasonable options to a downstream human or system, who can use that information to efficiently choose amongst reasonable options.
Multiple Solutions in RL: Diverse, Safe and (High) Quality Policies for Off-Policy RL

The on-policy RL setting of Chapter 6 is widely used in the robotics world, gaming applications and other continuous control tasks, however, in more critical domains (such as healthcare), there is typically only observational (batch) data available to indicate the performance of some behavior policy. In order to learn reasonable policies from this batch of data we resort to off-policy RL techniques. Our goal in this Chapter is to find a diverse set of policies through batch data in order to responsibly explore the space of possible policies (to present later perhaps as part of a recommender system). To that end, we introduce a method for jointly optimizing a collection of policies under the batch setting in order to find a diverse set of high-quality policies whose recommended actions are supported by the data.

7.1 Problem Description

Patients in the intensive care unit (ICU) are among the sickest in the hospital, and require many different types of interventions to control and respond to their unstable physiological conditions. For instance, antibiotics are given to control infections (Ibrahim et al., 2000), anticoagulants are given to dialysis patients to prevent thrombosis (Berbece and Richardson, 2006), and fluid bolus therapy (Glassford et al., 2014) and vasopressors (Havel et al., 2011) are given to treat hypotension and shock.
Patients with the highest acuity may be given more aggressive and invasive interventions such as mechanical ventilation (Esteban et al., 2002) as well. Due to the complexity and high morbidity of patients in the ICU, it is often challenging for physicians to determine safe and effective treatments in a timely fashion, creating an opportunity for clinical decision support tools to help make intervention recommendations.

There has been substantial interest in using machine learning to learn treatment strategies from observational data. Reinforcement learning (RL) is a branch of machine learning focused on learning optimal sequences of actions to take to maximize a numerical reward signal, and is the most common approach for tackling such treatment optimization problems. Examples of applications of RL in healthcare include Komorowski et al. (2018) who use it to learn optimal treatments for septic patients, Shortreed et al. (2011) who use RL to make drug recommendations for patients with schizophrenia, Prasad et al. (2017) who use RL to learn strategies for weaning ICU patients off ventilators, and Nemati et al. (2016) who learn optimal heparin dosing in the ICU.

However, as noted in Gottesman et al. (2019), it can be very hard to quantify how good a learned treatment policy is solely from observational data. In this Chapter, rather than trying to find a single optimal treatment strategy, we focus on identifying different types of treatment strategies that already exist in the observational data. Identifying such diverse policies can provide insights into multiple versions of treatments that may be of similar efficacy, and it also provides a step toward providing personalized recommendations by creating a space of reasonable treatment options. Identifying potential treatment policies from the variation in current practice—that is, actions currently taken by clinicians—also ensures that our options are likely to be safe, or at least as safe as current practice.

**Technical Significance** Our work makes several technical contributions in this area of identifying diverse policies in reinforcement learning. We propose an objective function that simultaneously balances these three distinct goals of interest: learning policies that are diverse, safe, and high-quality.

Our objective function contains two terms that correspond to the goals of learning diverse and high-quality policies, as well as a safety constraint that forces the learned policy to stay near the observed behavior data. During training we jointly learn several policies through a joint optimization.

In order to evaluate the value of our learned policies, we turn to methods from off-policy value evaluation. In particular, we will make use of a specific form of importance sampling estimator that re-weights observed rewards by how likely the observed actions were under the observed behavior.
and evaluation policies.

**Clinical Relevance**  In this Chapter we focus on the challenging task of management of hypotensive patients in the ICU. Previous work has found that hypotension is associated with overall higher morbidity and mortality in diverse populations, including populations with sepsis (Maheshwari et al., 2018) and populations in the emergency department (Jones et al., 2006). However, despite the importance of addressing this problem, ICU decision making for hypotension management is not standardized, and treating these patients effectively is challenging. Typical treatment includes fluid bolus therapy, and administration of vasopressors, which both try to raise blood pressure and correct hypoperfusion. The choice of bolus size and timing, as well as which vasopressor to use and in what doses is not well understood, although it has been studied extensively. For instance, Girkar et al. (2018) specifically looks at predicting the efficacy of fluid therapy in order to identify groups of patients where it is most likely to be effective.

The overarching theme for how patients are currently treated is that it’s done safely and conservatively to avoid iatrogenic harm to patients. Thus it’s of the utmost importance to ensure that any decision support tool to help improve physician treatment decisions does not brashly recommend treatments that would never be performed in practice. When leveraging reinforcement learning in healthcare, it is often sensible to assume that most actions performed were reasonable, and so it is desirable to not deviate too far from these accepted medical practices. In this Chapter, we aim to identify different treatment strategies for hypotension and compare the observed physician behavior policy with policies learned by an RL algorithm.

### 7.2 Background

In this section, we provide some background on reinforcement learning, specifically as it applies to off-policy settings where only observational data is available.

**Off-Policy Evaluation**  The off-policy RL framework applies when trajectories are collected from a behavior policy that is distinct from the policy being trained, e.g., using observations (batches) from clinician behavior to optimize an agent’s behavior. In the batch setting, there are model-based and importance sampling approaches to estimating the value of a policy. Model-based approaches learn a dynamics model for the transitions and use those to simulate outcomes of a policy in order to estimate its value. Importance sampling approaches appropriately re-weight existing batch data to
estimate the value for a different policy. In this Chapter, we will use an importance sampling estimator (CWPDIS) from Thomas (2015).

**Importance Sampling** Traditional RL domains are able to sample from the true environment (e.g. playing atari games) but due to the critical nature of clinical work, we need to use alternative methods to estimate the value of new policies without directly implementing them. The basic idea behind importance sampling allows us to do exactly that.

In order to estimate an expectation $f(x)$ over a desired distribution $p$, we use samples from a distribution $q$ that we can sample from. The desired expectation can be re-written as

$$
\mathbb{E}_p[f(x)] = \mathbb{E}_q \left[ \frac{f(X)p(X)}{q(X)} \right] \quad (7.1)
$$

A vanilla estimate of the desired expectation uses $N$ weighted samples from the distribution $q$.

$$
\mathbb{E}_p[f(x)] \approx \frac{1}{N} \sum_{n=1}^{N} f(X_i)p(X_i) \frac{p(X_i)}{q(X_i)}, \quad X_i \sim q \quad (7.2)
$$

The weights are defined to be the ratio of probabilities $w_i = \frac{p(X_i)}{q(X_i)}$. However, even though $N$ samples are used in this expectation, there may be some weights $w_i$ that are much larger than the others and therefore dominate the expectation term. It is therefore standard practice to keep track of the effective sample size (ESS) which is given by

$$
\text{ESS} = \frac{\left( \sum_{n=1}^{N} w_n \right)^2}{\sum_{n=1}^{N} w_n^2} \quad (7.3)
$$

A small value of ESS could indicate lack of robustness of the importance sampling estimator. In this Chapter, we use the ESS as an indicator of how ‘safe’ the actions proposed by a policy (which we would ideally want to sample from) are in comparison to the behavior policy (the distribution we have samples from).

In this Chapter, we shall seek multiple near-optimal policies that form a high-quality and diverse selection.
7.3 Related Work

The off-policy reinforcement literature generally focuses on safe and efficient learning using off-policy evaluation techniques (Munos et al., 2016; Thomas et al., 2015b; Gottesman et al., 2019). However, there is limited exploration of learning a diverse collection of agents. Additionally, the notion of safety we employ in this work is informed by the assumption that clinicians generally perform well. This is distinct from other notions of safe RL such as those comparing the bounds on different off-policy evaluation metrics (e.g. Thomas et al. (2015a)).

Prior work on diversity involves notions of diversity that are not aligned with the kind of efficient exploration-amongst-safe-options settings we are interested in. Liu et al. (2017), Haarnoja et al. (2017) use notions of diversity that don’t directly compare action probabilities but rather features such as neural network parameter differences or entropy in a single agent’s action probabilities. Smith et al. (2018) learn a policy over options and can train multiple options (in an off-policy manner) using a rollout from a single option. The distinct options can give rise to behavior that is diverse, however there is no explicit diversity component in the objective and it is unclear how to summarize the kinds of distinct trajectories that are possible and what combination of options leads to the most interesting policies.

Our motivation for seeking diverse policies in the reinforcement learning setting is aligned most closely with the end goal of Sohrabi et al. (2016): presenting a diverse set of representative solutions as a tool for hypothesis generation and to discover specific directions of interest for further inquiry. In our case, we believe that multiple policies learned using data from the healthcare domain can serve to inform clinicians of multiple hypotheses for treatment strategies.

Masood and Doshi-Velez (2019) learn diverse policies using a divergence metric between distribution of trajectories induced by policies. Their work focuses largely on on-policy settings where a simulator of the environment is available and collection of policies is learned sequentially rather than jointly.

There are several related papers that tackle similar applications as ours. Komorowski et al. (2018) and Raghu et al. (2017) also use RL to learn fluid and vasopressor treatment strategies, but specifically in septic patients, and their focus is on optimality and not safety and diversity. Girkar et al. (2018) focuses on predicting response to fluid bolus therapy, as the treatment does not work in all cases. There are also many papers that attempt to predict onset of both various kinds of interventions (e.g. Ghassemi et al. (2017)) and onset of hypotension events (e.g. Hatib et al. (2018) and Ghosh et al. (2014)).
Table 7.1: Baseline characteristics of the set of ICU stays used in our experiments.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Overall Cohort (N=9,860)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (25/50/75% quantiles)</td>
<td>57.4, 69.4, 80.5</td>
</tr>
<tr>
<td>Female (%)</td>
<td>47.6%</td>
</tr>
<tr>
<td>Weight in kg (25/50/75% quantiles)</td>
<td>65.0, 77.1, 90.9</td>
</tr>
<tr>
<td>Surgical ICU (%)</td>
<td>48.2%</td>
</tr>
<tr>
<td>Initial SOFA score</td>
<td>2, 4, 7</td>
</tr>
<tr>
<td>Initial OASIS score</td>
<td>27, 33, 39</td>
</tr>
<tr>
<td>Initial SAPS score</td>
<td>15, 19, 22</td>
</tr>
<tr>
<td>Non-white (%)</td>
<td>23.7%</td>
</tr>
<tr>
<td>Emergency Admission</td>
<td>82.4%</td>
</tr>
<tr>
<td>Urgent Admission</td>
<td>1.3%</td>
</tr>
<tr>
<td>Time from Admission to ICU (mean; 25/50/75% quantiles)</td>
<td>48.5; 0.0, 0.1, 26.9</td>
</tr>
</tbody>
</table>

7.4 Cohort

Our data source is the publicly-available MIMIC-III database (Johnson et al., 2016). The full dataset contains static and dynamic information for nearly 60,000 patients treated in the critical care units of Beth-Isreal Deaconess Medical Center (BIDMC) in Boston between 2001-2012. In our work we use version 1.4 of MIMIC-III, released in September 2016. Much of our data processing is reused from original queries from Ghassemi et al. (2017), with some additional processing pulled from the public code released by Komorowski et al. (2018). In Table 7.1, we present some baseline characteristics of the selected cohort.

7.4.1 Cohort Selection

Each observation in our final dataset is a single ICU stay. In some cases it is possible that a single patient appears multiple times if they had multiple ICU stays within a single hospital admission, or if they had multiple admissions to BIDMC during the period of interest.

We started with all non-pediatric patients, filtering to only those with an age of at least 15 on admission. We further filtered to only include patients in MIMIC-III that came from MetaVision, as these were the only patients where we could reliably and easily extract both start and end times for the interventions of interest. Next, we filtered out the bottom and top 1% quantiles in terms of length of stay, so that only ICU stays that were at least 14 hours but no longer than 622 hours were included. Lastly, we filtered to only include ICU stays that had at least three distinct measurements.
of mean arterial pressure (MAP), and at least one MAP below 65, indicating hypotension. This resulted in a cohort of 9,860 ICU stays.

7.4.2 Data Extraction

Our dataset contains 11 static features available on admission to the ICU (or shortly thereafter): age, biological sex, weight, whether the ICU was a surgical ICU, three overall severity scores (SOFA (Vincent et al., 1996), OASIS (Johnson et al., 2013), and SAPS (Le Gall et al., 1993)), race, whether the hospital admission was urgent, whether the hospital admission was an emergency, and hours from hospital admission to ICU admission.

We also have a total of 20 clinical time series variables measured over the course of a patient’s ICU stay. These include 8 vitals: diastolic blood pressure, heart rate, mean arterial pressure (MAP), systolic blood pressure, pulse oximetry, respiration rate, temperature, and urine output. Vitals are typically recorded about once an hour, although in practice they are captured at the bedside continuously. We also include 12 laboratory measurements: fraction of inspired oxygen, blood urea nitrogen, creatinine, glucose, bicarbonate, hematocrit, lactate, magnesium, platelets, potassium, sodium, and white blood cell count. These are typically only measured a few times a day from blood samples drawn from patients.

Lastly, we extracted information on the interventions of interest: fluid therapy and vasopressor therapy. We combine different types of fluids and blood products together when forming our fluid action variable; in particular, we filter to only include common NaCl 0.9% solution, lactated ringer’s, packed red blood cells, fresh frozen plasma, and platelets. We include five different types of vasopressors for the vasopressor action variable: dopamine, epinephrine, norepinephrine, vasopressin, and phenylephrine. We map these five drugs into a common dosage amount based off norepinephrine equivalents, following the preprocessing in Komorowski et al. (2018).

7.4.3 Feature Choices

The state space in our RL formulation consists of the 31 previously listed variables. In addition, for the labs, we include binary indicator variables that denote whether a lab was measured since the last time, as labs are typically informatively missing and the clinical decision to measure a certain lab can be important (Agniel et al., 2018).

We discretize time into 6 hour windows in order to reduce sparsity in the original data so that more time points have a non-null action. We impute any unobserved vital or lab value with the
population median value. Once a variable is observed in a given hospital admission, we then use the last observed measurement of a variable until a new value is measured. The only exception to this is the three blood pressure variables; we use the 25% quantile among all blood pressure values per six hour bin as an aggregate value, rather than using the last blood pressure in each six hour window. We do this because generally clinicians treat patients based on their most recent worst blood pressure value.

We discretize the two types of interventions into a low and high dose, so that there are three possible doses of fluids and three possible amounts of vasopressors in each six hour window (none, low, and high). In total, this results in nine unique actions that may be taken at each decision time. We aggregate the total amount of vasopressors given in each six hour window, using a threshold of 240 mcg/kg within the six hours to break up low vs high doses. For fluids, we only include fluids boluses given at a rate of at least 100mL/hour, and use a threshold of 2L of fluid per six hour window to divide up the low vs high doses.

Lastly, we detail the reward function used. Since our task is explicitly to optimize blood pressure, we use the common target of a MAP of 65. See Figure 7.1 for a visualization of the specific function we used. This reward function has inflection points at 55, 60, and 65. Values 65 and above are considered optimal, and the reward reaches 0 at 28 (the lowest observed MAP in our data). Having a normal value for urine output causes the reward to not penalize moderately hypotensive values. We leave to future work a more thorough investigation of potential reward functions.

After the review period, we will make publicly available all pre-processing, cohort selection, and modeling code.

7.5 Methods

We propose an algorithm for learning a collection \( C \) of diverse, safe and (high) quality policies from batch data \( B \) in Reinforcement Learning (DSQRL). We do this by combining differentiable objectives for diversity \( f^D \) and quality \( f^Q \) with a safety constraint \( C \in S_B \) and jointly optimizing to find an optimal collection \( C^* \).

\[
C^* = \max_{C \in S_B} f^Q(B, C) + \lambda f^D(B, C)
\]  

(7.4)

Below, we describe in detail how the diversity and quality objective are computed and how the safety constraint is applied to the collection of neural network policies we learn in this Chapter.
Figure 7.1: Reward function used in our analysis. The reward depends primarily on MAP, with inflection points at 55, 60, and 65. Values 65 and above are considered optimal.

**Diversity** Each learned policy $\pi$ induces a distribution over the action probabilities given a state; we denote this vector of probabilities $\pi(s)$. We can use a well-established divergence measure such as the symmetric Kullback-Leibler (KL) divergence between two discrete probability distributions to quantify this difference. We define the diversity between two policies as an average of this symmetric KL over all observed states in the batch $B$.

$$\text{Div}_{\pi_i, \pi_j}^B = \mathbb{E}_{s \sim B} \left[ \frac{1}{2} \text{KL}(\pi_i(s) || \pi_j(s)) + \frac{1}{2} \text{KL}(\pi_j(s) || \pi_i(s)) \right]$$

(7.5)

For a collection of policies $C = \{\pi_1, \pi_2, \ldots, \pi_N\}$, we define the diversity measure as the sum of the pairwise diversity measure for pairs that are distinct.

$$f^D(C, B) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{Div}_{\pi_i, \pi_j}^B \quad (7.6)$$

**Safety** We propose adding a safety constraint that defines a mask over allowable actions for a given state. We implement this using a nearest neighbor search to discover the counts of different
actions taken under similar conditions (states) and use a minimum count threshold to define this mask. The Mask\(^B\)(s) ∈ (0, 1\(^\)\(^N\)\(_A\)) is a binary vector of length \(N\_A\), where \(N\_A\) is the number of discrete actions possible. A mask value of one indicates that the corresponding action is ‘safe’. This occurs when there are a sufficient number (greater than \(N\_Cutoff\)) of similar (from \(N\_Nearest\) observations in the batch data \(B\)) observations where that action has been taken.

\[
\text{Safe}^B[\pi](s) = \frac{\pi(s) \odot \text{Mask}^B(s)}{\sum \pi(s) \odot \text{Mask}^B(s)}
\]  

(7.7)

Here, \(\odot\) indicates element-wise multiplication of the action probabilities vector \(\pi(s)\) and the mask \(\text{Mask}(s)\). Note that applying the \(\text{Safe}^B\) operator once has the same effect as applying it any number of times. We can use this criteria to define the set of all safe policies as those that are unchanged under this operator

\[
\mathcal{S}^B = \{\pi \text{ s.t. } \pi = \text{Safe}^B[\pi]\}
\]  

(7.8)

This is the set of policies we are interested in optimizing the quality and diversity over.

**Quality** Levine and Koltun (2013) note that gradient-based optimization of importance sampling estimates is difficult with complex policies and long rollouts. While we always use the CW-PDIS estimator for the evaluation of our policies, we propose using a surrogate to optimize with respect to. Our surrogate is based on replicating the behavior policy and leverages the assumption that clinician behavior is near-optimal. Therefore, trying to emulate it should lead to high-quality policies. Computing this quality term requires first learning a behavior policy (\(\pi\_\text{Behavior}\)) which will typically need to be inferred as the batch data only shows actions that were taken, not the probability of all actions in every state. Assuming a behavior policy \(\pi\_\text{Behavior}\) is learned (e.g., using nearest neighbors over states, the approach we use), we define the quality measure \(D[\pi(s), \pi\_\text{Behavior}(s)]\) for a single policy \(\pi\) and compute its expectation over the observed states. This measure is then summed over individual policies to obtain the quality objective for a collection \((C)\)

\[
f^Q(C, B) = \sum_{i=1}^{N} E_{s \sim B} [D[\pi(s), \pi\_\text{Behavior}(s)]]
\]  

(7.9)

In this Chapter we use the mean-squared-error for \(D[\pi(s), \pi\_\text{Behavior}(s)]\).
7.6 Experimental Setup and Results

In this Section, we first introduce the metrics we use to evaluate DSQRL, and then present our results on a synthetic data task and on the MIMIC hypotension dataset previously introduced.

7.6.1 Evaluation Metrics

We evaluate DSQRL using several quantitative metrics, and compute them on the held-out test set. We estimate the safety of a policy by reporting the effective sample size, as this provides a notion of how many “effective” trajectories this policy is based on. Lastly, we estimate the value of a policy by using the CWPDIS estimator of the policy’s value.

We compare a number of ablations of our proposed approach along with several very simple baselines. The “Random-Uniform” baseline is a simple policy that always proposes all actions with equal probability. The “Random-Empirical” baseline is a policy that always proposes all actions with a fixed probability equal to their observed rate of occurrence in the behavior data.

7.6.2 Synthetic Patient Task

We first construct a synthetic example, broadly inspired by the clinical treatment task. The purpose of this task is to distill the task of finding diverse, safe and high-quality treatment strategies into a simple problem where we fully understand the patient states and the effect of different actions on the state. We designed a task where a population can have one of two diseases (A or B) and there exist multiple (two) treatment strategies for disease A whereas only one appropriate strategy for disease B. In this setting, we wish to learn a collection of policies that learns the diversity of treatment strategies for patients with disease A and learns the single optimum treatment strategy for patients with disease B.

We construct patients with a three-dimensional state to indicate type of disease, severity and time since admission.

The action space consists of four actions: two for the treatment for Disease A, one for Disease B, and a ‘do nothing’ action:

1. None: Do nothing
2. Surgery A: A fast-acting (but also invasive) medical procedure for the treatment of Disease A
3. Drug A: A slow-acting (but less invasive) drug for the treatment of Disease A
Table 7.2: Quantitative results from the synthetic patient dataset.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Safety (ESS)</th>
<th>Quality (CWPDIS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed Behavior</td>
<td>2000</td>
<td>0.647</td>
</tr>
<tr>
<td>Random-Uniform</td>
<td>17.35</td>
<td>0.205</td>
</tr>
<tr>
<td>Random-Empirical</td>
<td>7.69</td>
<td>0.357</td>
</tr>
<tr>
<td>Safety</td>
<td>Quality</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>369</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>4.83</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>426</td>
</tr>
</tbody>
</table>

4. Drug B: A ‘slow-acting’ drug designed for the treatment of Disease B

The transition dynamics in this setting are deterministic. There is no effect of the None action. The Surgery A action reduces the full severity of Disease A to zero if given, however when given to a Patient with Disease B causes a significant increase in the severity of the patient’s disease. Both Drug A and Drug B reduce the severity of Disease A and B (respectively) by small amounts but cause a small increase in severity when it’s the wrong drug for the disease. The reward function is defined as the negative of the change in severity of the patient state.

Table 7.2 shows quantitative results for the Safety and Quality metrics on a test set of 2000 synthetic patients after training on a set of 2000 observed trajectories (each of length 5). The Random-Uniform and Random-Empirical baselines serve to provide useful lower bounds for the metrics of interest and these baselines are obtained readily from the data without the use of any RL algorithm. We find that both instances where we use the safety constraints give significantly higher results on the safety metric (ESS is two orders of magnitude higher). Similarly, both variations where the quality term is used for optimization results in higher CWPDIS estimates than without. Overall, the safety constraint (even without the quality term) results in higher quality policies than the random variants. This is indicative that staying close to the observed policy is also related with good performance. The combination of the safety constraint and quality objective outperforms the other variants considered.

Figure 7.2 demonstrates that the collection of agents trained by our algorithm DSQRL learns the two treatment strategies for Disease A and the unique optimal strategy for Disease B as desired.
Figure 7.2: Two synthetic patient trajectories along with the recommended actions by the two agents in one run of our algorithm (DSQRL). In the left pane the two agents suggest different treatment strategies that ultimately yield the same return of 0.60 for one type of patient (with disease A), while in the right pane the two agents agree on an identical unique optimal treatment for this type of patient (with disease B).

<table>
<thead>
<tr>
<th>Policy</th>
<th>Safety</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed Behavior</td>
<td>5.001</td>
<td>13.6</td>
</tr>
<tr>
<td>Random (Uniform)</td>
<td>2.3</td>
<td>7.44</td>
</tr>
<tr>
<td>Random (Empirical)</td>
<td>5.36</td>
<td>8.92</td>
</tr>
<tr>
<td>DSQRL</td>
<td>48.5</td>
<td>11.4</td>
</tr>
</tbody>
</table>

Table 7.3: Results from the hypotension dataset.

7.6.3 Hypotension Management Task

We now present results from running DSQRL on the hypotension dataset previously described. Overall, we find that DSQRL is able to learn reasonably safe and high-quality policies, and that differ from each other and from the behavior policy.

In Table 7.3, we show the quantitative results of DSQRL, along with the observed physician behavior policy and the two random policies. DSQRL learns a collection of policies that are almost as high in value as the behavior policy, and that have an ESS of an order of magnitude higher than the random policies.

Figure 7.3 shows qualitative results from DSQRL along with the observed physician behavior policy. Each column of the figure corresponds to one of the nine possible actions, and each row is a different policy. The figure shows a histogram of action probabilities for each type of action and for each policy. For instance, the top left pane shows the distribution of probabilities assigned by the
behavior policy of taking no action (Fluid 0, Vaso 0); this histogram has a peak at 1, as it is relatively common (\(\sim 85\%\)) for physicians to not perform any of our predetermined actions at a specific time.

There are several interesting findings in the figure that display the diversity of policies learned by DSQRL. The first finding of note is that the policies learned by DSQRL look qualitatively different than the behavior policy. While the behavior policy often has high probabilities of no action, the RL agent policies tend to prefer more actions, in general. This is especially apparent in the last few columns (e.g. actions with Vaso 2 or the high dose of vasopressors).

In addition, we see that the individual agent policies also exhibit differences between each other. For instance, the agent labeled 1 in the third row of the figure has much higher action probabilities for the high dose of both fluids and vasopressor action (last column) than any of the other agents. In general all of the actions involving a high dose of fluids appear to differ between the four agents. The fact that there is more diversity between agents in the higher doses and more aggressive actions perhaps implies that there are interesting findings here to explore in future collaboration with clinical partners.

### 7.7 Discussion

In this paper we introduced DSQRL, a reinforcement learning approach for identifying diverse, safe, and high quality treatment strategies from observational data. We applied DSQRL to the problem of learning interventions in hypotension, and found that it was able to learn diverse policies of high quality that did not stray too far from the behavior policy.

Despite the promise of this approach, there are a number of limitations that we acknowledge and may motivate future work. One limitation is the inherent assumption of Markovianity. In practice, data from an ICU is probably not truly Markov, and so it may be worth incorporating features that summarize the past, or addressing the potential partial observability problem through other means. Another limitation is the need for hyperparameter grid search to determine how much weight in the objective to give to each of the competing goals, i.e. how to choose the explicit tradeoff factor between diversity and quality. Lastly, the particular way we framed the problem of hypotension management as an RL problem might be improved in the future. Both the choice of six hour windows and the binning of treatments into only two doses of low and high may be overly coarse. It would be interesting in future work to explore how and why different types of vasopressors are given, especially settings where more than one are given (e.g. vasopressin, which is often combined
with another drug like norepinephrine). Our choice of reward might be improved in future work as well; there are a myriad of factors that can affect a patient’s blood pressure and a single fixed target value may be somewhat overly simple. Furthermore, blood pressure is in some sense only an intermediate proxy for the types of patient outcomes that are ultimately of true interest, although those can be harder to measure and pose more challenging credit assignment problems.

We feel that the ideas presented in this Chapter around diversity and safety are under-explored in the reinforcement learning for healthcare literature. The proposed DSQRL works well to find a diverse collection of safe and high-quality policies which were used to discover insights for hypotension management that could be further explored. We hope to work closely with clinical colleagues in the future to do chart reviews and investigate on more individual cases the types of scenarios where each policy might be preferred. It is our hope that eventually these methods might be of practical value to help providers actually improve treatment strategies for critically ill patients.
Figure 7.3: Overall action probabilities. Each column corresponds to one of the nine actions in our action space. The top row shows the physician behavior probabilities aggregated across all patients in the test set. The bottom four rows show probabilities from the four different agents from one run of our algorithm.
Conclusion & Future Work

We explored the presence of multiple solutions in Bayesian non-negative matrix factorization as well as reinforcement learning and demonstrated that existing approaches are often unable to discover meaningful diversity (which typically exists). In all the works presented here, we have questioned existing notions of diversity and often had to reformulate according to novel metrics (e.g. angular distance between NMFs and MMD-based distance between trajectory distributions). In some cases however, diversity is best captured through performance on some end-goal e.g. classifier accuracy based on latent representations from different NMFs or captured more qualitatively through visualization of trajectories induced by policies.

The algorithms presented in this thesis are all designed to find collections of multiple solutions aimed at capturing this meaningful diversity. While our proposed algorithms rely on various specific properties and techniques, we find that under a more broad lens their success can be attributed to how they facilitate incorporating existing state of the art techniques and employ a flexible modeling framework. For example, in Chapter 5 we present an algorithm that allows users to pick any NMF objective of their choice and use any NMF algorithm to generate candidate particles. We facilitate the choice of state of the art NMF (including non-Bayesian) algorithms which are widely used in applied work instead of restricting the user to typical Bayesian approaches which are often limited due to their modeling assumptions and/or inference techniques. Similarly, in Chapter 6, we propose a regularization term that can be augmented to any policy gradient technique and a diversity term that
can be any function of the trajectories induced by the policy. This is in contrast to existing baseline approaches which typically limit the notion of diversity to entropy in the action space and limit the learning algorithm to one that encourages diversity but does not otherwise achieve state of the art performance.

The different algorithms proposed have helped identify many specific areas for future work that are discussed in the respective chapters introducing the algorithms. Below, we present an overview of important future directions for the broader topic of multiple solutions in machine learning.

- Tuning trade-off between diversity and quality: In many cases we find that there is a tradeoff between the diversity of a collection of solutions and its overall quality. Future work that develops guidelines for how to manage this tradeoff would be very useful for practitioners.

- Sequential versus joint optimization: While sequentially finding multiple solutions can have computational benefits, it may lead to sub-optimal collections as the objective function can only be defined on the current size of the collection. On the other hand, joint optimization may be more suitable to finding a collection but may lead to computational difficulties with optimization and it may be difficult/restrictive to decide in advance on the size of the collection desired.

- Role of transfer learning: We saw a successful application of transfer learning in the NMF exploration. In reinforcement and other machine learning problems in general, there may be many other instances where different datasets/domains have a lot of common structure that can be leveraged by learning algorithms.

Perhaps most importantly, we find that the machine learning framework of presenting a collection of solutions to domain experts in order to evaluate over a rich set of criteria to help decide the ‘best’ solution requires more promotion within the academic community. Working closely between practitioners and machine learning experts to help refine this framework and demonstrate its usefulness would be valuable.
This appendix contains supplementary figures and tables to accompany the work presented in Chapter 2. We show the performance of our algorithm as well as baselines on both the large and small dataset across a range of quality and diversity metrics.
### Table A.1: Comparison of Performance of Sampling Method: Likelihood

<table>
<thead>
<tr>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One Mode</td>
<td>-1.74e+01</td>
<td>-1.74e+01</td>
<td>-1.12e+03</td>
<td>-2.30e+02</td>
<td>-2.38e+01</td>
</tr>
<tr>
<td></td>
<td>-1.84e+01</td>
<td>-1.84e+01</td>
<td>-1.22e+03</td>
<td>-2.34e+01</td>
<td>-2.52e+01</td>
</tr>
<tr>
<td></td>
<td>-1.78e+01</td>
<td>-1.78e+01</td>
<td>-1.09e+03</td>
<td>-2.34e+02</td>
<td>-4.48e+01</td>
</tr>
<tr>
<td></td>
<td>-1.62e+01</td>
<td>-1.62e+01</td>
<td>-1.03e+03</td>
<td>-2.37e+01</td>
<td>-2.54e+01</td>
</tr>
</tbody>
</table>

| Two Modes | -1.72e+01 | -1.70e+01       | -2.17e+03                  | -2.81e+02 | -2.38e+01 | -1.88e+01 |
|           | -1.80e+01 | -1.80e+01       | -2.14e+03                  | -2.86e+01 | -4.02e+01 | -1.90e+01 |
|           | -1.78e+01 | -1.78e+01       | -2.18e+03                  | -2.79e+01 | -4.48e+01 | -1.94e+01 |
|           | -1.62e+01 | -1.62e+01       | -2.07e+03                  | -2.77e+01 | -2.54e+01 | -1.88e+01 |

| Infinite Modes | -2.20e+01 | -2.18e+01       | -1.20e+05                  | -3.20e+02 | -4.21e+01 | -2.09e+01 |
| Infinite Modes | -2.39e+01 | -2.39e+01       | -1.22e+05                  | -3.27e+01 | -4.30e+01 | -2.20e+01 |
| Infinite Modes | -2.18e+01 | -2.18e+01       | -1.18e+05                  | -3.10e+01 | -4.27e+01 | -2.09e+01 |
| Infinite Modes | -1.94e+01 | -1.94e+01       | -1.10e+05                  | -3.00e+01 | -4.10e+01 | -1.90e+01 |
Table A.2: Comparison of Performance of Sampling Method: IAT

<table>
<thead>
<tr>
<th></th>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One Mode IAT</strong></td>
<td>1.009</td>
<td>1.009</td>
<td>1.014</td>
<td>1.014</td>
<td>728.498</td>
<td>837.699</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.02</td>
<td>1.02</td>
<td>884.883</td>
<td>822.669</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.02</td>
<td>1.02</td>
<td>727.725</td>
<td>727.725</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.015</td>
<td>1.015</td>
<td>584.909</td>
<td>584.909</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>830.907</td>
<td>830.907</td>
</tr>
<tr>
<td><strong>Two Modes IAT</strong></td>
<td>1.009</td>
<td>1.004</td>
<td>1.013</td>
<td>1.013</td>
<td>677.867</td>
<td>813.753</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.00</td>
<td>1.013</td>
<td>1.013</td>
<td>645.436</td>
<td>716.395</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.013</td>
<td>1.013</td>
<td>686.236</td>
<td>716.395</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.0175</td>
<td>1.0175</td>
<td>719.475</td>
<td>726.511</td>
</tr>
<tr>
<td><strong>Infinite Modes IAT</strong></td>
<td>1.008</td>
<td>1.008</td>
<td>1.009</td>
<td>1.009</td>
<td>313.22</td>
<td>710.851</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>390.987</td>
<td>686.975</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>306.475</td>
<td>706.795</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>320.795</td>
<td>744.81</td>
</tr>
</tbody>
</table>

129
<table>
<thead>
<tr>
<th></th>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Angle Max</strong></td>
<td>0.185</td>
<td>0.185</td>
<td>34.763</td>
<td>4.184</td>
<td>0.289</td>
<td>0.221</td>
</tr>
<tr>
<td><strong>One Mode</strong></td>
<td>0.18</td>
<td>0.18</td>
<td>28.359</td>
<td>3.998</td>
<td>0.18</td>
<td>0.11</td>
</tr>
<tr>
<td><strong>Random Restarts</strong></td>
<td>0.19</td>
<td>0.19</td>
<td>19.169</td>
<td>4.09</td>
<td>0.285</td>
<td>0.204</td>
</tr>
<tr>
<td><strong>Random Restarts (filtered)</strong></td>
<td>0.09</td>
<td>0.09</td>
<td>9.624</td>
<td>4.98</td>
<td>0.2975</td>
<td>0.2275</td>
</tr>
<tr>
<td><strong>Gibbs</strong></td>
<td>0.01</td>
<td>0.01</td>
<td>0.091</td>
<td>0.01</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>HMC</strong></td>
<td>0.19</td>
<td>0.19</td>
<td>0.289</td>
<td>0.221</td>
<td>0.11</td>
<td>0.11</td>
</tr>
</tbody>
</table>

| **MM ℓ₁ Max**  | 0.0      | 0.0       | 0.697          | 0.091                     | 0.01        | 0.0         |
| **Two Modes**  | 0.0      | 0.0       | 0.79           | 0.09                      | 0.01        | 0.0         |
| **Random Restarts** | 0.0      | 0.0       | 0.79           | 0.09                      | 0.01        | 0.0         |
| **Random Restarts (filtered)** | 0.0      | 0.0       | 0.79           | 0.09                      | 0.01        | 0.0         |
| **Gibbs**      | 0.0      | 0.0       | 0.79           | 0.09                      | 0.01        | 0.0         |
| **HMC**        | 0.0      | 0.0       | 0.79           | 0.09                      | 0.01        | 0.0         |

| **Angle Max**  | 0.152    | 36.965    | 38.132         | 37.895                    | 0.278       | 0.219       |
| **Two Modes**  | 0.0      | 0.67      | 0.738          | 0.694                     | 0.01        | 0.0         |
| **Random Restarts** | 0.0      | 0.67      | 0.738          | 0.694                     | 0.01        | 0.0         |
| **Random Restarts (filtered)** | 0.0      | 0.67      | 0.738          | 0.694                     | 0.01        | 0.0         |
| **Gibbs**      | 0.0      | 0.67      | 0.738          | 0.694                     | 0.01        | 0.0         |
| **HMC**        | 0.0      | 0.67      | 0.738          | 0.694                     | 0.01        | 0.0         |

| **Angle Max**  | 0.187    | 44.973    | 90.0           | 44.99                     | 1.187       | 16.229      |
| **Infinite Modes** | 0.188    | 44.973    | 90.0           | 44.99                     | 1.185       | 16.199      |
| **Random Restarts** | 0.187    | 44.973    | 90.0           | 44.99                     | 1.185       | 16.199      |
| **Random Restarts (filtered)** | 0.185    | 44.973    | 90.0           | 44.99                     | 1.185       | 16.199      |
| **Gibbs**      | 0.009    | 1.0       | 2.0            | 1.004                     | 0.032       | 0.383       |
| **HMC**        | 0.005    | 1.0       | 2.0            | 1.004                     | 0.032       | 0.383       |

130
<table>
<thead>
<tr>
<th></th>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle</td>
<td>0.069</td>
<td>0.069</td>
<td>0.873</td>
<td>0.615</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>Mean</td>
<td>0.07</td>
<td>0.07</td>
<td>0.75</td>
<td>0.61</td>
<td>0.0</td>
<td>0.09</td>
</tr>
<tr>
<td>One Mode</td>
<td>0.07</td>
<td>0.07</td>
<td>0.91</td>
<td>0.6175</td>
<td>0.0</td>
<td>0.09</td>
</tr>
<tr>
<td>MM $\ell_1$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.019</td>
<td>0.01</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0</td>
<td>0.0</td>
<td>0.03</td>
<td>0.03</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.03</td>
<td>0.03</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Angle</td>
<td>0.06</td>
<td>18.467</td>
<td>18.61</td>
<td>18.563</td>
<td>0.102</td>
<td>0.089</td>
</tr>
<tr>
<td>Mean</td>
<td>0.06</td>
<td>18.475</td>
<td>18.615</td>
<td>18.565</td>
<td>0.1</td>
<td>0.09</td>
</tr>
<tr>
<td>Two Modes</td>
<td>0.0</td>
<td>0.33</td>
<td>0.34</td>
<td>0.34</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MM $\ell_1$</td>
<td>0.0</td>
<td>0.35</td>
<td>0.54</td>
<td>0.54</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0</td>
<td>0.35</td>
<td>0.54</td>
<td>0.54</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.35</td>
<td>0.54</td>
<td>0.54</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Angle</td>
<td>0.078</td>
<td>16.572</td>
<td>27.261</td>
<td>12.835</td>
<td>0.371</td>
<td>5.16</td>
</tr>
<tr>
<td>Mean</td>
<td>0.08</td>
<td>16.475</td>
<td>27.265</td>
<td>12.545</td>
<td>0.371</td>
<td>5.16</td>
</tr>
<tr>
<td>Infinite Modes</td>
<td>0.0</td>
<td>0.373</td>
<td>0.566</td>
<td>0.284</td>
<td>0.011</td>
<td>0.123</td>
</tr>
<tr>
<td>MM $\ell_1$</td>
<td>0.0</td>
<td>0.37</td>
<td>0.56</td>
<td>0.28</td>
<td>0.0</td>
<td>0.0425</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0</td>
<td>0.37</td>
<td>0.56</td>
<td>0.28</td>
<td>0.0</td>
<td>0.0425</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.3775</td>
<td>0.57</td>
<td>0.2975</td>
<td>0.0</td>
<td>0.1475</td>
</tr>
</tbody>
</table>
### Comparison of Metrics and Algorithms (Large Dataset)

**Table A.5: Comparison of Performance of Sampling Method: Likelihood**

<table>
<thead>
<tr>
<th></th>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One Mode</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.22e+04</td>
<td>-1.22e+04</td>
<td>-1.68e+06</td>
<td>-1.00e+06</td>
<td>-1.25e+04</td>
<td>-1.25e+04</td>
</tr>
<tr>
<td></td>
<td>-1.23e+04,</td>
<td>-1.23e+04,</td>
<td>-1.82e+06</td>
<td>-1.04e+06</td>
<td>-1.26e+04,</td>
<td>-1.26e+04,</td>
</tr>
<tr>
<td></td>
<td>-1.22e+04,</td>
<td>-1.22e+04,</td>
<td>-1.69e+06</td>
<td>-9.99e+05</td>
<td>-1.24e+04,</td>
<td>-1.24e+04,</td>
</tr>
<tr>
<td></td>
<td>-1.22e+04</td>
<td>-1.22e+04</td>
<td>-1.60e+06</td>
<td>-9.57e+05</td>
<td>-1.24e+04</td>
<td>-1.24e+04</td>
</tr>
<tr>
<td><strong>Two Modes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.22e+04</td>
<td>-1.22e+04</td>
<td>-3.60e+06</td>
<td>-1.09e+06</td>
<td>-1.25e+04</td>
<td>-1.25e+04</td>
</tr>
<tr>
<td></td>
<td>-1.23e+04,</td>
<td>-1.23e+04,</td>
<td>-1.74e+06</td>
<td>-1.14e+06</td>
<td>-1.26e+04,</td>
<td>-1.26e+04,</td>
</tr>
<tr>
<td></td>
<td>-1.22e+04,</td>
<td>-1.22e+04,</td>
<td>-1.59e+06</td>
<td>-1.10e+06</td>
<td>-1.24e+04,</td>
<td>-1.24e+04,</td>
</tr>
<tr>
<td></td>
<td>-1.22e+04</td>
<td>-1.22e+04</td>
<td>-1.60e+06</td>
<td>-1.04e+06</td>
<td>-1.24e+04</td>
<td>-1.24e+04</td>
</tr>
<tr>
<td><strong>Infinite Modes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.20e+04</td>
<td>-1.20e+04</td>
<td>-1.82e+06</td>
<td>-9.98e+05</td>
<td>-1.25e+04</td>
<td>-1.25e+04</td>
</tr>
<tr>
<td></td>
<td>-1.20e+04,</td>
<td>-1.20e+04,</td>
<td>-1.84e+06</td>
<td>-1.01e+06</td>
<td>-1.26e+04,</td>
<td>-1.26e+04,</td>
</tr>
<tr>
<td></td>
<td>-1.20e+04,</td>
<td>-1.20e+04,</td>
<td>-1.84e+06</td>
<td>-9.99e+05</td>
<td>-1.24e+04,</td>
<td>-1.24e+04,</td>
</tr>
<tr>
<td></td>
<td>-1.20e+04</td>
<td>-1.20e+04</td>
<td>-1.69e+06</td>
<td>-9.78e+05</td>
<td>-1.24e+04</td>
<td>-1.24e+04</td>
</tr>
</tbody>
</table>
### Table A.6: Comparison of Performance of Sampling Method: IAT

<table>
<thead>
<tr>
<th></th>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One Mode IAT</strong></td>
<td>1.01</td>
<td>1.01</td>
<td>1.011</td>
<td>1.009</td>
<td>645.903</td>
<td>1173.558</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.0025</td>
<td>1.0025</td>
<td>600.07</td>
<td>1076.483</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.075</td>
<td>1.01</td>
<td>694.575</td>
<td>1266.646</td>
</tr>
<tr>
<td><strong>Two Modes IAT</strong></td>
<td>1.01</td>
<td>1.011</td>
<td>1.012</td>
<td>1.082</td>
<td>524.915</td>
<td>1105.867</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.0025</td>
<td>1.0630</td>
<td>414.635</td>
<td>1076.483</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>1.1</td>
<td>530.575</td>
<td>1180.912</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.01</td>
<td>1.01</td>
<td>1.0025</td>
<td>565.51</td>
<td>1180.912</td>
</tr>
<tr>
<td><strong>Infinite Modes IAT</strong></td>
<td>1.01</td>
<td>1.008</td>
<td>1.005</td>
<td>1.014</td>
<td>1067.452</td>
<td>1077.456</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.0</td>
<td>1.005</td>
<td>1.0050</td>
<td>1099.7475</td>
<td>1077.456</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.0</td>
<td>1.005</td>
<td>1.01</td>
<td>1068.9</td>
<td>1076.965</td>
</tr>
<tr>
<td></td>
<td>1.01</td>
<td>1.009</td>
<td>1.01</td>
<td>1.01</td>
<td>1077.79</td>
<td>1086.9875</td>
</tr>
<tr>
<td></td>
<td>One Mode</td>
<td>All Modes</td>
<td>Random Restarts</td>
<td>Random Restarts (filtered)</td>
<td>Gibbs</td>
<td>HMC</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------------</td>
<td>-----------------------------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td><strong>Angle Max</strong></td>
<td>0.01</td>
<td>0.01</td>
<td>25.504</td>
<td>16.326</td>
<td>0.027</td>
<td>0.092</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.04</td>
<td>22.95</td>
<td>14.753</td>
<td>0.0226</td>
<td>0.04</td>
</tr>
<tr>
<td><strong>Random Restarts</strong></td>
<td>0.01</td>
<td>0.04</td>
<td>26.68</td>
<td>18.963</td>
<td>0.05</td>
<td>0.09</td>
</tr>
<tr>
<td><strong>Gibbs</strong></td>
<td>0.01</td>
<td>0.04</td>
<td>28.19</td>
<td>17.797</td>
<td>0.05</td>
<td>0.0975</td>
</tr>
<tr>
<td><strong>HMC</strong></td>
<td>0.01</td>
<td>0.04</td>
<td>28.15</td>
<td>17.797</td>
<td>0.05</td>
<td>0.0975</td>
</tr>
<tr>
<td><strong>MM ℓ_1 Max</strong></td>
<td>0.0</td>
<td>0.0</td>
<td>0.444</td>
<td>0.258</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.4825</td>
<td>0.28</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>Two Modes</strong></td>
<td>0.01</td>
<td>10.779</td>
<td>22.131</td>
<td>5.084</td>
<td>0.02</td>
<td>0.072</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>10.659</td>
<td>22.131</td>
<td>5.084</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>0.01</td>
<td>10.659</td>
<td>25.08</td>
<td>4.64</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>10.659</td>
<td>26.25</td>
<td>4.64</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td><strong>MM ℓ_1</strong></td>
<td>0.0</td>
<td>0.16</td>
<td>0.382</td>
<td>0.075</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.16</td>
<td>0.418</td>
<td>0.09</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>0.0</td>
<td>0.16</td>
<td>0.44</td>
<td>0.09</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.16</td>
<td>0.44</td>
<td>0.09</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>Infinite Modes</strong></td>
<td>0.01</td>
<td>21.412</td>
<td>42.943</td>
<td>35.901</td>
<td>0.202</td>
<td>0.556</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>21.3</td>
<td>42.943</td>
<td>35.901</td>
<td>0.202</td>
<td>0.556</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>0.01</td>
<td>21.411</td>
<td>42.999</td>
<td>35.901</td>
<td>0.202</td>
<td>0.556</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>21.411</td>
<td>43.86</td>
<td>35.901</td>
<td>0.202</td>
<td>0.556</td>
</tr>
<tr>
<td><strong>MM ℓ_1</strong></td>
<td>0.0</td>
<td>0.348</td>
<td>0.985</td>
<td>0.796</td>
<td>0.0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.348</td>
<td>0.985</td>
<td>0.796</td>
<td>0.0</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>0.0</td>
<td>0.348</td>
<td>1.0375</td>
<td>0.8175</td>
<td>0.0</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>One Mode</th>
<th>All Modes</th>
<th>Random Restarts</th>
<th>Random Restarts (filtered)</th>
<th>Gibbs</th>
<th>HMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle</td>
<td>0.01</td>
<td>0.01</td>
<td>5.172</td>
<td>4.293</td>
<td>0.02</td>
<td>0.041</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One Mode MM ℓ₁</td>
<td>0.01</td>
<td>0.01</td>
<td>0.077</td>
<td>0.062</td>
<td>0.0</td>
<td>0.033</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two Modes Angle</td>
<td>0.01</td>
<td>5.38</td>
<td>3.467</td>
<td>2.457</td>
<td>0.02</td>
<td>0.033</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infinite Modes</td>
<td>0.01</td>
<td>7.935</td>
<td>16.088</td>
<td>11.978</td>
<td>1.108</td>
<td>0.307</td>
</tr>
<tr>
<td>Angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infinite Modes</td>
<td>0.01</td>
<td>0.129</td>
<td>0.258</td>
<td>0.186</td>
<td>0.0</td>
<td>0.001</td>
</tr>
<tr>
<td>MM ℓ₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Persistent Minimum Covering Number Plots (Small Dataset)

(a) Similarity measure: Max Angle

Figure A.1: Unique Mode

(b) Similarity measure: MinMatch ($\ell_1$)

(a) Similarity measure: Max Angle

Figure A.2: Two Modes
Figure A.3: Infinite modes

(a) Similarity measure: Max Angle

(b) Similarity measure: MinMatch ($\ell_1$)
Persistent Minimum Covering Number Plots (Large Dataset)

Figure A.4: Unique Mode

(a) Similarity measure: Max Angle
(b) Similarity measure: MinMatch ($\ell_1$)

Figure A.5: Two Modes

(a) Similarity measure: Max Angle
(b) Similarity measure: MinMatch ($\ell_1$)
Figure A.6: Infinite modes
This appendix provides greater insight into the performance of the Stein Variational Gradient Descent (SVGD) algorithm. Additionally, it motivates the usage of weights in particle based inference as it leads to greater expressivity in the approximate posterior (for a finite number of particles). The findings presented here have partly motivated the work presented in Chapter 5.

Local gradients are not enough

In order to demonstrate the challenges of escaping local modes in the baseline Stein Variational Gradient Descent (SVGD) algorithm, we consider a simple 2-D Gaussian Mixture. Upon initializing the SVGD algorithm in one of the modes, we observed that the ability of SVGD to move across modes is dependent on the number of particles. Figure B.2 shows for particles initialized at the central mode (at (0,0)), we need 150 particles to be able to escape out into the remaining modes. Figure B.3 shows how the largest distance between two points in the approximating set grows with number of particles $N_q$ in SVGD. As more particles are being packed into a single mode, we see a growing repulsion between the particles. The vector field drawn in Figure B.2 is the score function, and it shows that the outermost particles of the 100 particle approximation are still within the basin of attraction of the central mode. By increasing the precision of this central mode from 15$I$ to 20$I$,
we notice that it becomes more difficult to escape it – requiring 300 particles (figure B.1).

This pattern is driven by two competing forces. The sharper changes in likelihood of the distribution give stronger gradient information to stay within that mode. A larger number of particles provides a greater repulsive force (encouraging the diversity component of the Stein discrepancy).

Further, Sminchisescu et al. (2007) notes that in high dimensional spaces, we expect there to be many ridges of probability as there are likely to be some directions in which the posterior density decays sharply. This would make it increasingly challenging for SVGD to transform particles to move across modes.

This motivates the need for a framework that can incorporate particle streams, where the generated particles may come from algorithms that can find multiple modes in addition to performing local exploration of connected regions of high posterior density.

**Figure B.1: SVGD escape**

Similar to the 1d case, Figure B.3 shows that the maximum distance between particles within the single central mode increases as the number of particles increases.

**Weighted collections are more expressive**

In finite-particle approaches, information about the importance of different regions can be inferred by the density of particles. In our aim to minimize the number of particles needed to approximate posteriors, we leverage weighted particle collections which allow us to obtain equal or lower Stein Discrepancies for fewer particles.
STEIN DISCREPANCY FOR A WEIGHTED COLLECTION

Recall that the Stein discrepancy is the expectation of the kernel $K_p$ over the joint distribution of two identically distributed variables. For a discrete distribution with probabilities $p, q$, we can specify this joint distribution in terms of the outer product $P_J = p_qp_q^T$. Note that $P_J[i, j] = p_q[i]q[i]q[j]$ as required. Since the kernel $K_p$ is symmetric, we only need to evaluate it for $\frac{N_q^2 + N_q}{2}$ pairs (where $N_q$ is the number of particles in $q$). Once the relevant kernel values are calculated, the weights on the particles can be considered parameters to optimize in order to minimize Stein discrepancy with a fixed collection.

EMPirical verification

We demonstrate the expressivity of weighted collections in a 2-D Gaussian Mixture where we take a collection of 200 particles obtained from SVGD and show how lower Stein discrepancies can be obtained with weighted subsets of this collection.

Figure B.2: A collection of 100 particles is unable to escape the central mode. A large collection has a higher repulsive force and is able to escape into the two other modes.

Figure B.3: An empirical verification of the repulsive force for particles localized in a single mode. We see an increase in the maximum distance between SVGD particles is observed as the total size of the approximating set increases.
We take the SVGD collection and find optimum weights corresponding to it. We remove particles in order of increasing weight (lowest first). Once these particles are removed, we compute the Stein discrepancy for renormalized weights (labelled Cut in Figure B.4), and also by re-optimizing the weights (labelled Cut and Opt in Figure B.4) for the smaller collection. Figure B.4 shows that by simply cutting and re-normalizing weights, we can do better than the 200 particle SVGD approximation by picking a 140 particle subset of it. For the same subsets, if we re-optimize the weights, we can do equally well in terms of Stein discrepancy with a weighted subset that has even fewer particles (around 55).

**Figure B.4:** Using weights on subsets of a 200 particle selection allows us to use fewer particles to achieve smaller Stein discrepancy
This appendix contains supplementary figures to accompany the work presented in Chapter 5. We show the performance of our algorithm as well as baselines on all our datasets across a range of quality and diversity metrics along with runtime information.

**Quality of factorizations**

We measure the quality of factorizations in terms of the log of the joint likelihood (figures C.4 and C.1) as well the Frobenius NMF objective (figures C.5 and C.2). We see that both quality measures are in agreement with each other. We also assess the quality of factorizations via the reconstruction error of heldout data (figures C.6 and C.3).

Overall, Q-Transform, Random, NNDSVDAr, HMC and (for the exponential-Gaussian model) Gibbs produce high quality factorizations. The initialization-based approaches all work well as they use specialized NMF algorithms that are designed to find high quality factorizations. HMC was given a warm start with a high likelihood initialization and the chain continues to stay in high likelihood regions. The remaining gradient-based approaches for optimizing a collection of particles (SVGD and DSGD) fail to produce high quality factorizations. This is indicative of the need for specialized NMF algorithms designed to work with the constraints and structure of the NMF problem.
and highlight how difficult it is to apply a naive gradient descent approach for finding NMFs.

Gaussian Exponential BNMF Log of Joint Likelihood

Figure C.1: The joint likelihood of factorizations shows that SVGD and DSGD generally produce the worst quality factorizations. The remaining algorithms produce higher quality factorizations.
Figure C.2: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that meet the threshold quality (black line) for useful factorizations. The other approaches consistently produce factorizations that meet this minimum quality requirement.
Figure C.3: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that result in low error on heldout data. The other approaches consistently produce factorizations which generalize better and give low error on heldout data.
Figure C.4: The joint likelihood of factorizations shows that SVGD and DSGD generally produce the worst quality factorizations. HMC, NNDSVDar, Random and $Q^2$-Transform produce high quality factorizations.
Figure C.5: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that meet the threshold quality (black line) for useful factorizations. The other approaches consistently produce factorizations that meet this minimum quality requirement.
Figure C.6: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that result in low error on heldout data. The other approaches consistently produce factorizations which generalize better and give low error on heldout data. The matrix completion error on the missing data variant of the Olivetti faces dataset also gives similar results.
Diversity of factorizations

Similarity of factorizations is measured by the kernel (equation 5.7) for the base RKHS (figures C.10 and C.7) used in evaluating the Stein discrepancy and by pairwise distances between basis matrices (figures C.11 and C.8) and weights matrices (figures C.12 and C.9).

Generally, the HMC chain exhibits the least exploration of the factorization space. Remaining algorithms, particularly $Q$-Transform exhibits higher amounts of diversity in the factorization space. The diversity metrics indicate that SVGD and DSGD give diverse factorizations but the quality metrics indicate that these factorizations are poor quality (do not correspond to high likelihood regions of the posterior) therefore such diversity is of little interest.

Exponential-Gaussian NMF Kernel similarity

Figure C.7: The kernel similarity indicates that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. $Q$-Transform and Random are the only algorithms that produce factorizations of high quality that are not similar.
Figure C.8: The pairwise distance between basis matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSV Dar, $Q$-SVGD, DSGD and SVGD factorizations are also very similar. $Q$-Transform and Random produce basis matrices that are more distinct than the Gibbs sampler.
Exponential-Gaussian BNMF Frobenius distance between weights matrices

Figure C.9: The pairwise distance between weights matrices shows that factorization collections obtained by DSGD, SVGD and Q-SVGD lead to diverse weights matrices. The remaining methods lead to similar levels of diversity within the factorization collections.
Figure C.10: The kernel similarity indicates that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVdar factorizations are also very similar.
Figure C.11: The pairwise distance between basis matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. $Q$-Transform and Random are the only algorithms that produce factorizations of high quality that are different.
Figure C.12: The pairwise distance between weights matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSV-Dar factorizations are also very similar. $Q$-Transform and Random are the only algorithms that produce factorizations of high quality that are different.
Baseline Performance on Prediction Tasks

We show the performance of other baseline algorithms on the prediction tasks for the 20-Newsgroups and Autism datasets (see figures 5.6 and 5.8 in main text for reference)

**Autism Prediction Task: Performance of baseline algorithms**

![Autism Prediction Task using SVGD](image1)

![Autism Prediction Task using SVGD-Q](image2)

![Autism Prediction Task using HMC](image3)

![Autism Prediction Task using NNDSVDar](image4)

![Autism Prediction Task using Gibbs Gauss-Ex](image5)

![Autism Prediction Task using RANDOM](image6)

*Figure C.13:* Variability in the prediction task on the Autism dataset shows that SVGD factorizations yield poor prediction (that are slightly improved upon by $Q$-SVGD), HMC and NNDSVDar factorizations predictions are high but not diverse, and Gibbs and random restarts yields performance similar to $Q$-Transform
Figure C.14: Variability in the prediction task on the 20 Newsgroups dataset shows that SVGD factorizations yield poor prediction (that is drastically improved upon by $Q$-SVGD), HMC factorizations are not diverse, NNDSVDar, Gibbs and random restarts yields some variability in prediction that is similar to $Q$-Transform
**Discrete posteriors for $M = \{25, 50\}$**

We provide results on the quality of discrete BNMF posteriors for the SILF and exponential-Gaussian models for $M = 25$ and $M = 50$.

**Exponential-Gaussian BNMF Posterior Quality with $M = 25$**

![Figure C.15](image)

*Figure C.15*: For each dataset we show the quality of the BNMF approximate posterior ($M = 25$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).
Exponential-Gaussian NMF Posterior Quality with $M = 50$

**Figure C.16:** For each dataset we show the quality of the BNMF approximate posterior ($M = 50$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).
SILF BNMF Posterior Quality with $M = 25$

Figure C.17: For each dataset we show the quality of the BNMF approximate posterior ($M = 25$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).
Figure C.18: For each dataset we show the quality of the BNMF approximate posterior ($M = 50$) and the corresponding runtime of $Q$-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the $Q$-Transform initializations (in red).
Runtime Calculation

Here we provide full details on how our runtimes were calculated. Recall that both our transfer-based algorithm and the baselines all can be considered as having two phases: (1) Producing candidate particles $\{\theta_m\}_{m=1}^{M}$, and (2) Calculating the optimal weights $\{w_m\}_{m=1}^{M}$ to minimize the Stein discrepancy. Importantly, to give all the methods their best performance, we calculate optimal weights for all the methods, including those such as MCMC that produce unweighted collections.

Runtime for generating candidate particles  
Thus, all the methods only differ in how the candidate particles $\{\theta_m\}_{m=1}^{M}$ are found. To focus on the parts that differ, in our figures, we only report on the computational time required to produce these candidate particles. Below we describe the computations for obtaining the candidate particles:

- MCMC Methods: We keep track of the time from the initialization of the chain through each element added to the chain. At various points, we thin the current chain to the desired number of particles and record the time elapsed since initialization.

- Gradient-based Optimization: We keep track of the time taken from initialization through all gradient updates. At various points, we output the current particles and record the time elapsed since initialization.

- Initialization Approaches: For each factorization $\theta_m$, we keep track of the time taken from initialization through optimization using NMF solvers. In the case of $Q$-Transform, we also include the time taken to find the transferrable $Q$ matrices (less than 0.5 seconds for all 100 pairs of $Q$-Transform matrices!), even though that cost is shared by all the datasets because we re-use the same matrices. The reported runtime is the sum of the time taken for the initialization procedure and for running the NMF solver.

Runtime for optimizing weights  
We do not report the time required to weight the particles because it is the same for all methods. In figure C.19 we see the time to optimize the weights depends only on the number of particles and not their parameter values or quality.

Note for efficient batch weight optimization  
To produce the confidence intervals in our experiments, we had to run our approaches many times. To speed up these repetitions, we observed that the weight optimization step first requires computing the pairwise kernel matrix...
$K_{ij} = \mathcal{K}_p(\theta_i, \theta_j)$ (from equation 5.3) and given $\mathbf{K}$, running a solver to find the weights $\{w_m\}_{m=1}^M$.

When computing the weights for many subsets of a collection of factorizations (as is the case with the initialization based approaches), we can avoid recomputing terms in the kernel matrix multiple times by calculating the pairwise kernel matrix $\mathbf{K}_{\text{max}}$ for a large collection of $M_{\text{max}}$ factorizations. Subsequently, the matrix $\mathbf{K}_M$ for a given subset of size $M$ (from this large collection) can be determined by simply choosing the relevant columns and rows of $\mathbf{K}_{\text{max}}$. We use this approach to compute quickly compute weights for any $M$-sized collection sampled from a larger set of $M_{\text{max}}$ factorizations. Note that this efficiency in experimental design trick does not affect our reported numbers, which include only the time to generate the candidate particles, but may help others achieve faster computational times if they are replicating our results.

**Figure C.19:** The runtime for calculating optimal weights (left: low-quality random parameters, right: “good” parameters obtained from NMF solvers) does not depend on the parameters; both have the same runtime costs and depend only on collection size.
References

20NG. The 20 newsgroups text dataset scikit-learn 0.19.1 documentation.  


David Aldous and Jim Fill. Reversible markov chains and random walks on graphs.


165


José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse


Theodore S Motzkin, Howard Raiffa, Gerald L Thompson, and Robert M Thrall. The double description method. 1953.


