UNIVERSITY OF CALIFORNIA SAN DIEGO

On the Scalable Construction of Measure Transport Maps and Applications in Health Analytics

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Bioengineering

by

Marcela Patricia Mendoza Martinez

Committee in charge:

Professor Todd P. Coleman, Chair
Professor Gert Cauwenberghs
Professor Tara Javidi
Professor Lawrence K. Saul
Professor Terrence J. Sejnowski

2018

The Dissertation of Marcela Patricia Mendoza Martinez is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2018

DEDICATION

To all the students and workers who are and will be in the fight for social justice in this university; there is still a long way to go. May we never give up hope.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Todd P. Coleman for his support as the chair of my committee. Professor Coleman has guided me tremendously in developing my skills in quantitative analysis.

I would also like to acknowledge Professor Terrence J. Sejnowski and his research group for their support during my first years as a graduate student. The Sejnowski lab helped shape a lot of my research and inspired me as I saw the possibilities of perseverance in scientific research.

I would like to acknowledge the Coleman lab members as they each helped advice me in career and life. In particular, I would like to acknowledge Gabriel Schamberg for the many mathematical discussions that led to the completion of my work. Joanne Shin for discussions on machine learning and her help in coding. Armen Gharibans and Dae Kang for much valuable career advice. Mridu Sinha, Sheila Rosenberg, Mike Bajema, and Marianne Catanho for their friendship, mentorship, and support. Finally, the co-authors of my publications Justin Tantiongloc, Diego Mesa, Alexis Allegra, and Sanggyun Kim. Their team spirit made the completion of my dissertation work into a fun experience.

My friends who helped get me get through these years: Valentin, Elaine, Dimitar, Sayali, Sohini, Vineet, Niki, among others. You were my grad school family.

I want to acknowledge Alan De Anda who has been my rock and support for twelve years and counting. I hope there's many more adventures by your side.

Finally, I want to acknowledge my immigrant family who has made so many sacrifices to allow me the opportunities I now have. My father Valente, my mother Patricia, my brothers Hector and Luis, my sister-in-law Rocio, and my nephew Diego.

Chapter 1, in full, has been submitted for publication of the material as it may appear in Neural Computation 2018. Mesa, Diego A; Tantiongloc, Justin; Mendoza, Marcela, Coleman, Todd. The dissertation author was the co-primary investigator and author of this paper along with Justin Tantiongloc and Diego Mesa.

Chapter 2, in full, has been submitted for publication of the material as it may appear in

Bayesian Analysis 2018. Mendoza, Marcela; Allegra, Alexis; Coleman, Todd. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is currently being prepared for submission for publication of the material. Mendoza, Marcela; Coleman, Todd P. The dissertation author was the primary investigator and author of this material.

VITA

2008-2009	Undergraduate Research Assistant, University of Texas at El Paso
2009-2011	Undergraduate Research Assistant, University of Texas at Austin
2012	B.S. in Biomedical Engineering, University of Texas at Austin
2016	Research Intern, IBM Research- Almaden
2017	M.A. in Applied Mathematics, University of California San Diego
2018	Ph.D. in Bioengineering, University of California San Diego

PUBLICATIONS

M. Mendoza, D. Mesa, J. Tantiongloc, T. P. Coleman, "A Distributed Framework for the Construction of Transport Maps", (in review at Neural Computation)

M. Mendoza, A. Allegra, T. P. Coleman, "Bayesian Lasso Posterior Sampling via Parallelized Measure Transport", (submitted to Bayesian Analysis)

A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, D. Modha, "A Low Power, Fully Event-Based Gesture Recognition System"; The IEEE Conference on Computer Vision and Patter Recognition (CVPR), 2017

T. P. Coleman,J. Tantiongloc, A. Allegra, D. Mesa, D. Kang, M. Mendoza, "Diffeomorphism Learning via Relative Entropy Constrained Optimal Transport", IEEE Global Conference on Signal and Image Processing, 2016.

M. Mendoza, S. Kim, and T. P. Coleman, "Bayesian LASSO in a Distributed Architecture", IEEE Global Conference on Signal and Image Processing, December 2015.

ABSTRACT OF THE DISSERTATION

On the Scalable Construction of Measure Transport Maps and Applications in Health Analytics

by

Marcela Patricia Mendoza Martinez

Doctor of Philosophy in Bioengineering

University of California San Diego, 2018

Professor Todd P. Coleman, Chair

Characterizing and sampling from probability distributions is useful to reason about uncertainty in large, complex, and multi-modal datasets. One established and increasingly popular method to do so involves finding transformations or transport maps between one distribution to another. The computation of these transport maps is the subject of the field of Optimal Transportation, a rich area of mathematical theory that has led to many applications in machine learning, economics, and statistics. Finding these transport maps, however, usually comprises computational difficulties, particularly when datasets are large both in dimension and the number of samples to learn from.

Building upon previous work in our group, we introduce a formulation to find transport

maps that is parallelizable and solvable with convex optimization methods. We show applications in the field of health analytics encompassing scalable Bayesian inference, density estimation, and generative models. We show how this formulation is scalable with the dimension of data and can be parallelized utilizing a sweep of architectures such as cloud computing services and Graphics Processing Units. Within the context of Bayesian inference, we present a distributed framework for finding the full posterior distribution associated with LASSO problems and show advantages compared to traditional methods of computing this posterior. Finally, we discuss novel methods to reduce the number of parameters necessary to approximate transport maps.

Chapter 1

A Distributed Framework for the Construction of Transport Maps

1.1 Introduction

While scientific problems of interest continue to grow in size and complexity, managing uncertainty is increasingly paramount. As a result, the development and use of theoretical and numerical methods to reason in the face of uncertainty, in a manner that can accommodate large datasets, has been the focus of sustained research efforts in statistics, machine learning, information theory and computer science. The ability to construct a mapping which transforms samples from one distribution P to another distribution Q enables the solution to many problems in machine learning.

One such problem is Bayesian inference, [17, 8, 51], where a latent signal of interest is observed through noisy observations. Fully characterizing the posterior distribution is in general notoriously challenging, due to the need to calculate the normalization constant pertaining to the posterior density. Traditionally, point estimation procedures are used, which obviate the need for this calculation, despite their inability to quantify uncertainty. Generating samples from the posterior distribution enables approximation of any conditional expectation, but this is typically performed with Markov Chain Monte Carlo (MCMC) methods [22, 1, 26, 18, 37] despite the following drawbacks: (a) the convergence rates and mixing times of the Markov chain are generally unknown, thus leading to practical shortcomings like "sample burn in" periods;

and (b) the samples generated are necessarily correlated, lowering effective sample sizes and propagating errors throughout estimates [47]. If we let P be the prior distribution and Q the posterior distribution for Bayesian inference , then an algorithm which can transform independent samples from P to Q, without knowledge of the normalization constant in the density of Q, enables calculation of any conditional expectation with fast convergence.

As another example, generative modeling problems entail observing a large dataset with samples from an unknown distribution P (in high dimensions) and attempting to learn a representation or model so that new independent samples from P can be generated. Emerging approaches to generative modeling rely on the use of deep neural networks and include variational autoencoders [32], generative adversarial networks [23] and their derivatives [35], and auto-regressive neural networks [33]. These models have led to impressive results in a number of applications, but their tractability and theory are still not fully developed. If P can be transformed into a known and well-structured distribution Q (e.g. a multivariate standard Gaussian), then the inverse of the transformation can be used to transform new independent samples from Q into new samples from P.

While these issues relate to the functional attractiveness of the ability to characterize and sample from non-trivial distributions, there is also the issue of computational efficiency. There continues to be an ongoing upward trend of the availability of distributed and hardware-accelerated computational resources. As such, it would be especially valuable to develop solutions to these problems that are not only satisfactory in a functional sense, but are also capable of taking advantage of the ever-increasing scalability of parallelized computational capability.

1.1.1 Main Contribution

The main contribution of this work is to extend our previous results on finding transport maps to provide a more general transport-based push-forward theorem for pushing independent samples from a distribution P to independent samples from a distribution Q. Moreover, we show

how given only independent samples from P, knowledge of Q up to a normalization constant, and under the traditionally mild assumption of the log-concavity of Q, it can be carried out in a *distributed* and *scalable* manner, leveraging the technique of alternating direction method of multipliers (ADMM) [12]. We also leverage variational principles from nonequilibrium thermodynamics [28] to represent a transport map as an aggregate composition of simpler maps, each of which minimizes a relative entropy along with a transport-cost-based regularization term. Each map can be constructed with a complementary, ADMM-based formulation, resulting in the construction of a measure transport map smoothly and sequentially with applicability in high-dimensional settings.

Expanding on previous work on the real-world applicability of these general-purpose algorithms, we showcase the implementation of a Bayesian LASSO-based analysis of the Boston Housing dataset [25] and a high-dimensional example of using transport maps for generative modeling for the MNIST handwritten digits dataset [34].

1.1.2 Previous Work

A methodology for finding transport maps based on ideas from optimal transport within the context of Bayesian inference was first proposed in [16] and expanded upon in conjunction with more traditional MCMC-based sampling schemes in [40, 44, 45, 52].

Our previous work used ideas from optimal transport theory to generalize the posterior matching scheme, a mutual-information maximizing scheme for feedback signaling of a message point in arbitrary dimension [39, 38, 54]. Building upon this, we considered a relative entropy minimization formulation, as compared to what was developed in [16], and showed that for the class of log-concave distributions, this is a convex problem [30]. We also previously described a distributed framework [41] that we expand upon here.

In the more traditional optimal transportation literature convex optimization has been used to varying success in specialized cases [42], as well as gradient-based optimization methods [46, 7, 6]. The use of *stochastic* optimization techniques in optimal transport is also of current interest

[20]. In contrast, our work below presents a specific distributed framework where extensions to stochastic updating have been previously developed in a general case. Incorporating them into this framework remains to be explored.

Additionally, of much recent interest is the efficient and robust calculation of Wasserstein *barycenters* (center of mass) across partial empirical distributions calculated over batches of samples/data [15, 14], and has also found application in Bayesian inference [53]. While related, our work focuses instead on calculating the *full* empirical distribution through various efficient parameterizations discussed below.

Building on much of this, there is growing interest in specific applications of these transport problems in various areas [2, 55]. These *derived* transport problems are proving to be a fruitful alternative approach and are the subject of intense research. The framework presented below is general purpose and could benefit many of the derived transport problems.

Excellent introductory and references to the field can be found in [57, 49].

The rest of this paper is organized as follows: in Section 1.2, we provide some necessary definitions and background information; in Section 1.3, we describe the distributed general push-forward framework and provide several details on its construction and use; in Section 1.4, we formulate a specialized version of the objective specifically tailored for sequential composition; in Section 1.5, we discuss applications and examples of our framework; and we provide concluding remarks in Section 1.6.

1.2 Preliminaries

In this section we make some preliminary definitions and provide background information for the rest of this paper.

1.2.1 Definitions and Assumptions

Assume the space for sampling is given by $\mathsf{W} \subset \mathbb{R}^D$, a convex subset of D-dimensional Euclidean space. Define the space of all probability measures on W (endowed with the Borel

sigma-algebra) as $\mathsf{P}(\mathsf{W})$. If $P \in \mathsf{P}(\mathsf{W})$ admits a *density* with respect to the Lebesgue measure, we denote it as p.

Assumption 1. *We assume that $P, Q \in \mathsf{P}(\mathsf{W})$ admit densities p, q with respect to the Lebesgue measure.*

This work is fundamentally concerned with trying to find an appropriate *push-forward* between two probability measures, P and Q:

Definition 1.2.1 (Push-forward)**.** *Given $P, Q \in \mathsf{P}(\mathsf{W})$ we say that a map $S : \mathsf{W} \to \mathsf{W}$ pushes forward P to Q (denoted as $S_{\#}P = Q$) if a random variable X with distribution P results in $Y \triangleq S(X)$ having distribution Q.*

Of interest to us is the class of invertible and "smooth" push-forwards:

Definition 1.2.2 (Diffeomorphism)**.** *A mapping S is a diffeomorphism on W if it is invertible, and both S and S^{-1} are differentiable. Let $\mathscr{D}$ be the space of all diffeomorphisms on W.*

A subclass of these, are those that are "orientation preserving":

Definition 1.2.3 (Monotonic Diffeomorphism)**.** *A mapping $S \in \mathscr{D}$ is orientation preserving, or monotonic, if its Jacobian is positive-definite:*

$$J_S(u) \succeq 0, \quad \forall u \in \mathsf{W}$$

Let $\mathscr{D}_+ \subset \mathscr{D}$ be the set of all monotonic diffeomorphisms on W.

The Jacobian $J_S(u)$ can be thought of as how the map "warps" space to facilitate the desired mapping. Any monotonic diffeomorphism necessarily satisfies the following Jacobian equation:

Lemma 1.2.4 (Monotonic Jacobian Equation)**.** *Let $P, Q \in \mathsf{P}(\mathsf{W})$ and assume they have densities p and q. Any map $S \in \mathscr{D}_+$ for which $S\#P = Q$ satisfies the following Jacobian equation:*

$$p(u) = q(S(u)) \det(J_S(u)) \quad \forall u \in \mathsf{W} \tag{1.1}$$

We will now concern ourselves with two different notions of "distance" between probability measures.

Definition 1.2.5 (KL Divergence)**.** *Let $P, Q \in \mathsf{P}(\mathsf{W})$ and assume they have densities p and q. The Kullback-Leibler (KL) divergence, or relative entropy, between P and Q is given by*

$$D(P\|Q) \quad = \quad \mathbb{E}_P \left[\log \frac{p(X)}{q(X)} \right]$$

The KL divergence is non-negative and is zero if and only if $p(u) = q(u)$ for all u.

Definition 1.2.6 (Wasserstein Distance)**.** *For $P, Q \in \mathsf{P}(\mathsf{W})$ with densities p and q, the Wasserstein distance of order two between P and Q can be described as*

$$d(P,Q)^2 \triangleq \inf \left\{ \mathbb{E}_{P_{X,Y}}[\|X - Y\|^2] : X \sim P, Y \sim Q \right\} \tag{1.2}$$

The following theorem will be useful throughout:

Theorem 1.2.7 ([13, 56])**.** *Under Assumption 4, $d(P,Q)$ can be equivalently expressed as*

$$d(P,Q)^2 \triangleq \inf \left\{ \mathbb{E}_P[\|X - S(X)\|^2] : S_\# P = Q \right\} \tag{1.3}$$

and there is a unique minimizer S^* which satisfies $S^* \in \mathscr{D}_+$.

Note that this implies the following corollary:

Corollary 1.2.8. *For any P, Q satisfying Assumption 4, there exists a $S \in \mathscr{D}_+$ for which $S_\# P = Q$, or equivalently, for which* (3.1) *holds.*

1.3 KL Divergence-based Push-Forward

In this section, we present the distributed push-forward framework that relies on our previously published relative entropy-based formulation of the measure transport problem, and discuss several issues related to its construction.

1.3.1 General Push-Forward

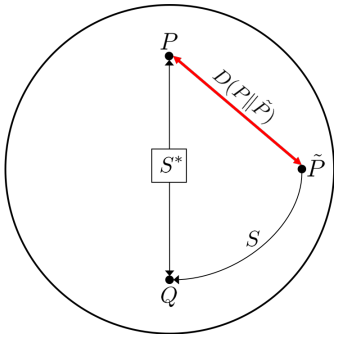


Figure 1.1. General Push-Forward: Probability measures $P, \tilde{P}(\cdot;S)$ and Q are represented as points in $\mathsf{P}(\mathsf{W})$. When Q is assumed to be constant, an arbitrary map $S \in \mathscr{D}_+$ can be thought of as inducing a distribution $\tilde{P}(\cdot;S)$. Thus, S pushes $\tilde{P}(\cdot;S)$ to Q (the solid black line labeled S in the figure).

According to Lemma 3.2.4, a monotonic diffeomorphism pushing P to Q will necessarily satisfy the Jacobian equation (3.1). Note that although we think of a map S as pushing *from P to Q*, we have written (3.1) so that p appears by itself on the left-hand side, while S is being *acted on* by q on the right-hand side. This notation is suggestive of the following interpretation: If we think of the destination density q as an *anchor point*, then for any *arbitrary* mapping $S \in \mathscr{D}_+$, we can describe an *induced* density for $\tilde{p}(u;S)$ according to Eq. (3.1) as:

$$\tilde{p}(u;S) = q(S(u))\det(J_S(u)) \qquad \text{for all } u \in \mathsf{W} \tag{1.4}$$

With this notation, we can interpret $(\tilde{p}(u;S) : S \in \mathscr{D}_+)$ as a parametric family of densities, and for any fixed $S \in \mathscr{D}_+$, $\tilde{p}(u;S)$ is a density which integrates to 1. We note that by construction, any $S \in \mathscr{D}_+$ necessarily pushes $\tilde{P}(\cdot;S)$ to Q: $S_\# \tilde{P}(\cdot;S) = Q$. We can then cast the transport problem

as finding the mapping $S \in \mathscr{D}_+$ that minimizes the relative entropy between P and the induced $\tilde{P}$.

$$S^* = \underset{S \in \mathscr{D}_+}{\arg\min} \; D(P \| \tilde{P}(\cdot; S)) \tag{1.5}$$

This perspective is represented visually in Fig. 1.1.

If we again make another natural assumption:

Assumption 2. *P admits a density p such that:*

$$\mathbb{E}\left[|\log p(X)|\right] < \infty$$

We can expand Eq. (3.2) and combine with (1.4) to write:

$$\begin{aligned}
S^* &= \underset{S \in \mathscr{D}_+}{\arg\min} \; D(P \| \tilde{P}(\cdot; S)) \\
&= \underset{S \in \mathscr{D}_+}{\arg\min} \; \mathbb{E}_P \left[\log \frac{p(X)}{\tilde{p}(X; S)} \right] \\
&= \underset{S \in \mathscr{D}_+}{\arg\min} \; -h(p) - \mathbb{E}_P \left[\log \tilde{p}(X; S) \right] && (1.6) \\
&= \underset{S \in \mathscr{D}_+}{\arg\min} \; -\mathbb{E}_P \left[\log \tilde{p}(X; S) \right] && (1.7) \\
&= \underset{S \in \mathscr{D}_+}{\arg\min} \; -\mathbb{E}_P \left[\log q(S(X)) + \log \det J_S(X) \right] && (1.8)
\end{aligned}$$

where in (3.3), $h(p)$ is the Shannon differential entropy of p, which is fixed with respect to S; (3.4) is by Assumption 2 and Jensen's inequality implying that $|h(p)| < \infty$ and the non-negativity of KL divergence; (3.5) is by combining with (1.4).

We now make another assumption for which we can guarantee efficient methods to solve for (3.2).

Assumption 3. *The density q is log-concave.*

We can now state the main theorem of this section [31, 41]:

Theorem 1.3.1 (General Push-Forward)**.** *Under Assumptions 4-3,*

$$\min_{S\in\mathscr{D}_+} \; D(P\|\tilde{P}(\cdot;S)) \tag{\textbf{GP}}$$

is a convex optimization problem.

Proof. For any $S,\tilde{S}\in\mathscr{D}_+$, we have that $J_S, J_{\tilde{S}} \succeq 0$. For any $\lambda\in[0,1]$ we have that $\tilde{S}_\lambda \triangleq \lambda S + (1-\lambda)\tilde{S}$ and $J_{\tilde{S}_\lambda} = \lambda J_S + (1-\lambda)J_{\tilde{S}} \succeq 0$. Since $\log\det$ is strictly concave over the space of positive definite matrices [11], and by assumption $\log q(\cdot)$ is concave, we have that $-\mathbb{E}_P[\log\tilde{p}(X;S)]$ is a convex function of S on $\mathscr{D}_+$. Existence of $S^*\in\mathscr{D}_+$ for which $D\big(P\|\tilde{P}(\cdot;S^*)\big)=0$ is given by Corollary 1.2.8. □

An important remark on this theorem:

Remark 1. *Theorem 1.3.1 does not place any structural assumptions on P. It need not be log-concave, for example.*

Beginning with Eq. (3.5) above, we see that Problem (**GP**) can then be solved through the use of a Monte-Carlo approximation of the expectation, and we arrive at the following sample-based version of the formulation:

$$S^* = \arg\min_{S\in\mathscr{D}_+} \; \frac{1}{N}\sum_{i=1}^{N}\left[-\log q(S(X_i)) - \log\det(J_S(X_i))\right] \tag{1.9}$$

where $X_i \sim p(X)$.

1.3.2 Consensus Formulation

The stochastic optimization problem in (1.9) takes the general form of:

$$\min_{S} \; \sum_{i=1}^{N} f_i(S)$$

From this perspective, S can be thought of as a *complicating variable*. That is, this optimization problem would be entirely separable across the sum were it not for S. This can be instantiated as a *global consensus* problem:

$$\begin{aligned} \min_{S} \quad & \sum_{i=1}^{N} f_i(S_i) \\ \text{s.t.} \quad & S_i - S = 0 \end{aligned}$$

where the optimization is now separable across the summation, but we must achieve global consensus over S. With this in mind, we can now write a global consensus version of (1.9) as:

$$\begin{aligned} \min_{S_i \in \mathscr{D}_+} \quad & -\frac{1}{N} \sum_{i=1}^{N} \log q(S_i(X_i)) + \log\det(J_{S_i}(X_i)) \\ \text{s.t.} \quad & S_i = S, \quad i =, 1 \ldots, N \end{aligned} \tag{1.10}$$

In this problem, we can think of each (batch of) *sample* as independently inducing some random $\tilde{P}_i$ through a function S_i. The method proposed below can then be thought of as iteratively reducing the distance between each $\tilde{P}_i$ and the true P by reducing the distance between each S_i.

This problem is still over an infinite dimensional space of functions $S \in \mathscr{D}_+$, however.

1.3.3 Transport Map Parameterization

To address the infinite dimensional space of functions mentioned above, as in [41, 30, 31, 40] we parameterize the transport map over a space of multivariate polynomial basis functions formed as the product of D-many univariate polynomials of varying degree. That is, given some $\mathbf{x} = (x_1, \ldots, x_a, \ldots, x_D) \in \mathsf{W} \subset \mathbb{R}^D$, we form a basis function $\phi_{\mathbf{j}}(\mathbf{x})$ of multi-index degree $\mathbf{j} = (j_1, \ldots, j_a, \ldots, j_D) \in \mathscr{J}$ using univariate polynomials ψ_{j_a} of degree j_a as:

$$\phi_{\mathbf{j}}(\mathbf{x}) = \prod_{a=1}^{D} \psi_{j_a}(x_a)$$

This allows us to represent one component of $S \in \mathscr{D}_+$ as a weighted linear combination of basis functions with weights $w_{d,\mathbf{j}}$ as:

$$S^d(\mathbf{x}) = \sum_{\mathbf{j} \in \mathscr{J}} w_{d,\mathbf{j}}\, \phi_{\mathbf{j}}(\mathbf{x})$$

where $\mathscr{J}$ is a set of multi-indices in the representation specifying the order of the polynomials in the associated expansion, and d denotes the d^{th} component of the mapping. In order to make this problem finite-dimensional, we must *truncate* the expansion to some fixed maximum-order O.

$$\mathscr{J} = \left\{ \mathbf{j} \in \mathbb{N}^D : \sum_{i=1}^{D} j_i \leq O \right\}$$

We can now approximate any nonlinear function $S \in \mathscr{D}_+$ as:

$$S(\mathbf{x}) = W\Phi(\mathbf{x})$$

where $K \triangleq |\,\mathscr{J}\,|$ the size of the index-set, $\Phi(\mathbf{x}) = [\phi_{\mathbf{j}_1}(\mathbf{x}), \ldots \phi_{\mathbf{j}_K}(\mathbf{x})]^T$, and $W \in \mathbb{R}^{D \times K}$ is a matrix of weights.

With this, we can now give a finite-dimensional version of (1.10) as:

$$\begin{aligned} \min_{W_i \in \mathbb{R}^{D \times K}} \; & -\frac{1}{N} \sum_{i=1}^{N} \left[\log q(W_i \Phi(X_i)) + \log \det(W_i J_\Phi(X_i)) \right] \\ \text{s.t.} \quad & W_i = W, \quad W_i J_\Phi(X_i) \succeq 0, \quad i = 1, \ldots, N \end{aligned} \tag{1.11}$$

with:

$$\begin{aligned} W_i &= [w_1, \ldots, w_K] && D \times K \\ \Phi(\cdot) &= \left[\phi_{\mathbf{j}_1}(\cdot), \ldots, \phi_{\mathbf{j}_K}(\cdot) \right]^T && K \times 1 \\ J_\Phi(\cdot) &= \left[\frac{\partial \phi_{\mathbf{j}_i}}{\partial x_j}(\cdot) \right]_{i,j} && K \times D \end{aligned}$$

where we have made explicit the implicit constraint that $\det(J_S) \geq 0$ by ensuring that $WJ_\Phi \succeq 0$.

We now provide two important remarks:

Remark 2. *In principle, any basis of polynomials whose finite-dimensional approximations are sufficiently dense over* W *will suffice. In applications where P is assumed known, the basis functions are chosen to be orthogonal with respect to the reference measure P:*

$$\int_{\mathsf{W}} \phi_{\mathbf{j}}(\mathbf{x})\ \phi_{\mathbf{i}}(\mathbf{x})\ p(x)dx = \mathbb{1}_{\mathbf{i}=\mathbf{j}}$$

Within the context of Bayesian inference, for instance, this greatly simplifies computing conditional expectations, corresponding conditional moments, etc. [50].

Remark 3. *When it is important to ensure that the approximation satisfies the properties of a diffeomorphism, we can project* $S(\mathbf{x})$ *onto* $\mathscr{D}_+$ *with solving a quadratic optimization problem, as discussed in Section 1.8.*

We also note that the polynomial representation presented above is chosen to best approximate a transport map, independent of a specific application or representation of the data (Fourier, wavelet, etc.). As mentioned in Remark 7 above, in principle any dense basis will suffice.

1.3.4 Distributed Push-Forward with Consensus ADMM

In this section we will reformulate (3.6) within the framework of the alternating direction method of multipliers (ADMM), and provide our main result, Corollary 1.3.2.

Distributed Algorithm

Using ADMM, we can reformulate (3.6) as a global consensus problem to accommodate a parallelizable implementation. For notational clarity, we write $\Phi_i \triangleq \Phi(X_i)$ and $J_i \triangleq J_\Phi(X_i)$. We then introduce the following auxiliary variables:

$$W_i \triangleq B, \quad B\Phi_i \triangleq p_i, \quad BJ_i \triangleq Z_i$$

We can now write (1.10) as:

$$\begin{aligned}
\min_{\{W,Z,p\}_i,B} \quad & \frac{1}{N}\sum_{i=1}^{N} -\log q(p_i) - \log\det Z_i + \frac{1}{2}\rho\|W_i - B\|_2^2 \\
& + \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\rho\|B\Phi_i - p_i\|_2^2 + \frac{1}{2}\rho\|BJ_i - Z_i\|_2^2 \\
\text{s.t.} \quad & B\Phi_i = p_i : \qquad \gamma_i \quad (D\times 1) \\
& BJ_i = Z_i : \qquad \lambda_i \quad (D\times D) \\
& W_i - B = 0 : \qquad \alpha_i \quad (D\times K) \\
& Z_i \succeq 0
\end{aligned}$$

where in the feasible set, we have denoted the Lagrange multiplier that will be associated with each constraint to the right. We can now raise the constraints to form the fully-penalized Lagrangian as:

$$\begin{aligned}
L_\rho(W,Z,p,B;\gamma,\lambda,\alpha) \;=\; & \frac{1}{N}\sum_{i=1}^{N} -\log q(p_i) - \log\det Z_i \\
+ \; & \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\rho\|W_i - B\|_2^2 + \frac{1}{2}\rho\|B\Phi_i - p_i\|_2^2 \\
+ \; & \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\rho\|BJ_i - Z_i\|_2^2 + \gamma_i^T(p_i - B\Phi_i) \\
+ \; & \frac{1}{N}\sum_{i=1}^{N} \mathbf{tr}\left(\lambda_i^T(Z_i - BJ_i)\right) + \mathbf{tr}\left(\alpha_i^T(W_i - B)\right)
\end{aligned}$$

The key property we leverage from the ADMM framework is the ability to minimize this Lagrangian across each optimization variable *sequentially*, using only the *most recently* updated estimates. After simplification (details can be found in the Appendix), the final ADMM update

equations for the remaining variables are:

$$B^{k+1} = \mathscr{B}_i \cdot \mathscr{B}_s \tag{1.12a}$$

$$W_i^{k+1} = -\frac{1}{\rho}\alpha_i^k + B^{k+1} \tag{1.12b}$$

$$Z_i^{k+1} = Q\tilde{Z}_i Q^T \tag{1.12c}$$

$$\gamma_i^{k+1} = \gamma_i^k + \rho(p_i^{k+1} - B^{k+1}\Phi_i) \tag{1.12d}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(Z_i^{k+1} - B^{k+1}J_i) \tag{1.12e}$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho(W_i^{k+1} - B^{k+1}) \tag{1.12f}$$

$$p_i^{k+1} = \arg\min_{p_i} -\log q(p_i) + \text{pen}(p_i) \tag{1.12g}$$

We look first at the consensus variable B^{k+1}. We can separate its update into two pieces: a static component $\mathscr{B}_s$, and an iterative component $\mathscr{B}_i$:

$$\mathscr{B}_i = \frac{1}{N}\sum_{i=1}^{N}\left[\rho\left(W_i^k + p_i^k\Phi_i^T + Z_i^k J_i^T\right) + \gamma_i^k\Phi_i^T + \lambda_i^k J_i^T + \alpha_i^k\right] \tag{1.13a}$$

$$\mathscr{B}_s = \left[\rho\left(I + \frac{1}{N}\sum_{i=1}^{N}\Phi_i\Phi_i^T + J_iJ_i^T\right)\right]^{-1} \tag{1.13b}$$

The consensus variable can then be thought of as averaging the effect of all other auxiliary variables, and forming the current best estimate for consensus among the distributed computational nodes.

The p-update is the only remaining minimization step that cannot necessarily be solved in closed form, as it completely contains the structure of the q density. In its penalization, all other optimization variables are fixed:

$$\text{pen}(p_i) = \frac{1}{2}\rho\|B^{k+1}\Phi_i - p_i\|_2^2 + \gamma_i^{kT}(p_i - B^{k+1}\Phi_i)$$

The formulation of (1.12) has the following desirable properties:

- Eqs. (1.12a), (1.12b), (3.12c) and (3.12e) to (3.12g) admit closed form solutions. In particular, Eqs. (1.12b) and (3.12e) to (3.12g) are simple arithmetic updates;

- Eq. (1.12g) is a penalized d-dimensional-vector convex optimization problem that entirely captures the structure of Q. In particular, any changes to the problem specifying a different structure of Q will be entirely confined in this update; furthermore, algorithm designers can utilize any optimization procedure/library of their choosing to perform this update.

With this, we can now give an efficient, distributed version of the general push-forward theorem:

Corollary 1.3.2 (Distributed Push-Forward)**.** *Under Assumption 4 and Assumption 3,*

$$\begin{aligned} \min_{W_i \in \mathbb{R}^{d\times K}} \quad & -\frac{1}{N}\sum_{i=1}^{N} \log q(W_i\Phi_i) + \log\det(W_i J_i) \\ s.t. \quad & W_i = W,\ WJ_i \succeq 0 \quad i = 1,\dots,N \end{aligned} \tag{1.14}$$

is a convex optimization problem.

Remark 4. *ADMM convergence's properties are robust to inaccuracies in the initial stages of the iterative solving process [12]. Additionally several key concentration results provide very strong bounds for averages of random samples from log-concave distributions, showing that the approximation is indeed robust [9, Thrm 1.1, 1.2].*

The above framework, under natural assumptions, facilitates the efficient, distributed and scalable calculation of an optimal map that pushes forward some P to some Q.

1.3.5 Structure of the Transport Map

An important consideration in ensuring the construction of transport maps is efficient is their underlying *structure*. In Section 1.3.3 we described a parameterization of the transport map through the multi-index set $\mathscr{J}$ - the indices of polynomial orders involved in the expansion.

However, this parameterization tends to be unfeasible to use in high dimension or with high order polynomials due to the exponential rate at which the number of polynomials increases with respect to these two properties.

In [40], two less expressive, but more computationally feasible map structures that can be used to generate the transport map were discussed, which we briefly reproduce here, along with some useful properties. For more specific details and examples of multi-index sets pertaining to each mode for implementation purposes, see Section 1.8

The first alternative to the map pertaining to the fully-expressive mapping is the Knothe-Rosenblatt map [10], which our group also previously used within the context of generating transport maps for optimal message point feedback communication [38]. Here, each component of the output, S^d, is only a function of the first d components of the input, resulting in a mapping that is *lower-triangular*. Both the Knothe-Rosenblatt and dense mapping described above perform the transport from one density to another, but with *different* geometric transformations. An example of these differences can be found in Figures 3 and 4 of [38].

A Knothe-Rosenblatt arrangement gives the following multi-index set (note that the index-set is now sub-scripted according to dimension of the data denoting the dependence on data component):

$$\mathscr{J}_d^{KR} = \left\{ \mathbf{j} \in \mathbb{N}^D : \sum_{i=1}^{D} j_i \leq O \wedge j_i = 0, \forall i > d \right\}, d = 1, \ldots, D$$

An especially useful property of this parameterization is the following identity for the Jacobian of the map:

$$\log\det(J_S(X_i)) = \sum_{d=1}^{D} \log \partial_d S^d(X_i) \tag{1.15}$$

where $\partial_d S^d(X_i)$ represents the partial derivative of the d^{th} component of the mapping

with respect to the d^{th} component of the data, evaluated at X_i.

Furthermore, the positive-definiteness of the Jacobian can equivalently be enforced for a lower-triangular mapping by ensuring the following:

$$\partial_d S^d > 0, \quad 1 \leq d \leq D \tag{1.16}$$

We can then write a Knothe-Rosenblatt special-case version of Eq. (1.10) as:

$$\begin{aligned} &\min_{S_i \in \mathscr{D}_+^{KR}} -\frac{1}{N}\sum_{i=1}^{N} \log q(S_i(X_i)) + \sum_{d=1}^{D} \log \partial_d S_i^d(X_i) \\ &\text{s.t.} \quad S_i = S, \quad i = 1,\ldots,N \end{aligned} \tag{1.17}$$

Indeed, we use this to our advantage in Section 1.4.

Finally, in the event that the Knothe-Rosenblatt mapping also proves to have too high of model complexity, an even less expressive mapping is a Knothe-Rosenblatt mapping that ignores all multivariate polynomials that involve more than one data component of the input at a time, resulting in the following multi-index set:

$$\mathscr{J}_d^{KRSV} = \left\{ \mathbf{j} \in \mathbb{N}^D : \sum_{i=1}^{D} j_i \leq O \wedge j_i j_l = 0, \forall i \neq l \wedge j_i = 0, \forall i > d \right\}, \quad d = 1,\ldots,D$$

Although less expressive and less precise than the total order Knothe-Rosenblatt map, these maps can often still perform at an acceptable level of accuracy with respect to many problems.

1.3.6 Algorithm for Inverse Mapping with Knothe-Rosenblatt Transport

It may be desirable to compute the inverse mapping of a given sample from Q, that is, $S^{-1}(X), X \sim Q$. When the forward mapping S is constrained to have Knothe-Rosenblatt structure, and a polynomial basis is used to parameterize the mapping, the process of inverting a sample from Q reduces to solving a sequential series of polynomial root-finding problems [40]. We give a more detailed implementation-based explanation of this process alongside a discussion of implementation details for the Knothe-Rosenblatt maps in Section 1.8.

1.4 Sequential Composition of Optimal Transportation Maps

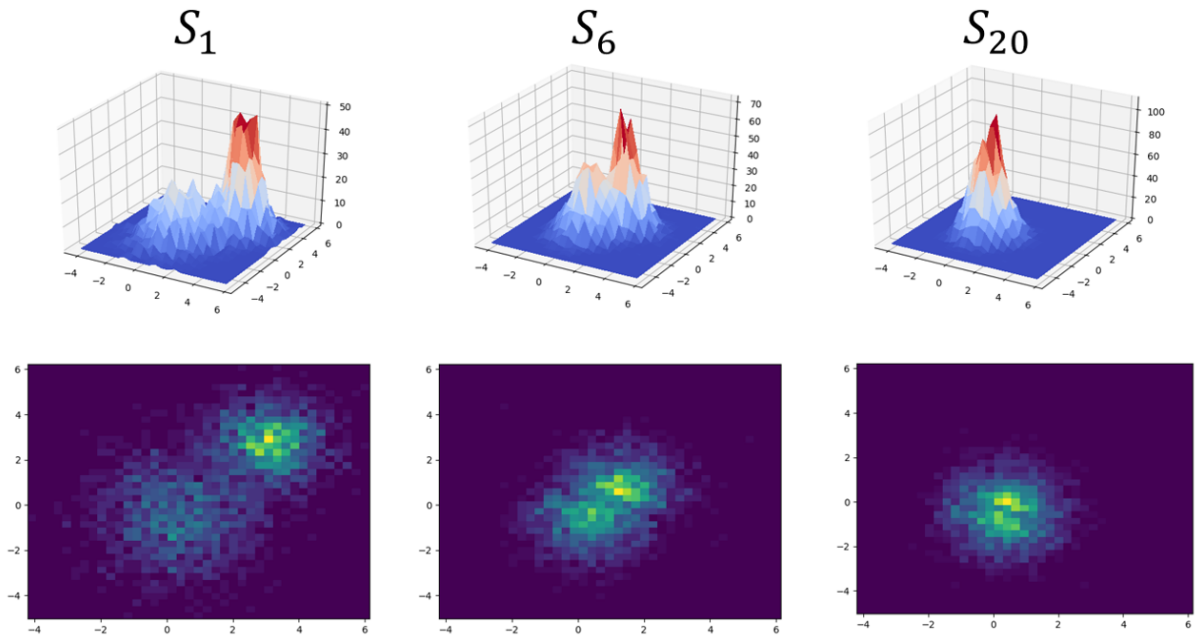


Figure 1.2. A visual representation of the effect a sequential composition has over the density of a set of samples shown at intermediary stages of the mapping sequence. P is a 2-dimensional bimodal distribution, and Q is standard Gaussian

In this section, we introduce a scheme for using many individually computed maps in sequential composition to achieve an overall effect of a single large mapping from P to Q. By using a sequence of maps to transform P to Q instead of a single one-shot map, one can

theoretically rely on models of lower complexity to represent each map in the sequence, as although each map is, on its own, "weak" in the sense of its ability to induce large changes in the distribution space, the combined action of many such maps together can potentially successfully transform samples as desired. This is especially attractive for model structures that increase exponentially in complexity with problem size, such as the dense polynomial chaos structure discussed on the previous section. This sequential composition process is visually represented in Figure 1.2.

Moving forward, we first take a brief look at a non-equilibrium thermodynamics interpretation of this methodology to further justify the use of such a scheme, and then derive a slightly different ADMM problem to implement it.

1.4.1 Non-Equilibrium Thermodynamics and Sequential Evolution of Distributions

One approach to interpreting sequential composition of maps is to borrow ideas from statistical physics, where we can interpret q as the equilibrium density (ρ_∞) of particles in a system, which at time 0 is out of equilibrium with density P (also termed ρ_0). Since q is an equilibrium density, it can be written as a Gibbs distribution (with temperature equal to 1 for simplicity): $q(u) \equiv \rho_\infty(u) = Z^{-1}\exp(-\Psi(u))$. For instance, if Q pertains to a standard Gaussian, then $\Psi(u) = \frac{1}{2}u^2$. Assuming the particles obey the Langevin equation, it is well known that the evolution of the particle density as a function of time $(\rho_t : t \geq 0)$ obeys the Fokker-Planck equation. It was shown in [29] that the trajectory of $(\rho_t : t \geq 0)$ can be interpreted from variational principles. Specifically,

Theorem 1.4.1 ([29] Thm 5.1)**.** *Define $\rho_0 = p$ and $\rho_\infty = q$ and assume that $D(\rho_0\|\rho_\infty) < \infty$. For any $h > 0$, consider the following minimization problem:*

$$A(\rho) \triangleq \frac{1}{2}d(\rho_{k-1},\rho)^2 + hD(\rho\|\rho_\infty) \tag{1.18}$$

$$\rho_k \triangleq \underset{\rho\in\mathsf{P}(\mathsf{W})}{\arg\min} A(\rho) \tag{1.19}$$

Then as $h \downarrow 0$, the piecewise constant interpolation which equals ρ_k for $t \in [kh, (k+1)h)$ converges weakly in $L_1(\mathbb{R}^D)$ to $(\rho_t : t \geq 0)$, the solution to the Fokker-Planck equation.

The log-concave structure of q we have exploited previously also has implications for exponential convergence to equilibrium with this statistical physics perspective:

Theorem 1.4.2 ([5])**.** *If q is uniform log-concave, namely*

$$\nabla^2 \Psi(u) \succeq \lambda I_D$$

for some $\lambda > 0$ with I_D the $D \times D$ identity matrix, then:

$$D(\rho_t \| \rho_\infty) \quad \leq \quad e^{-2\lambda t} D(\rho_0 \| \rho_\infty)\,.$$

Note that if q is the density of a standard Gaussian, this inequality holds with $\lambda = 1$.

1.4.2 Sequential Construction of Transport Maps

We now note that for any $h > 0$, (1.19) encodes a sequence $(\rho_k : k \geq 0)$ of densities which evolve towards $\rho_\infty \equiv q$. For notational conciseness in this section, we will be using the subscript on S to denote the position of the map in a sequence of maps. As such, from corollary Corollary 1.2.8, there exists an $S_1 \in \mathscr{D}_+$ for which $S_1 \# \rho_0 = \rho_1$, and more generally, for any $k \geq 0$, there exists an $S_k \in \mathscr{D}_+$ for which $S_k \# \rho_{k-1} = \rho_k$.

Lemma 1.4.3. *Define $B : \mathscr{D}_+ \to \mathbb{R}$ as*

$$\begin{aligned} B(S) \quad &\triangleq \quad \frac{1}{2}\mathbb{E}_{\rho_{k-1}}\left[\|X - S(X)\|^2\right] + hD(\rho_{k-1} \| \tilde{p}(\cdot; S)) \\ S_k \quad &\triangleq \quad \arg\min_{S \in \mathscr{D}_+} B(S) \end{aligned} \tag{1.20}$$

Then $A(\rho_k) = B(S_k)$ and $S_k \# \rho_{k-1} = \rho_k$.

Proof. From the definition of $\tilde{p}_{S,Q}$ in (1.4) and the invariance of relative entropy under an invertible transformation, any $S \in \mathscr{D}_+$ satisfies

$$D(\rho_{k-1}\|\tilde{p}(\cdot;S)) = D\big(\rho_{k-1}\|S^{-1}\#\rho_\infty\big) = D(S\#\rho_{k-1}\|\rho_\infty)\,.$$

As such, moving forward with the proof, we will exploit how $B(S) = \tilde{B}(S)$ where

$$\tilde{B}(S) \triangleq \frac{1}{2}\mathbb{E}_{\rho_{k-1}}\left[\|X - S(X)\|^2\right] + hD(S\#\rho_{k-1}\|\rho_\infty)\,.$$

From Theorem 1.2.7, $d(\rho_{k-1}, S\#\rho_{k-1}) \leq \mathbb{E}_{\rho_{k-1}}\left[(X - S(X))^2\right]$ for any $S \in \mathscr{D}_+$. Also, since the relative entropy terms of $\tilde{B}(S)$ and $A(S\#\rho_{k-1})$ are equal, it follows that follows that $\tilde{B}(S) \geq A(S\#\rho_{k-1})$ for any $S \in \mathscr{D}_+$. Moreover, from Corollary 1.2.8, we have that there exists an $S_k \in \mathscr{D}_+$ for which $S_k\#\rho_{k-1} = \rho_k$ and

$$\mathbb{E}_{\rho_{k-1}}\left[\|X - S_k(X)\|^2\right] = d(\rho_{k-1}, \rho_k)^2.$$

Thus $\tilde{B}(S) = A(S\#\rho_{k-1})$. □

As such, a natural composition of maps underlies how a sample from $P \equiv \rho_0$ gives rise to a sample from ρ_k:

$$\rho_k = S_k\#\rho_{k-1} = S_k \circ S_{k-1}\#\rho_{k-2} = S_k \circ \cdots \circ S_1\#\rho_0 \tag{1.21}$$

Moreover, since as $h \downarrow 0$, $\rho_k \simeq \rho_{k-1}$ and so S_k approaches the identity map. Thus for small $h > 0$, each S_k should be estimated with reasonable accuracy using lower-order maps. That is, S can be described as the composition of T maps as

$$S(x) = S_T \circ \ldots \circ S_2 \circ S_1(x) \tag{1.22}$$

for all $x \in \mathbb{R}^d$, such that each S_i is of relative low-order in the polynomial chaos expansion.

Note that $B(S)$ as written above involves a sum of expectations with respect to ρ_{k-1}. Since our scheme operates sequentially, we have already estimated $S_1, S_2, \ldots, S_{k-1}$ and can generate approximate i.i.d. samples from ρ_{k-1} by first generating $(X_i : i \geq 1)$ i.i.d. from $\rho_0 \equiv p$ and constructing

$$Z_i = S_{k-1} \circ \cdots \circ S_1(X_i), \quad i \geq 1.$$

We below will demonstrate efficient ways to solve the below convex optimization problem which replaces the expectation with respect to ρ_{k-1} instead with the empirical expectation with respect to $(Z_i : i = 1, \ldots, N)$.

$$\min_{S \in \mathscr{D}_+} \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{2} \|Z_i - S(Z_i)\|^2 - h \log \tilde{p}(Z_i; S) \right]$$

To reiterate, we consider a distribution ρ_{k-1} formed by the sequential composition of *previous* mappings as

$$\rho_{k-1} = S_{k-1}^* \circ \cdots \circ S_1^* \# \rho_0,$$

where $\rho_0 \equiv p$. We then try to find a map S_k^* that pushes ρ_{k-1} forward closer to $\rho_\infty \equiv Q$. Each S_k is solved by the optimization problem (1.20), which we term **SOT**. As the number of compositions T in (1.22) increases, ρ_T approaches ρ_∞. When q is uniform log-concave, this greedy, sequential approach still guarantees exponential convergence.

In the context of Knothe-Rosenblatt maps, for every map in the sequence we can solve the following optimization problem (in the following equation, we will be dropping the subscript k that indicates the sequential map index, as the formulation is not dependent on position in the map sequence, and we will once again be replacing the subscript with i to indicate the distributed

variables for the consensus problem instead):

$$\min_{S_i \in \mathscr{D}_+^{KR}} \theta ||S_i(X_i) - X_i||_2^2 - \frac{1}{N}\sum_{i=1}^{N} \log q(S_i(X_i)) + \sum_{d=1}^{D} \log \partial_d S^d(X_i) \tag{1.23}$$
$$\text{s.t.} \quad S_i = S, \quad \forall 0 \leq i \leq N$$

where $\theta = h^{-1}$ can be interpreted as an inverse "step-size" parameter.

Though each map in the sequence must be calculated *sequentially* after the previous one, each mapping can still be calculated in the distributed framework described above. This implies that at each round, one could *adaptively* decide the parameters for the next-round's solve.

1.4.3 ADMM Formulation for Learning Sequential Maps

We now showcase an ADMM formulation for the optimal transportation-based objective function, similar in spirit to that of Eq. (1.12).

We first introduce the following conventions:

- Φ_i^d represents the partial derivative of Φ_i taken with respect to the d^{th} component. Therefore, $B\Phi_i^d = \partial_d S(X_i)$, and $\partial_d S^d(X_i)$ is the d^{th} component of $B\Phi_i^d$.
- $\mathbf{1}_d$ represents a one-hot vector of length D with the one in the d^{th} position

We can then introduce a finite-dimensional representation of the transport map, as well as auxiliary variables and a consensus variable to Eq. (1.23) and rewrite the problem as:

$$
\begin{aligned}
&\min_{\{W,p\}_i,\{Y,Z\}_i^d,B} \theta||B\Phi_i - x_i||_2^2 + \frac{1}{N}\sum_{i=1}^{N} -\log q(p_i) \\
&+ \frac{1}{2}\rho||W_i - B||_2^2 + \frac{1}{2}||B\Phi_i - p_i||_2^2 \\
&+ \sum_{d=1}^{D} -\log Z_i^d + \frac{1}{2}\rho(Y_i^d\mathbf{1}_d - Z_i^d)^2 + \frac{1}{2}\rho||B\Phi_i^d - Y_i^d||_2^2 \\
\text{s.t} \quad & B\Phi_i = p_i \qquad \gamma_i \quad (D\times 1) \\
& W_i - B = 0 \qquad \alpha_i \quad (D\times K) \\
& Y_i^d\mathbf{1}_d = Z_i^d \qquad \beta_i^d \quad (1\times 1) \\
& B\Phi_i^d = Y_i^d \qquad \lambda_i^d \quad (D\times 1) \\
& Z_i^d > 0
\end{aligned}
\tag{1.24}
$$

where we have once again denoted the corresponding Lagrange multipliers to the right of each constraint. The superscript d notation represents the fact that in this formulation, in addition to having separable variables for each data sample, some variables are now unique to an index over dimension as well. For example, there are DN-many Z variables that must be solved for. We can now raise the constraints to form the fully-penalized Lagrangian as:

$$
\begin{aligned}
&L_{\rho,\theta}(W,Z,Y,p,B;\gamma,\alpha,\beta,\lambda) \\
&= \theta||B\Phi_i - x_i||_2^2 + \frac{1}{N}\sum_{i=1}^{N} -\log q(p_i) \\
&+ \frac{1}{2}\rho||W_i - B||_2^2 + \frac{1}{2}\rho||B\Phi_i - p_i||_2^2 \\
&+ \gamma_i^T(p_i - B\Phi_i) + \mathbf{tr}(\alpha_i^T(F_i - B)) \\
&+ \sum_{d=1}^{D} -\log Z_i^d + \frac{1}{2}\rho(Y_i^d\mathbf{1}_d - Z_i^d)^2 + \frac{1}{2}\rho||B\Phi_i^d - Y_i^d||_2^2 \\
&+ \beta_i^d(Z_i^d - Y_i^d\mathbf{1}_d) + \lambda_i^{dT}(Y_i^d - B\Phi_i^d)
\end{aligned}
\tag{1.25}
$$

The final ADMM update equations for each variable are once again all closed-form, with the exception of the optimization over p_i. For the sake of brevity, we refer the reader to Section 1.8 of the Appendix for the exact update equations.

However, one notable difference between this formulation and that of Section 1.3.4 as noted in the previous section is that the update for Z_i^d has been simplified from requiring an eigenvalue decomposition, to requiring a simple scalar computation, thus significantly reducing computation time, especially in higher dimensions.

1.4.4 Scaling Parallelization with GPU Hardware

Given the parallelized formulations given above, we implemented our algorithm using the Nvidia CUDA API to get as much performance as possible out of our formulation, and to maximize the problem sizes we could reasonably handle, while keeping computation time as short as possible. To test the algorithm's parallelizability, we ran our implementation on a single Nvidia GTX 1080ti GPU, as well as on a single p3.16xlarge instance available on Amazon Web Services, which itself contains 8 on-board Tesla V100 GPUs.

For this test, we have sampled synthetic data from a bimodal P distribution specified as a combination of two Gaussian distributions, for a wide range of problem dimensions, specifically $D = 5, 10, 20, 50, 100, 150, 200$, and a constant number of samples from P set to $N = 1000$. We then find a transport pushing P to $Q = \mathcal{N}(\mathbf{0}, \mathbf{I})$, composed of a sequence of 10 individual Knothe-Rosenblatt maps with no mixed multivariate terms. We then monitor the convergence of dual variables for proper termination of the algorithm.

Figure 1.3 shows the result of this analysis. The 1 GPU curve corresponds to performance using the single GTX 1080ti, and the AWS curve corresponds to the performance using the 8-GPU system on Amazon Web Services. The trending of the curves shows that, as expected, as problem dimension increases, a multi-GPU system will continue to maintain reasonable computation times, at least with respect to a single-GPU system, however fewer GPU's will begin to accumulate increasingly high computational costs. In addition, the parallelizability

of our algorithm also has a subtle benefit of helping with memory-usage issues; since we can distribute samples across multiple devices, we can also subsequently distribute all corresponding ADMM variables as well. Indeed, the single GTX 1080ti ran out of on-board memory roughly around $D = 230$, whereas the 8-GPU system can go well beyond that.

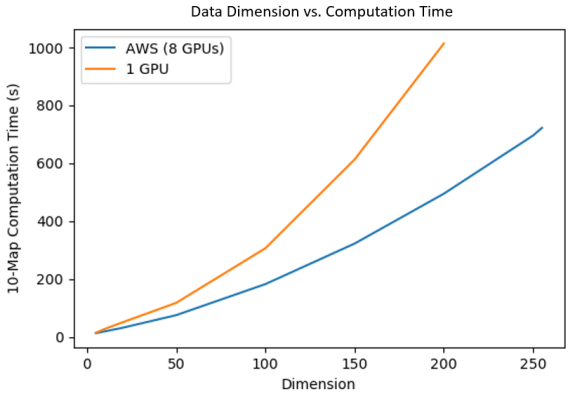


Figure 1.3. A comparison of using a single-GPU system vs. an 8-GPU system to compute maps in increasingly high dimension. The trending of the two plots clearly shows the more reasonable growth in computation time of the 8-GPU system relative to the single-GPU system, as the samples from P are distributed among the multiple devices

1.5 Applications

The framework presented above is general-purpose, and works to push-forward a distribution P to a log-concave distribution Q. Below we discuss some interesting applications, namely Bayesian inference and a generative model, and show results with real-world datasets.

1.5.1 Bayesian Inference

A very important instantiation of this framework comes when we consider $P \equiv P_X$ to represent a prior distribution, and $Q \equiv P_{X|Y=y}$ to be a Bayesian posterior:

$$f_{X|Y=y}(x) = \frac{f_{Y|X}(y|x) f_X(x)}{\beta_y}$$

where β_y is a constant that does not vary with x, given by:

$$\beta_y = \int_{v \in \mathscr{X}} f_{Y|X}(y|v) f_X(v) dv$$

Using Eq. (3.1) and combining with Bayes' rule above we can write:

$$\begin{aligned} f_X(x) &= f_{X|Y=y}(S^*_{(y)}(x)) \det\left(J_{S^*_{(y)}(x)}\right) \\ &= \frac{f_{Y|X}(y|S^*_{(y)}(x)) f_X\left(S^*_{(y)}(x)\right)}{\beta_y} \det\left(J_{S^*_{(y)}(x)}\right) \end{aligned}$$

where the notation $S^*_{(y)}(x)$ indicates that the optimal map is found with respect to observations y. We note that since $q(u) = \frac{f_X(u) f_{Y|X}(y|u)}{\beta_y}$, log-concavity of q is equivalent to log-concavity of the prior density $f_X(u)$ and log-concavity of the likelihood density $f_{Y|X}(y|u)$ in u: the same criterion for an MAP estimation procedure to be convex. Thus Corollary 1.3.2 extends to the special case of Bayesian inference; i.e. we can generate i.i.d. samples from the posterior distribution by solving a convex optimization problem in a distributed fashion.

Due to the unique way the ADMM steps were structured, this special case only requires specifying a particular instance of Eq. (1.12g):

$$p_i^* = \arg\min_{p_i} -\log \underbrace{f_{Y|X}(y|p_i)}_{\text{likelihood}} - \log \underbrace{f_X(p_i)}_{\text{prior}} + \text{pen}(p_i)$$

Remark 5. *This specific case establishes an important property. If the prior is chosen so that it is easy to sample from, and the prior and likelihood are both log-concave, then a deterministic function S can be efficiently computed that takes I.I.D samples from the prior distribution, and results in I.I.D samples from the posterior distribution. The assumption of log-concavity is also typically used in large-scale point estimates, though this framework goes beyond point estimates and generates I.I.D samples form the posterior.*

As an instantiation of this framework, we consider a Bayesian estimation of regression

parameters $x_1, ..., x_d$ in the model $y = \mu \mathbf{1_n} + \Phi x + \varepsilon$, where y is the n-dimensional vector of responses, μ is the overall mean, Φ is a $n \times d$ regressor matrix, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is a noise vector. The LASSO solution,

$$x^* = \arg\min_{x \in \mathbb{R}^d} ||y - \Phi x||_2^2 + \lambda ||x||_1 \tag{1.26}$$

for some $\lambda \geq 0$ induces sparsity in the latent coefficients. The solution to (2.1) can be seen as a posterior mode estimate when the regression parameters are distributed accordingly to a Laplacian prior.

$$p(x; \lambda) = \prod_{i=1}^{d} \frac{\lambda}{2} e^{-\lambda |x_i|} \tag{1.27}$$

A number of Bayesian LASSO Gibbs samplers, which are Markov Chain Monte Carlo algorithms, are used as standard methods by which to sample from the posterior associated with problem (2.1) [43], [24].

We study the accuracy and modularity of our measure transport methodology through a Bayesian LASSO analysis of the Boston Housing data set, first analyzed by Harrison and Rubinfeld [25], which is a common dataset used when comparing regression problems. We compare our results to those obtained from utilizing a corresponding Gibbs sampler. The Boston Housing data set consists of 13 independent predictors of the median value of owner occupied homes and 506 cases. We are interested in which combination of these 13 variables best predict the median value of homes observed in y, and if we can eliminate variables that do not contribute much to prediction. The LASSO gives an automatic way for feature selection by forcing the coefficients of the predictors represented by x^* to be zero. The Bayesian LASSO solution, allows for uncertainty quantification of feature selection, as we can obtain credible intervals corresponding to the coefficients of the estimates.

We used a Gibbs sampler as presented in [24] where the variance variable σ^2 is non-random. We used 3000 samples of burn-in and sampled 10000 samples from the posterior distribution with a fixed λ chosen by minimizing the Bayes Information Criterion (BIC) [59]. We

compared that to sampling from a generated transport map with the same λ. We used $N = 2000$ samples from a Laplace prior to learn a fourth-order transport map of interest. In this case, we used a one-shot, dense map structure as described in Section Section 1.3.

We note that the modularity of our problem allows for sampling from the posterior distribution of the Bayesian LASSO, by only specifying the optimization problem of Eq. (1.12g) to correspond to the likelihood and prior.

Figure 1.4 shows the posterior median estimates and the corresponding 95% credible intervals for the marginal distributions of the first 10 variables of the Boston housing data set. The LASSO estimates are shown for comparison. Figure 1.5 shows the Kernel Density Estimates for these variables constructed with 10000 samples of either the Bayesian LASSO Gibbs sampler or the measure transported samples. The density estimates of both methods are similar, verifying the accuracy of our methodology.

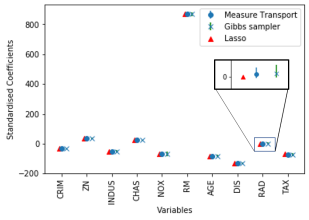


Figure 1.4. Posterior median Bayesian LASSO estimates and corresponding credible intervals for the ten first variables of the Boston Housing dataset. Median estimates were obtained with samples from a Gibbs sampler and a Measure Transport map. LASSO estimates are shown for comparison.

1.5.2 High-Dimensional Maps Using the MNIST Dataset

The parallelizability of our formulation of the optimal transportation-based mapping for sequential transport maps also allows us to efficiently compute maps for relatively high-dimensional data. As a demonstration of this, we used the MNIST handwritten digits dataset [34] as a subject of experimentation.

Similar to the density estimation case, we assume that samples from each class of MNIST data is drawn from some P_{digit}, where $digit$ denotes the MNIST written digit associated with that

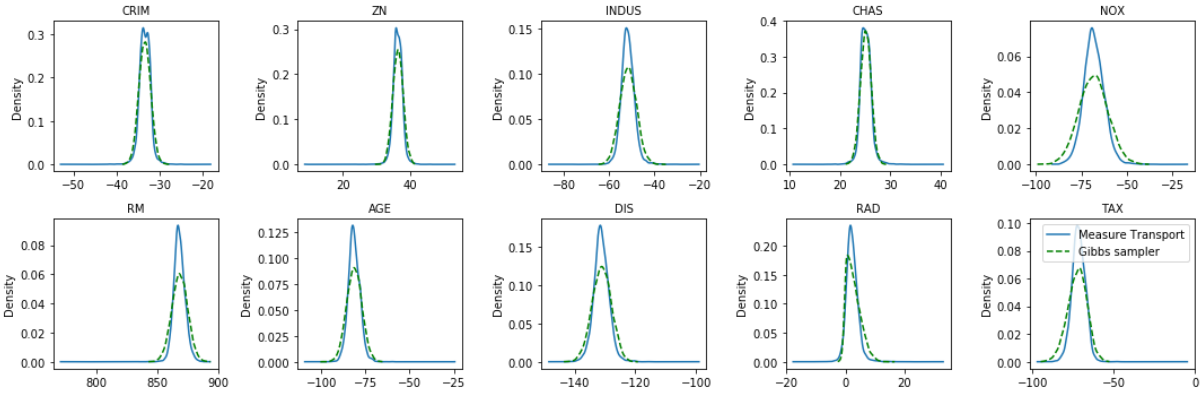


Figure 1.5. Kernel Density Estimate comparisons of marginal posteriors for the Boston Housing data set.

distribution. We then attempt to construct a (sequential) mapping, $S_{(digit)}$ that pushes P_{digit} to a reference distribution, $Q = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Again, similar to before, the selection of the Q density to be a standard Gaussian is expressly for the purpose of analytical simplicity; Q can theoretically be anything we like, so it benefits us during the generative step to select Q such that it is easy to sample from. Each image in MNIST is a 28x28 pixel image, therefore after flattening each image into a vector of data, our maps operate in $\mathbf{D} = \mathbf{784}$. We then solve for each map $S_{(digit)}$ for every handwritten digit class in the MNIST set.

We can then treat the inverse map as a generative model; with the maps $S_{(digit)}$ in hand, we can theoretically draw samples from Q, and push these samples through the inverse map, $S^{-1}_{(digit)}$, resulting in randomly generated samples from P_{digit}.

Fig. 1.6 shows the result of this process using a sequential composition of 15 maps, with maximum order of the basis of each sequential map being set to 2, and each sequential map using the Knothe-Rosenblatt basis with no mixed multivariate terms from Section 1.3.5. Our results show that even in high dimension, and even while using a relatively weak polynomial basis per sequential map, the resulting transport maps can effectively generate approximate samples from P_{digit} in this way.

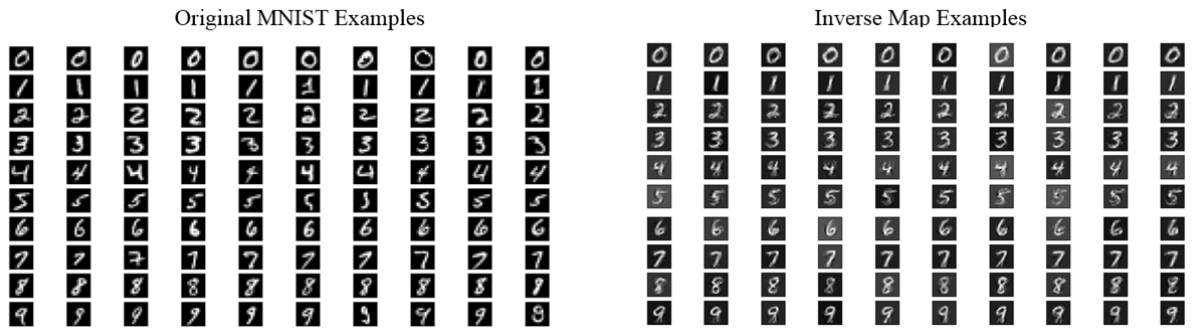


Figure 1.6. A comparison of original MNIST data samples vs. random samples drawn using the inverse map. The left-most 10 columns of images pertain to randomly selected data examples from the original MNIST set, and the rightmost 10 columns of images are randomly generated by the inverse map, $S^{-1}_{(digit)}(X), X \sim Q$. Each mapping for this example was a sequential composition of 15 maps of maximum order 2, using the Knothe-Rosenblatt mapping with no mixed terms.

1.6 Discussion

In this work, we have proposed a general purpose framework for pushing independent samples from one distribution P to independent samples from another distribution Q through the *efficient* and *distributed* construction of transport maps, with only independent samples from P, and knowledge of Q up to a normalization constant. We showed that when the target distribution Q is log-concave, this problem is *convex*. Using ADMM, we instantiated two finite dimensional problems for finding both one-shot and sequential transport maps, and provided distributed algorithms for carrying out the underlying optimization problems. As our framework is distributed by nature, we can continue to take advantage of the ever-increasing availability and evolution of distributed computational resources to further speed up computation, with little to no changes to our formulation whatsoever.

We applied our framework to a Bayesian LASSO problem, that, while it requires that the prior and likelihood to be log-concave, is no different than existing frameworks that carry out efficient point estimates in that regard; however, by contrast, our framework does succeed in efficiently generating *independent* samples from the actual target distribution Q. We emphasize that the class of log-concave distributions is quite large and widely used in various applications

[4], and that this is the same convexity condition required for Bayesian point (MAP) estimation using many modern techniques. As such, we have shown that from the perspective of convexity, we can go from point estimation to fully Bayesian estimation, without requiring significantly more.

Finally, we applied our framework to a high-dimensional problem of approximating a generative model for the MNIST dataset, and provided a qualitatively striking demonstration of how well the construction of sequential transport maps can give rise to such a model. The connection and comparison of this method to other generative models, especially deep learning-based methods such as generative adversarial networks [23] and variational autoencoders [32], remains to be explored and is the subject of future work. We believe that this alternate form of generative model, one based on calculating a transport map that is parameterized over the space of polynomial basis functions orthogonal to the distribution of the data, stands in contrast to the black-box nature of neural networks. Moreover, although certain works have explored the invertibility of deep neural networks [36], [21], in general a single output of a neural network might map to multiple latent vectors. Our transport maps, chosen over the space of diffeomorphisms, remain necessarily invertible and indeed this property is exploited in the generation of samples. One can surmise that this invertibility leads to more tractability of the generative model. The general connection to Optimal Transport and deep generative models is a subject of recent interest and has incited pertinent work in the literature [19], [48].

We also stress that ADMM and other related large-scale optimization methods have many existing refinements [27, 29, 58, 3] from which this framework would immediately benefit. Future work could explore these refinements, and applications as approximations to non-convex problems.

Although we have established convexity of these schemes, further work needs to be done characterizing the fundamental limits of sample complexity of this approach, and can help guide how these architectures may possibly be soundly implemented. Optimizing architectures for hardware optimization, and understanding performance-energy-complexity trade-offs, will

further allow for wider exploration of these methods within the context of emerging applications.

1.7 Acknowledgements

Chapter 1, in full, has been submitted for publication of the material as it may appear in Neural Computation 2018. Mesa, Diego A; Tantiongloc, Justin; Mendoza, Marcela, Coleman, Todd. The dissertation author was the co-primary investigator and author of this paper along with Justin Tantiongloc and Diego Mesa.

1.8 Appendix

Here we provide some additional details on several aspects of the main paper.

Derivation of Dense ADMM Formulation

Here we show a more complete derivation of the ADMM formulation from Section 1.3.4. ADMM yields the following sequential updates to the penalized Lagrangian:

$$B^{k+1} = \arg\min_{B} L_\rho(W^k, Z^k, p^k, B; \gamma^k, \lambda^k, \alpha^k) \tag{1.28a}$$

$$W^{k+1} = \arg\min_{W} L_\rho(W, Z^k, p^k, B^{k+1}; \gamma^k, \lambda^k, \alpha^k) \tag{1.28b}$$

$$Z^{k+1} = \arg\min_{Z \succ 0} L_\rho(W^{k+1}, Z, p^k, B^{k+1}; \gamma^k, \lambda^k, \alpha^k) \tag{1.28c}$$

$$p^{k+1} = \arg\min_{p} L(W^{k+1}, Z^{k+1}, p, B^{k+1}; \gamma^k, \lambda^k, \alpha^k) \tag{1.28d}$$

$$\gamma_i^{k+1} = \gamma_i^k + \rho(p_i^{k+1} - B^{k+1}\Phi_i) \quad 1 \le i \le n \tag{1.28e}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(Z_i^{k+1} - B^{k+1}J_i) \quad 1 \le i \le n \tag{1.28f}$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho(W_i^{k+1} - B^{k+1}) \quad 1 \le i \le n \tag{1.28g}$$

The closed form solutions to the equations (2.24), (2.26a), and (2.26b) are given as

follows:

Firstly, as for (2.24), the cost function $C(B^{k+1})$ is given by:

$$\begin{aligned} C(B^{k+1}) = & \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\rho\|W_i^k - B\|_F^2 + \frac{1}{2}\rho\|B\Phi_i - p_i^k\|_2^2 \\ & + \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\rho\|BJ_i - Z_i^k\|_F^2 + \gamma_i^{kT}(p_i^k - B\Phi_i) \\ & + \frac{1}{N}\sum_{i=1}^{N}\mathbf{tr}\left(\lambda_i^{kT}(Z_i^k - BJ_i)\right) + \mathbf{tr}\left(\alpha_i^{kT}(W_i^k - B)\right). \end{aligned} \tag{1.29}$$

The first-order derivative of the equation (1.29) in terms of B^{k+1} is expressed as

$$\begin{aligned} \frac{\partial C(B^{k+1})}{\partial B^{k+1}} = & \frac{1}{N}\sum_{i=1}^{N}\rho(B - W_i^k) + \rho(B\Phi_i - p_i^k)\Phi_i^T \\ & + \frac{1}{N}\sum_{i=1}^{N}\rho(BJ_i - Z_i^k)J_i^T - \gamma_i^k\Phi_i^T \\ & + \frac{1}{N}\sum_{i=1}^{N} -\lambda_i^k J_i - \alpha_i^{kT}. \end{aligned} \tag{1.30}$$

By setting the equation (1.30) to zero and expressing it in terms of B, we get

$$\begin{aligned} & B\left[\rho\left(I + \frac{1}{N}\sum_{i=1}^{N}\Phi_i\Phi_i^T + J_iJ_i^T\right)\right] \\ & = \frac{1}{N}\sum_{i=1}^{N}\left[\rho\left(W_i^k + p_i^k\Phi_i^T + Z_i^kJ_i^T\right) + \gamma_i^k\Phi_i^T + \lambda_i^kJ_i^T + \alpha_i^k\right]. \end{aligned} \tag{1.31}$$

If we define

$$L \triangleq \left[\rho\left(I + \frac{1}{N}\sum_{i=1}^{N}\Phi_i\Phi_i^T + J_iJ_i^T\right)\right] \tag{1.32}$$

and

$$M \triangleq \frac{1}{N}\sum_{i=1}^{N}\left[\rho\left(W_i^k + p_i^k\Phi_i^T + Z_i^kJ_i^T\right) + \gamma_i^k\Phi_i^T + \lambda_i^kJ_i^T + \alpha_i^k\right] \tag{1.33}$$

Then we have:

$$B^{k+1} = M \cdot L^{-1} \tag{1.34}$$

Secondly, as for (2.26a), the cost function $C(W_i^{k+1})$ is given by

$$C(W_i^{k+1}) = \frac{1}{2}\rho \|W_i - B^{k+1}\|_2^2 + \mathbf{tr}\left(\alpha_i^{kT}(W_i - B^{k+1})\right) \tag{1.35}$$

The first-order derivative of the equation (1.35) in terms of W_i^{k+1} is expressed as

$$\frac{\partial C(W_i^{k+1})}{\partial W_i^{k+1}} = \rho(W_i - B^{k+1}) + \alpha_i^k. \tag{1.36}$$

Thus,

$$W_i^{k+1} = -\frac{1}{\rho}\alpha_i^k + B^{k+1} \tag{1.37}$$

Lastly, as for (2.26b), following the steps in [12], the first-order optimality condition using the equation (2.26b) is expressed as

$$-Z_i^{-1} + \rho(Z_i - B^{k+1}J_i) + \lambda_i^k = 0. \tag{1.38}$$

Rewriting this, we get

$$\rho Z_i - Z_i^{-1} = \rho B^{k+1} J_i - \lambda_i^k. \tag{1.39}$$

First, take the orthogonal eigenvalue decomposition of the right-hand side,

$$\rho B^{k+1} J_i - \lambda_i^k = Q \Lambda Q^T \tag{1.40}$$

where $\Lambda = \mathbf{diag}(v_1, ..., v_d)$, and $Q^T Q = Q Q^T = I$. Multiplying (1.39) by Q^T on the left and by Q on the right gives

$$\rho \tilde{Z}_i - \tilde{Z}_i^{-1} = \Lambda \tag{1.41}$$

where $\tilde{Z}_i = Q^T Z_i Q$. A diagonal solution of this equation is given by

$$\tilde{Z}_{i,(jj)} = \frac{v_j + \sqrt{v_j^2 + 4\rho}}{2\rho}, \tag{1.42}$$

and the final solution is given as

$$Z_i^{k+1} = Q \tilde{Z}_i Q^T. \tag{1.43}$$

Derivation of Knothe-Rosenblatt ADMM Formulation and Final Updates

In similar fashion, here we outline the derivation of the ADMM formulation from Section 1.4.3.

First, we note that the closed-form updates for W_i and p_i are identical as for the original formulation. So here we will show the derivation only for the remainder of updates. In what follows, ADMM iteration superscripts, k, are now enclosed in parentheses so as not to confuse them with the d superscript indexing over dimension:

The cost function $C(B^{(k+1)})$ is given by:

$$\begin{aligned} C(B^{(k+1)}) = & \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\rho||W_i^{(k)} - B||_2^2 + \theta||B\Phi_i - X_i||_2^2 \\ & + \frac{1}{2}\rho||B\Phi_i - p_i^{(k)}||_2^2 + \gamma_i^{(k)T}(p_i^{(k)} - B\Phi_i) + \\ & + \mathbf{tr}(\alpha_i^{(k)T}(W_i^{(k)} - B)) \\ & + \sum_{d=1}^{D}\frac{1}{2}\rho||B\Phi_i^d - Y_i^{d(k)}||_2^2 + \lambda_i^{d(k)T}(Y_i^{d(k)} - B\Phi_i^d) \end{aligned} \tag{1.44}$$

Taking the first-order derivative of Eq. (1.44) and setting to 0, we arrive at the following expression:

$$\begin{aligned} & B\left[\rho(\mathbf{I} + \frac{1}{N}\sum_{i=1}^{N}\Phi_i\Phi_i^T + \frac{2\theta}{\rho}\Phi_i\Phi_i^T + \sum_{d=1}^{D}\Phi_i^d\Phi_i^{dT})\right] \\ & = \frac{1}{N}\sum_{i=1}^{N}\rho W_i^{(k)} + \rho p_i^{(k)}\Phi_i^T + 2\theta X_i\Phi_i^T + \gamma_i^{(k)}\Phi_i^T + \alpha_i^{(k)T} \\ & + \sum_{d=1}^{D}\rho Y_i^{d(k)}\Phi_i^{dT} + \lambda_i^{d(k)}\Phi_i^{dT} \end{aligned} \tag{1.45}$$

If we define

$$\mathscr{B}_s \triangleq \rho\left(\mathbf{I} + \frac{1}{N}\sum_{i=1}^{N}\Phi_i\Phi_i^T + \frac{2\theta}{\rho}\Phi_i\Phi_i^T + \sum_{d=1}^{D}\Phi_i^d\Phi_i^{dT}\right) \tag{1.46}$$

and

$$\begin{aligned}\mathscr{B}_i \triangleq \frac{1}{N}\sum_{i=1}^{N} \rho W_i^{(k)} + \rho p_i^{(k)}\Phi_i^T + 2\theta X_i \Phi_i^T + \gamma_i^{(k)}\Phi_i^T + \alpha_i^{(k)T} \\ + \sum_{d=1}^{D} \rho Y_i^{d(k)}\Phi_i^{dT} + \lambda_i^{d(k)}\Phi_i^{dT}\end{aligned} \tag{1.47}$$

then we have:

$$B^{(k+1)} = \mathscr{B}_i \cdot \mathscr{B}_s^{-1} \tag{1.48}$$

The loss function associated with Z_i^d for a given i and d is the following:

$$\begin{aligned}C(Z_i^{d(k+1)}) = -\log Z_i^d + \frac{1}{2}\rho(Y_i^{d(k)}\mathbf{1}_d - Z_i^d)^2 \\ + \beta_i^{d(k)}(Z_i^d - Y_i^{d(k)}\mathbf{1}_d)\end{aligned}$$

Taking the derivative and setting to 0, we get the following quadratic expression:

$$\rho Z_i^{d2} + (\beta_i^{d(k)} - \rho Y_i^{d(k)}\mathbf{1}_d)Z_i^d - 1 = 0 \tag{1.49}$$

As we would like $Z_i^{d(k+1)}$ to be greater than 0 according to our constraints, we set the closed-form solution to the positive root of this quadratic equation:

$$Z_i^{d(k+1)} = \frac{\rho Y_i^{d(k)}\mathbf{1}_d - \beta_i^{d(k)} + \sqrt{(\rho Y_i^{d(k)}\mathbf{1}_d - \beta_i^{d(k)})^2 + 4\rho}}{2\rho} \tag{1.50}$$

The loss function associated with Y_i^d for a given i and d is the following:

$$\begin{aligned}C(Y_i^{d(k+1)}) = \frac{1}{2}\rho(Y_i^d\mathbf{1}_d - Z_i^{d(k+1)})^2 + \frac{1}{2}\rho||B^{(k+1)}\Phi_i^d - Y_i^d||_2^2 \\ + \beta_i^{d(k)}(Z_i^{d(k+1)} - Y_i^d\mathbf{1}_d) + \lambda_i^{d(k)T}(Y_i^d - B^{(k+1)}\Phi_i^d)\end{aligned} \tag{1.51}$$

Taking the derivative with respect to Y_i^d and setting to 0, we get the following expression:

$$Y_i^{d(k+1)} = (\rho Z_i^{d(k+1)} \mathbf{1}_d^T + \rho B^{(k+1)} \Phi_i^d + \beta_i^{d(k)} \mathbf{1}_d^T - \lambda_i^{d(k)T}) \cdot (\rho \mathbf{1}_d \mathbf{1}_d^T + \rho \mathbf{I})^{-1} \tag{1.52}$$

Finally, our complete set of updates is:

$$B^{(k+1)} = \mathscr{B}_i \cdot \mathscr{B}_s \tag{1.53a}$$

$$W_i^{(k+1)} = -\frac{1}{\rho}\alpha_i^{(k)} + B^{(k+1)} \tag{1.53b}$$

$$Z_i^{d(k+1)} = \frac{\rho Y_i^{d(k)} \mathbf{1}_d - \beta_i^{d(k)} + \sqrt{(\rho Y_i^{d(k)} \mathbf{1}_d - \beta_i^{d(k)})^2 + 4\rho}}{2\rho} \tag{1.53c}$$

$$Y_i^{d(k+1)} = (\rho Z_i^{d(k+1)} \mathbf{1}_d^T + \rho B^{(k+1)} \Phi_i^d + \beta_i^{d(k)} \mathbf{1}_d^T - \lambda_i^{d(k)T}) \cdot (\rho \mathbf{1}_d \mathbf{1}_d^T + \rho \mathbf{I})^{-1} \tag{1.53d}$$

$$\gamma_i^{(k+1)} = \gamma_i^{(k)} + \rho(p_i^{(k+1)} - B^{(k+1)} \Phi_i) \tag{1.53e}$$

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} + \rho(W_i^{(k+1)} - B^{(k+1)}) \tag{1.53f}$$

$$\lambda_i^{d(k+1)} = \lambda_i^{d(k)} + \rho(Y_i^{d(k+1)} - B^{(k+1)} \Phi_i^d) \tag{1.53g}$$

$$\beta_i^{d(k+1)} = \beta_i^{d(k)} + \rho(Z_i^{d(k+1)} - Y_i^{d(k+1)} \mathbf{1}_d) \tag{1.53h}$$

$$p_i^{(k+1)} = \arg\min_{p_i} -\log q(p_i) + \text{pen}(p_i) \tag{1.53i}$$

where the p_i update can once again be performed using any number of appropriate optimization techniques.

Transport Map Multi-Indices Details

In this section, we give a few concrete examples of the various multi-index-sets presented in Section 1.3.5 for clarification in practical use-cases, as well as for actual implementation purposes.

In the case of a dense map, recall the index set:

$$\mathscr{J}^D = \left\{ \mathbf{j} \in \mathbb{N}^d : \sum_{i=1}^{d} j_i \leq O \right\}$$

For example, in the case where $D = O = 3$, the resulting index set will have the following form:

$$\mathscr{J}^D = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 2 & 1 & 0 & 1 & 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \end{bmatrix}$$

where every $\mathbf{j}^{th}$ column is one D-long multi-index for a single multivariate polynomial basis term, $\phi_{\mathbf{j}}$.

The size of this set $K \triangleq |\mathscr{J}^D|$ for any given maximum polynomial order O is:

$$K = \binom{D+O}{O}$$

In the case of the Total Order Knothe-Rosenblatt map, the index set is:

$$\mathscr{J}_d^{KR} = \left\{ \mathbf{j} \in \mathbb{N}^d : \sum_{i=1}^{d} j_i \leq O \wedge j_i = 0, \forall i > d \right\}, d = 1, \ldots, D$$

In this case, the size of the set $K_d \triangleq |\mathscr{J}_d^{KR}|$ becomes dependent on the component of the mapping.

Revisiting our previous example with $D = O = 3$ we have:

$$\begin{aligned}
\mathscr{J}_1^{KR} &= \left\{ \mathbf{j} \in \mathbb{N}^3 : \sum_{i=1}^{3} j_i \leq O \wedge j_2 = j_3 = 0 \right\} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathscr{J}_2^{KR} &= \left\{ \mathbf{j} \in \mathbb{N}^3 : \sum_{i=1}^{3} j_i \leq O \wedge j_3 = 0 \right\} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathscr{J}_3^{KR} &= \left\{ \mathbf{j} \in \mathbb{N}^3 : \sum_{i=1}^{3} j_i \leq O \right\} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 2 & 1 & 0 & 1 & 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \end{bmatrix}
\end{aligned}$$

In contrast to a dense mapping, this construction yields a weight matrix that has

$$|\mathscr{J}_d^{KR}| = \binom{d+O}{O} \tag{1.54}$$

many non-zero weights per row d, for a total of:

$$\sum_{d=1}^{D} \binom{d+O}{O} \tag{1.55}$$

non-zero weights. In terms of implementation, note that we can enforce a lower-triangular structure of the mapping simply by constructing Φ according to the full index set ordering of $\mathscr{J}_D^{KR}$, and constraining the coefficient matrix W to have zeros embedded with the following structure:

Definition 1.8.1 (Lower-Triangular Weight Matrix)**.** *A weight matrix $W \in \mathbb{R}^{D \times K}$ corresponds to*

a lower-triangular transport map if it can be expressed as:

$$W = \begin{bmatrix} \boldsymbol{w}_1^T & 0 & 0 & 0 & 0 & 0 & 0 \\ & \dots & \boldsymbol{w}_d^T & \dots & 0 & 0 & 0 \\ & & \dots & \boldsymbol{w}_D^T & \dots & & \end{bmatrix}$$

where each $\boldsymbol{w}_d$ *is a vector in* $\mathbb{R}^{|\mathscr{J}_d^{KR}|}$.

When constructed as such, $W\Phi_i = S(X_i)$, where S is a Knothe-Rosenblatt map.

In the case of the Single Univariate Knothe-Rosenblatt map, the index set becomes the following subset of $\mathscr{J}^{KR}$, again dependent on the component d:

$$\mathscr{J}_d^{KRSV} = \left\{ \mathbf{j} \in \mathbb{N}^d : \sum_{i=1}^{d} j_i \leq O \wedge j_i j_l = 0, \forall i \neq l \wedge j_i = 0, \forall i > d \right\},$$
$$d = 1, \dots, D$$

Revisiting our previous example with $D = O = 3$, we have the following multi-index sets:

$$\begin{aligned}
\mathscr{J}_1^{KRSV} &= \left\{ \mathbf{j} \in \mathbb{N}^3 : \sum_{i=1}^{3} j_i \leq O \wedge j_2 = j_3 = 0 \wedge j_i j_l = 0, \forall i \neq l \right\} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathscr{J}_2^{KRSV} &= \left\{ \mathbf{j} \in \mathbb{N}^3 : \sum_{i=1}^{3} j_i \leq O \wedge j_3 = 0 \wedge j_i j_l = 0, \forall i \neq l \right\} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathscr{J}_3^{KRSV} &= \left\{ \mathbf{j} \in \mathbb{N}^3 : \sum_{i=1}^{3} j_i \leq O \wedge j_i j_l = 0, \forall i \neq l \right\} \\
&= \begin{bmatrix} 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \end{bmatrix}
\end{aligned}$$

Here, all multivariate polynomial basis terms that are a product of mixed univariate

polynomial terms are eliminated from the basis, resulting in a weight matrix that has:

$$|\mathscr{J}_d^{KRSV}| = dO + 1 \tag{1.56}$$

many non-zero weights per row d, for a total of:

$$\sum_{d=1}^{D} dO + 1 \tag{1.57}$$

non-zero weights. In terms of implementation, the 0-embedding strategy from the Total Order Knothe-Rosenblatt mapping still applies, as long as the complete index set is constructed as $\mathscr{J}_D^{KRSV}$.

Ensuring Diffeomorphism Properties of Parameterized Maps

For any $\tilde{S} \in \mathscr{D}_+$ parameterized as in Section 1.3.3

$$\tilde{S}_K(x) = W\Phi(x) \tag{1.58}$$

We must ensure that $WJ_\Phi(x)$ is positive definite for all $\mathbf{x} \in \mathsf{W}$. Here we will define an additional optimization problem to ensure this property. We begin with the Euclidean Projection or the Proximal Operator of the indicator function of $\mathscr{D}_+$.

$$S_W(x) = \underset{m(x)=W\Phi(\mathbf{x}):J_\Phi(\mathbf{x})\geq 0}{\arg\min} ||m(x) - W\Phi(\mathbf{x})||^2 \tag{1.59}$$

As such, S_W retains the properties of a diffeomorphism.

Inverse Map Details

Computing the inverse map also becomes straightforward given the above methodology of representing B and Φ_i

We begin by first showing the Knothe-Rosenblatt property of the map in the complete forward-map equation assuming we are using our polynomial basis representation for a given X_i:

$$\underbrace{\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1(K_1)} & \cdots & 0 & 0 & 0 \\ b_{21} & b_{22} & \cdots & \cdots & b_{2(K_2)} & \cdots & 0 & 0 \\ \vdots & & & & & & & \\ b_{D1} & b_{D2} & \cdots & \cdots & \cdots & \cdots & \cdots & b_{D(K_D)} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} | \\ \Phi(X_i^1) \\ | \\ | \\ \Phi(X_i^1, X_i^2) \\ | \\ \vdots \\ | \\ \Phi(X_i^1, \ldots, X_i^D) \\ | \end{bmatrix}}_{\Phi_i}$$

$$= \begin{bmatrix} S(X_i^1) \\ S(X_i^2) \\ \vdots \\ S(X_i^D) \end{bmatrix} \tag{1.60}$$

where X_i^d represents the d^{th} component of the i^{th} sample.

Here, to fulfill our KR assumption, we assume that Φ_i is a column vector of the polynomial bases evaluated at X_i, ordered according to how many components of X_i the bases are a function of. I.e., if $K_d = |\mathscr{J}_d^{KR}|$, then $\Phi(X_i^1)$ are the first K_1 basis functions that are only a function of X_1, $\Phi(X_i^1, X_i^2)$ are the $K_2 - K_1$ basis functions that are only a function of X_1 and X_2,

and so on. As such, as only the first K_d elements of every d^{th} row of B are (potentially) non-zero, the map should have the appropriate Knothe-Rosenblatt structure by construction.

In the case where we want to invert a sample $S(X_i)$, this defines a system of equations that can be solved row by row for each component of the solution, $S(X_i^d)$, in the form of a polynomial root-finding problem for each row. For example, we first solve for X_i^1, the solution of which we can call X_i^{1*} by finding the (single variable) root of:

$$\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1(K_1)} \end{bmatrix} \begin{bmatrix} | \\ \Phi(X_i^1) \\ | \end{bmatrix} = S(X_i^1) \tag{1.61}$$

Subsequently, we can solve for X_i^2 plugging X_i^{1*} into the second equation:

$$\begin{bmatrix} b_{21} & b_{22} & \dots & \dots & b_{2(K_2)} \end{bmatrix} \begin{bmatrix} | \\ \Phi(X_i^{1*}) \\ | \\ | \\ \Phi(X_i^{1*}, X_i^2) \\ | \end{bmatrix} = S(X_i^2) \tag{1.62}$$

and so on. Note that this results in D-many single variable root-finding problems per sample to invert, and the order of the polynomial that must be solved for will be equal to the order of the polynomial chosen to represent the basis.

References

(1) Andrieu, C.; De Freitas, N.; Doucet, A.; Jordan, M. I. *Mach learn* **2003**, *50*.

(2) Arjovsky, M.; Chintala, S.; Bottou, L. *arXiv preprint arXiv:1701.07875* **2017**.

(3) Azadi, S.; Sra, S *Proceedings of The 31st International Conference on ...* **2014**, *32*.

(4) Bagnoli, M.; Bergstrom, T. *Economic theory* **2005**, *26*, 445–469.

(5) Bakry, D.; Émery, M. In *Séminaire de Probabilités XIX 1983/84*; Springer: 1985, pp 177–206.

(6) Benamou, J.-d.; Carlier, G.; Laborde, M.; Benamou, J.-d.; Carlier, G. **2015**.

(7) Benamou, J.-D.; Carlier, G.; Cuturi, M.; Nenna, L.; Peyré, G. *SIAM Journal on Scientific Computing* **2015**, *37*, A1111–A1138.

(8) Bernardo, J. M.; Smith, A. F. Bayesian theory., 2001.

(9) Bobkov, S.; Madiman, M. *The Annals of Probability* **2011**.

(10) Bonnotte, N. *SIAM Journal on Mathematical Analysis* **2013**, *45*, 64–87.

(11) Boyd, S.; Vandenberghe, L., *Convex optimization*; Cambridge University Press: 2004.

(12) Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. *Foundations and Trends in Machine Learning* **2011**.

(13) Brenier, Y. *CR Acad. Sci. Paris Sér. I Math.* **1987**, *305*, 805–808.

(14) Claici, S.; Chien, E.; Solomon, J. *arXiv preprint arXiv:1802.05757* **2018**.

(15) Cuturi, M.; Doucet, A. In *International Conference on Machine Learning*, 2014, pp 685–693.

(16) El Moselhy, T. A.; Marzouk, Y. M. *Journal of Computational Physics* **2012**, *231*, 7815–7850.

(17) Gelman, A.; Carlin, J. B.; Stern, H. S.; Dunson, D. B.; Vehtari, A.; Rubin, D. B., *Bayesian data analysis*; CRC press Boca Raton, FL: 2014; Vol. 2.

(18) Geman, S.; Geman, D. *IEEE Trans Pattern Anal Mach Intell* **1984**, 721–741.

(19) Genevay, A.; Peyré, G.; Cuturi, M. *arXiv preprint arXiv:1706.01807* **2017**.

(20) Genevay, A.; Cuturi, M.; Peyré, G.; Bach, F. In *Advances in Neural Information Processing Systems*, 2016, pp 3440–3448.

(21) Gilbert, A. C.; Zhang, Y.; Lee, K.; Zhang, Y.; Lee, H. *arXiv preprint arXiv:1705.08664* **2017**.

(22) Gilks, W. R., *Markov chain monte carlo*; Wiley Online Library: 2005.

(23) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. In *Advances in neural information processing systems*, 2014, pp 2672–2680.

(24) Hans, C. *Biometrika* **2009**, *96*, 835–845.

(25) Harrison, D.; Rubinfeld, D. L. *Journal of environmental economics and management* **1978**, *5*, 81–102.

(26) Hastings, W. K. *Biometrika* **1970**, *57*, 97–109.

(27) Jordan, R.; Kinderlehrer, D.; Otto, F. *Research Report No. 96-NA-011* **1996**.

(28) Jordan, R.; Kinderlehrer, D.; Otto, F. *SIAM journal on mathematical analysis* **1998**, *29*, 1–17.

(29) Jordan, R.; Kinderlehrer, D.; Otto, F. *SIAM Journal on Mathematical Analysis* **1998**, *29*, 1–17.

(30) Kim, S.; Ma, R.; Mesa, D.; Coleman, T. P. In *ISIT*, 2013.

(31) Kim, S.; Mesa, D.; Ma, R.; Coleman, T. P. *arXiv preprint arXiv:1509.08582* **2015**.

(32) Kingma, D. P.; Welling, M. *arXiv preprint arXiv:1312.6114* **2013**.

(33) Larochelle, H.; Murray, I. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp 29–37.

(34) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.

(35) Li, Y.; Swersky, K.; Zemel, R. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp 1718–1727.

(36) Lipton, Z. C.; Tripathi, S. *arXiv preprint arXiv:1702.04782* **2017**.

(37) Liu, J. S., *Monte Carlo Strategies in Scientific Computing*; Springer: 2008.

(38) Ma, R.; Coleman, T. In *Allerton*, 2011.

(39) Ma, R.; T.P., C. In *IEEE ISIT*, 2014.

(40) Marzouk, Y. M.; Moselhy, T.; Parno, M.; Spantini, A. **2016**, 1–33.

(41) Mesa, D.; Kim, S.; Coleman, T. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, 2015, pp 676–680.

(42) Papadakis, N.; Peyré, G.; Oudet, E. *SIAM Journal on Imaging Sciences* **2014**, *7*, 212–238.

(43) Park, T.; Casella, G. *Journal of the American Statistical Association* **2008**, *103*, 681–686.

(44) Parno, M.; Marzouk, Y. M. *ArXiv* **2014**, 1–48.

(45) Parno, M.; Moselhy, T.; Marzouk, Y. M. *SIAM/ASA Journal on Uncertainty Quantification* **2016**, *4*, 1160–1190.

(46) Rezende, D. J.; Mohamed, S. *Proceedings of the 32nd International Conference on Machine Learning* **2015**, *37*, 1530–1538.

(47) Robert, C. P.; Casella, G., *Monte Carlo statistical methods*; Citeseer: 2004; Vol. 319.

(48) Salimans, T.; Zhang, H.; Radford, A.; Metaxas, D. *arXiv preprint arXiv:1803.05573* **2018**.

(49) Santambrogio, F. *Birkäuser, NY* **2015**, 99–102.

(50) Schoutens, W., *Stochastic processes and orthogonal polynomials*; Springer Verlag: 2000; Vol. 146.

(51) Sivia, D.; Skilling, J., *Data analysis: a Bayesian tutorial*; OUP Oxford: 2006.

(52) Spantini, A.; Bigoni, D.; Marzouk, Y. M. **2016**, *02139*, 1–8.

(53) Srivastava, S.; Li, C.; Dunson, D. B. *arXiv preprint arXiv:1508.05880* **2015**.

(54) Tantiongloc, J.; Mesa, D.; Ma, R.; Kim, S.; Alzate, C. H.; Camacho, J. J.; Manian, V.; Coleman, T. P. *Proceedings of the IEEE* **2017**, *102*, 273–285.

(55) Tolstikhin, I.; Bousquet, O.; Gelly, S.; Schoelkopf, B. *arXiv preprint arXiv:1711.01558* **2017**.

(56) Villani, C., *Topics in Optimal Transportation*; AMS: 2003.

(57) Villani, C., *Optimal transport: old and new*; Springer: 2008; Vol. 338.

(58) Zhong, W.; Kwok, J. *Journal of Machine Learning Research* **2014**, *32*, 46–54.

(59) Zou, H.; Hastie, T.; Tibshirani, R. *The Annals of Statistics* **2007**, *35*, 2173–2192.

Chapter 2

Bayesian Lasso Posterior Sampling via Parallelized Measure Transport

2.1 Introduction

A quintessential formulation for sparse approximation is Tibshirani's Lasso, which simultaneously induces shrinkage and sparsity in the estimation of regression coefficients [30]. The formulation of the standard Lasso is as follows:

$$x^* = \underset{x \in \mathbb{R}^d}{\arg\min} \, ||y - \Phi x||_2^2 + \lambda ||x||_1 \tag{2.1}$$

where $y \in \mathbb{R}^n$ is a vector of responses, Φ is a $n \times d$ matrix of standardized regressors, and $x \in \mathbb{R}^d$ is the vector of regressor coefficients to be estimated.

It is known that the Lasso can be interpreted as a Bayesian posterior mode estimate with a Laplacian prior [30]. Imposing a Laplacian prior is equivalent to L_1-regularization, which has desirable properties, including robustness and logarithmic sample complexity [22]. Various algorithms for solving (2.1) are typically employed, including iterative soft-thresholding and its successors [6], [1], [11]. These methods are scalable, yet they only provide the maximum a posteriori (MAP) estimate, a point estimate. With i.i.d. samples $Z_1, \ldots, Z_K$ from the posterior distribution, for any set of possible decisions $\mathscr{D}$, and any loss function $l : \mathbb{R}^d \times \mathscr{D} \to \mathbb{R}$, the Bayes optimal decision, $d^*(y)$, can be approximately found by minimizing the empirical conditional

expectation:

$$d^*(y) = \arg\min_{d \in \mathscr{D}} \mathbb{E}[l(X,d)|Y=y] \simeq \arg\min_{d \in \mathscr{D}} \frac{1}{N} \sum_{k=1}^{K} l(Z_k, d) \tag{2.2}$$

Previous approaches have been developed [23, 13] to sample from the posterior distribution corresponding to the Lasso problem based on Markov Chain Monte Carlo methods (MCMC). However these methods necessarily introduce correlations between the generated samples, are sequential in nature, and do not often scale well with dataset size or model complexity [14],[21], [17].

We here consider a framework to generate i.i.d. samples $Z_1, \ldots, Z_K$ from the posterior distribution associated with the Lasso through a measure transport approach. We show a formulation for obtaining a transport map that transforms samples from the Laplacian prior to samples from the posterior distribution. We exploit previous results casting Bayesian inference as a measure transport problem [8], [19] and the Bayesian Lasso posteriors log-concavity to represent such a transport map with polynomial chaos and perform a relative entropy minimization, which results in a convex optimization problem amenable to parallelization [16], [20]. We further show that finding the optimal map that transforms prior Laplacian samples to posterior samples can be found with off-the-shelf Lasso solvers and closed-form linear algebra updates.

2.1.1 Relevant Work

Park and Casella proposed a Gibbs sampler for the Bayesian Lasso problem, based on a hierarchical formulation of the prior structure[23], where the Gauss-scale mixture property of the Laplacian prior distribution is exploited to formulate a fully Bayesian Lasso inference procedure, where latent scale variables also have a prior distribution. This structure leads to a tractable three-step Gibbs sampler that can be used to draw approximate samples from the posterior and construct credibility intervals. Hans [13] obviated the need for hyper-parameters and used a direct characterization of the posterior distribution to develop a Gibbs sampler to

generate posterior samples. As an MCMC algorithm, the Gibbs sampler generates a Markov chain of samples, each of which is correlated with its previous sample. The correlation between these samples can decay slowly and lead to burn-in periods where samples have to be discarded [26]. Although theoretical upper bounds on the convergence of Gibbs samplers have been proved [25], these guarantees are weaker in the case of Bayesian Lasso. [24] developed a two-step Gibbs sampler for the Bayesian Lasso with improved convergence behavior. However, a way to derive i.i.d. samples from the Bayesian Lasso posterior without burn-in periods has remained elusive.

Moshely et al. first proposed an alternative method for directly sampling from the posterior distribution based on a measure transport approach [8], [19], where a mapping is developed to transform samples from one distribution to another, using a polynomial chaos expansion [10]. Bayesian inference can be cast as a special case of this, where the original distribution is the prior (which in many cases is easy to sample from) and the target distribution is the posterior. Recently, Kim et al. further investigated the Bayesian transport sampling problem and showed that when the prior and likelihood satisfy a log-concavity property, the relative entropy minimization approach to find a transport map is a convex optimization problem [16]. Mesa et al. introduced an Alternating Direction Method of Multipliers (ADMM) reformulation and showed that minimization can be performed by solving a series of convex optimization problems in parallel [20]. Wang et al. used a measure transport approach to extend the randomize-then-optimize MCMC approach to sample from posteriors with $L1$ priors by transforming the $L1$ prior distribution to a Gaussian distribution [32].

2.1.2 Our Contribution

We present a technique to sample from the Bayesian Lasso posterior based on a measure transport approach [8], [16]. The formulation is conceptually different from the Gibbs sampler methodology, as the target computation are not samples from the posterior, but rather a transport map that once computed for a certain dataset can continually be used to generate an arbitrary number of posterior samples. We show that this transport map can be computed in a parallel

fashion based on an Alternating Direction Method of Multipliers (ADMM) formulation as in [20]. Furthermore, our solution only requires off-the-shelf Lasso solvers and linear algebra updates. Once the transport map is computed, we need only draw i.i.d. samples from the (Laplacian) prior and transform them with the transport map into i.i.d. samples from the posterior.

We exploit the ability to draw i.i.d. samples from the posterior to develop an Expectation Maximization algorithm for maximum likelihood estimation of the Bayesian Lasso parameter. Additionally, we compare our results to a traditional Bayesian Lasso Gibbs sampler and show that we achieve similar results when analyzing the diabetes dataset presented in [7]. We then show empirically in simulation that posterior sampling convergence to the Bayes risk with our sampling method is superior to doing so with Gibbs sampling. Finally we show that our parallel framework is amenable to implementation in GPU systems and architectures that leverage parallelization. We provide an example of our Bayesian Lasso framework implemented in a GPU.

The rest of the paper is organized as follows: in Section II, we provide some preliminaries and definitions. In Section III, we introduce a relative entropy minimization formulation for Bayesian Lasso posterior sampling via measure transport, and how this can be performed with ADMM methods from convex optimization formulation. We then show how this formulation can be reduced to solving a collection of Lasso problems in parallel and performing closed-form linear algebra updates. In Section IV, we derive an expectation maximization algorithm to find the regularization parameter associated with the Lasso. In Section V, we compare our Bayesian Lasso framework to a traditional Gibbs sampler Bayesian Lasso through the analysis of the diabetes dataset from Efron et al [7]. We also show empirical convergence results for our measure transport sampler and the Gibbs sample in attaining the Bayes Risk. In Section VI, we present a framework for a GPU implementation. In Section VII, we conclude and discuss future potential directions.

2.2 Definitions

We consider the following generative model of how a latent and sparse $x \in \mathbb{R}^d$ relates to a measurement $y \in \mathbb{R}^n$:

$$y = \Phi x + \varepsilon \tag{2.3}$$

and the measurement noise satisfies $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$.

We assume an i.i.d. Laplacian statistical model on x. Therefore, the following Bayesian Lasso regression model is specified as

$$p(y|x;\sigma^2) = \mathcal{N}(y|\Phi x, \sigma^2 \boldsymbol{I}_n) \tag{2.4}$$

$$p(x;\tau) = \prod_{i=1}^{d} \frac{\tau}{2} e^{-\tau|x_i|} \tag{2.5}$$

where $\mathcal{N}(t|m,S)$ represents the density function, evaluated at t, of a multivariate normal random variable with expectation m and covariance matrix S. Throughout the paper, we assume that Φ, τ, and σ are fixed and non-random. We note that the negative log posterior density satisfies

$$-\log p(x|y;\sigma^2,\tau) \propto \frac{1}{2\sigma^2}\|y-\Phi x\|_2^2 + \tau||x||_1 \tag{2.6}$$

As such, the standard Lasso problem for a given $\lambda \equiv 2\tau\sigma^2$

$$x^* = \underset{x\in\mathbb{R}^d}{\arg\min}\, ||y-\Phi x||_2^2 + \lambda||x||_1 \tag{2.7}$$

is a maximum a posteriori estimation problem for the Laplacian prior in (2.5).

In the model above and throughout our methodology, we assume that the parameter σ^2 is fixed and known, deviating from the results from the Bayesian Lasso Gibbs sampler first presented in [23], for which σ^2 is imparted with a prior. As such, we are considering the posterior

distribution associated with the original Bayesian interpretation to Lasso, for which the solution to (2.1) is the MAP estimate.

Park and Casella [23] constructed a hierarchical model to facilitate implementation of a Gibbs sampler for the Bayesian Lasso. The Gibbs sampler exploits that the double exponential (Laplace) distribution can be represented as a scale mixture of normals. (2.5) is replaced by the following prior:

$$\begin{aligned} p(\boldsymbol{x}|t_1^2,...,t_d^2) &= \mathcal{N}_d(\mathbf{0},D_t) \quad D_t = \text{diag}(t_1^2,...,t_d^2) \\ p(t_1^2,...,t_d^2;\tau) &= \prod_{j=1}^{d} \frac{\tau^2}{2} e^{-\tau^2 t_j^2/2} dt_j^2 \end{aligned} \tag{2.8}$$

[23] additionally extend the Bayesian Lasso regression model to account for uncertainty in the hyperparameters by placing a prior on σ^2 leading to the following hierarchical representation of the posterior:

$$\begin{aligned} p(\boldsymbol{x},\sigma^2|y;\tau) \propto \\ \pi(\sigma^2)(\sigma^2)^{-(n-1)/2} \exp -\tfrac{1}{2\sigma^2}(y-\Phi\boldsymbol{x})^T(y-\Phi\boldsymbol{x}) - \tau\textstyle\sum_{j=1}^{d}|x_j| \end{aligned} \tag{2.9}$$

with prior

$$\begin{aligned} p(\boldsymbol{x}|t_1^2,...,t_d^2,\sigma^2) &= \mathcal{N}_d(\mathbf{0},\sigma^2 D_t) \quad D_t = \text{diag}(t_1^2,...,t_d^2) \\ p(t_1^2,...,t_d^2;\tau) &= \prod_{j=1}^{d} \frac{\tau^2}{2} e^{-\tau^2 t_j^2/2} dt_j^2 \\ p(\sigma^2) &= \pi(\sigma^2)d\sigma^2 \end{aligned} \tag{2.10}$$

2.3 Bayesian Lasso via Measure Transport

In this section we provide background on measure transport theory and show that we can find a transport map that pushes samples from the prior distribution in (2.5) to the Bayesian Lasso posterior. We utilize the ADMM framework introduced in [20] and develop a distributed Bayesian Lasso solver. Furthermore, we show that Bayesian Lasso can be formulated as a batch of Lasso problems, which themselves can be solved with existing sparse approximation algorithms in a parallel manner.

2.3.1 Fully Bayesian Inference via Measure Transport

As an alternative to the sampling approaches described above, we consider finding transformations, or transport maps, between probability measures. Finding a mapping between two measures is the central problem in Optimal Transport Theory, a rich field with a wide variety of applications [31]. We seek a transport map S that transforms the prior distribution p to the posterior distribution q.

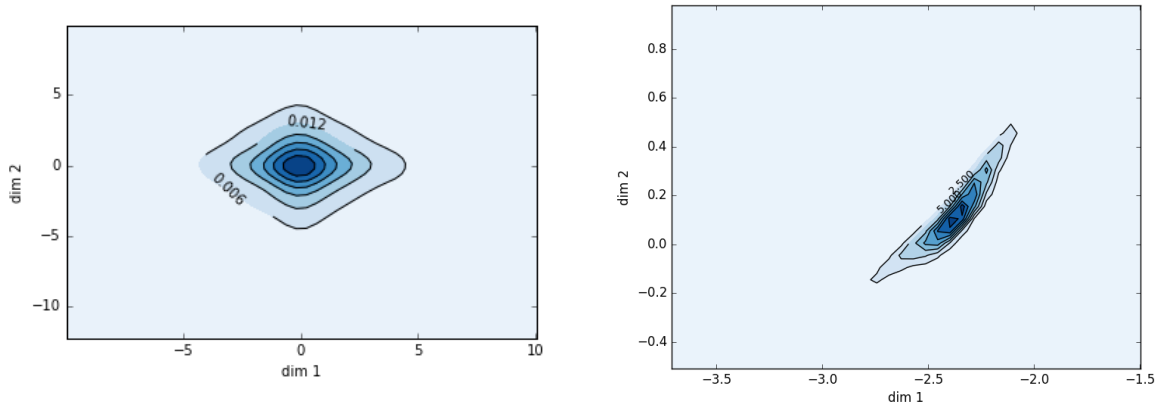
We restrict our search for S to the set of diffeomorphisms with positive-definite Jacobian:

$$\mathscr{D}_+ \triangleq \left\{ S : \mathbb{R}^d \to \mathbb{R}^d,\ J_S \succ 0 \right\}.$$

If an $S \in \mathscr{D}_+$ satisfies

$$p(u) = q(S(u)) \det(J_S(u)) \quad \text{for all } u \in \mathbb{R}^d \tag{2.11}$$

, then S is said to *push* p to q, i.e. it transforms a sample W from p into a sample $Z = S(W)$ from q. We note that we can always find a transport map in $\mathscr{D}_+$ that will satisfy (2.11) [4]. Given such a map we can sample from the posterior distribution by mapping i.i.d. samples from the prior $X_1, ..., X_K$ to i.i.d. samples from the posterior $S(X_1), ..., S(X_k)$. Figure 2.1 shows the effect of a transport map on samples from the Laplacian prior in (2.5).



(a) **(b)**

Figure 2.1. Effect of transport map S on prior samples (a) kernel density estimate of prior (Laplacian) distribution constructed by samples (b) kernel density estimate of samples transformed through transport map S; posterior density

2.3.2 A Convex Optimization Formulation

Recent work [16, 15] has shown that for the problems where the prior and likelihood are log-concave, developing a map that transforms i.i.d. samples from the prior into i.i.d. samples from the posterior can be performed with convex optimization.

Given an arbitrary $S \in \mathscr{D}_+$, then there will be an induced $\tilde{P}_S$ for which S pushes $\tilde{P}_S$ to q. That is:

$$\tilde{P}_S(u) = q(S(u)) \det(J_S(u)) \quad \text{for all } u \in \mathbb{R}^d \tag{2.12}$$

From this perspective, we can cast the transport problem as finding the transport map S^* that minimizes a distance between an induced $\tilde{P}$ and the true p. We use the Kullback-Leibler Divergence and arrive at the following optimization problem:

$$S^* = \underset{S \in \mathscr{D}_+}{\arg\min} D\big(P \| \tilde{P}_S\big) \tag{2.13}$$

By defining

$$g(z) \triangleq -[\log f_{Y|X}(y|z) + \log f_X(z)] \tag{2.14}$$

where the densities refer to the likelihood and prior, (2.13) becomes

$$S^* = \underset{S \in \mathscr{D}_+}{\arg\max}\, \mathbb{E}_P \left[-g(S(X)) + \log\det\left(J_{S(X)}\right) \right] \tag{2.15}$$

Moreover, when q is log-concave (equivalently when g is convex), this (infinite-dimensional) optimization problem is convex.

Parametrization of the Transport Map:

In order to solve (2.15), we parametrize the problem to arrive at a finite-dimensional convex optimization problem. We approximate any $S \in \mathscr{D}_+$ as a linear combination of basis functions through a Polynomial Chaos Expansion (PCE) [33], [9] where ϕ are the polynomials orthogonal with respect to the prior p:

$$S(x) = \sum_{j \in \mathscr{J}} b_j \phi^{(j)}(x) \tag{2.16}$$

$$\int_{x \in \mathscr{X}} \phi^{(i)}(x)\phi^{(j)}(x)p(x)dx = \delta_{i,j} \tag{2.17}$$

with $\delta_{i,j}$ being 1 if $i = j$ and 0 otherwise. Now define $K = |\,\mathscr{J}\,|$ and we have that for $\mathscr{X} \subset \mathbb{R}$:

$$\begin{aligned}
F &= [b_1, \ldots, b_K], && d \times K && (2.18)\\
A(x) &= [\phi^{(1)}(x), \ldots, \phi^{(K)}(x)]^T, && K \times 1 && (2.19)\\
S(x) &= FA(x), && d \times 1 && (2.20)\\
J(x) &= \left[\frac{\partial \phi^{(i)}}{\partial x_j}(x) \right]_{i,j}, && K \times d && (2.21)\\
J_S(x) &= FJ(x) && d \times d. && (2.22)
\end{aligned}$$

We can then approximate the expectation from (2.15) using an empirical expectation based

upon i.i.d. samples from p. Letting $A_i \triangleq A(X_i)$ and $J_i \triangleq J(X_i)$, we arrive at the following finite-dimensional problem:

$$F^* = \underset{F:FJ_i \succ 0}{\arg\max} \frac{1}{N} \sum_{i=1}^{N} -g(FA_i) + \log\det(FJ_i) \tag{2.23}$$

Whenever q is log-concave (equivalently g is convex), this is a finite-dimensional convex optimization problem. Moreover, as $K \to \infty$, from the PCE theory, the map $F^*A(x)$ converges to the optimal map S^* that pushes p to q.

2.3.3 Parallelized Convex Solver with ADMM

More recently, [20] demonstrated a scalable framework to solve (2.23) which only requires iterative linear algebra updates and solving, in parallel, a number of quadratically regularized point estimation problems. The distributed architecture involves an augmented Lagrangian and a concensus Alternating Direction Method of Multipliers (ADMM) formulation:

$$\begin{aligned}
\min_{F,Z,p,B} \quad & \tfrac{1}{N}\sum_{i=1}^{N} g(p_i) - \log\det Z_i + \quad \frac{1}{2}\rho\|F_i - B\|_2^2 \\
& + \tfrac{1}{N}\sum_{i=1}^{N} \tfrac{1}{2}\rho\|BA_i - p_i\|_2^2 + \quad \frac{1}{2}\rho\|BJ_i - Z_i\|_2^2 \\
\text{s.t.} \quad & BA_i = p_i : \qquad \gamma_i \quad (d \times 1) \\
& BJ_i = Z_i : \qquad \beta_i \quad (d \times d) \\
& F_i - B = 0 : \qquad \alpha_i \quad (d \times K) \\
& Z_i \succ 0
\end{aligned}$$

for any fixed $\rho > 0$.

A penalized Lagrangian is solved iteratively by first solving for B^{k+1}

$$B^{k+1} = \frac{1}{N}\sum_{i=1}^{N}\left[\rho\left(F_i^k + p_i^k A_i^T + Z_i^k J_i^T\right)\right.$$
$$\left. + \gamma_i^k A_i^T + \beta_i^k J_i^T + \alpha_i^k\right]\mathscr{M}, \tag{2.24}$$
$$\mathscr{M} \triangleq \left[\rho\left(I + \frac{1}{N}\sum_{i=1}^{N} A_i A_i^T + J_i J_i^T\right)\right]^{-1} \tag{2.25}$$

and then solving, in parallel for $1 \leq i \leq N$, the other variable updates:

$$F_i^{k+1} = -\frac{1}{\rho}\alpha_i^k + B^{k+1} \tag{2.26a}$$
$$Z_i^{k+1} = Q\tilde{Z}_i Q^T \tag{2.26b}$$
$$p_i^{k+1} = \arg\min_{p_i} g(p_i) + \frac{1}{2}\rho\|B^{k+1}A_i - p_i\|_2^2$$
$$+ \gamma_i^{kT}(p_i - B^{k+1}A_i) \tag{2.26c}$$
$$\gamma_i^{k+1} = \gamma_i^k + \rho(p_i^{k+1} - B^{k+1}A_i) \tag{2.26d}$$
$$\beta_i^{k+1} = \beta_i^k + \rho(Z_i^{k+1} - B^{k+1}J_i) \tag{2.26e}$$
$$\alpha_i^{k+1} = \alpha_i^k + \rho(F_i^{k+1} - B^{k+1}) \tag{2.26f}$$

ADMM guarantees convergence to the optimal solution [3]. To emphasize, each *i*th update in (2.26) can be solved in parallel. As (2.26b) is an eigenvalue-eigenvector decomposition (details can be found in [20]), it follows that all the updates involve linear algebra with the exception of (2.26c), which is a quadratically regularized point estimation problem.

2.3.4 Efficiently Solving the Bayesian Lasso

We exploit the unique problem structure of Bayesian Lasso to simplify a scalable implementation.

Lemma 2.3.1. *The PCE for the Laplacian distribution is* $\phi_L(x) = \phi_E(|x|)$ *where* ϕ_E *are the*

Laguerre polynomials.

Proof.

$$\begin{aligned}\int_{-\infty}^{\infty} \phi_E^i(|x|)\phi_E^j(|x|)p_L(x)dx &= \int_{-\infty}^{\infty} \phi_E^i(|x|)\phi_E^j(|x|)\frac{1}{2}p_E(|x|)dx \\ &= 2\int_0^{\infty} \phi_E^i(x)\phi_E^j(x)\frac{1}{2}p_E(x)dx \\ &= \delta_{i,j} \end{aligned} \tag{2.27}$$

Where the first equality holds because the Laplacian density $p_L(x)$ is related to the exponential density $p_E(x)$ by $p_L(x) = \frac{1}{2}p_E(|x|)$, the second equality holds by symmetry of the function being integrated, and the third follows because the PCE for the exponential distribution is obtained with the Laguerre polynomials $\phi_E^{(j)}$ [33]. □

We now show that for Bayesian Lasso, the only ADMM update that is not linear algebra is simply a Lasso problem.

Theorem 2.3.2. *For the Bayesian Lasso statistical model given by* (2.6)*, the ADMM update* (2.26c) *is a d-dimensional Lasso point estimation problem:*

$$p_i^{k+1} = \arg\min_{p_i} ||\hat{y} - \hat{\Phi}^T p_i||_2^2 + \lambda ||p_i||_1 \tag{2.28}$$

where $\hat{\Phi}$ *and* $\hat{y}$ *satisfy*

$$\begin{aligned}\hat{\Phi}^T\hat{\Phi} &= \Phi^T\Phi + \frac{1}{2}\rho I \\ \hat{y} &= \left(\left[y^T\Phi + \frac{1}{2}\rho(B^{k+1}A_i)^T - \frac{1}{2}\gamma_i^{kT}\right]\hat{\Phi}^+\right)^T\end{aligned} \tag{2.29}$$

and $\hat{\Phi}^+$ *represents the pseudo-inverse.*

Proof. Dropping indices of (2.26c), becomes

$$\begin{aligned} p^* &= \arg\min_p \ \text{quad}(p) + \lambda ||p||_1, \\ \text{quad}(p) &\triangleq p^T(\Phi^T\Phi + \frac{1}{2}\rho I)p + (\gamma^T - 2y^T\Phi - \rho(BA)^T)p. \\ &= p^T\hat{\Phi}^T\hat{\Phi}p + (\gamma^T - 2y^T\Phi - \rho(BA)^T)p \end{aligned} \tag{2.30}$$

where (2.30) follows from performing a Cholesky decomposition to build a unique $\tilde{\Phi} \in \mathbb{R}^{d\times d}$ and then zero padding to build $\hat{\Phi} \in \mathbb{R}^{n\times d}$, obeying the relationship given in (2.29). Then we complete the square in order to get an equation of the form $\|\hat{\Phi}p\|_2^2 - 2\hat{y}^T\hat{\Phi}p + \|\hat{y}\|_2^2 = \|\hat{y} - \hat{\Phi}p\|_2^2$:

$$-2\hat{y}^T\hat{\Phi}p = (\gamma^T - 2y^T\Phi - \rho(BA)^T)p.$$

□

Remark 6. *The problem of finding a map S^* to generate i.i.d. samples from the Bayesian Lasso posterior can be solved iteratively. Each step involves solving – in parallel – linear algebra problems and d-dimensional Lasso problems* (2.1).

The procedure for Bayesian Lasso via measure transport is outlined in Algorithm 1.

2.4 Choosing λ via Maximum Likelihood Estimation

The parameter of the standard Lasso in (2.1), λ, can be chosen by cross-validation, generalized cross-validation, and ideas based on unbiased risk minimization [30]. Park and Casella used Empirical Bayes Gibbs Sampling [5] to find a maximum likelihood estimate of λ via an Expectation Maximization (EM) algorithm [23]. This empirical scheme, however, is specific to the Gibbs sampler and the hierarchical model introduced in [5]. Here, we propose an Expectation Maximization algorithm to calculate a marginal Maximum Likelihood Estimate of λ.

Algorithm 1: Distributed Bayesian Lasso

function BayesianLasso ($\mathbf{x}_1,...,\mathbf{x}_N \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^n$, $\Phi \in \mathbb{R}^{n\times d}$, λ, ρ, K);
Input : Samples $\mathbf{x}_1,...,\mathbf{x}_N$ from prior in (2.5)
Output : B^∞ holds coefficients of map S such that $S(x) = B^\infty A(x)$
Construct A_i and J_i via Polynomial Chaos Expansion for $i = 1,...,N$ as in (2.19) and (2.21);
Construct $\mathscr{M}$ as in (2.25);
Initialize B^0 and $F_i^0, Z_i^0, p_i^0, \gamma_i^0, \beta_i^0, \alpha_i^0$ randomly for $i = 1,...,N$;
while B^k *has not converged* **do**
    Update B^{k+1} as in (2.24) ;
    Update in parallel for $i = 1,...,N$ $F_i^{k+1}, Z_i^{k+1}, \gamma_i^{k+1}, \beta_i^{k+1}, \alpha_i^{k+1}$ as in (2.26)
    p_i^{k+1} with a **Lasso solver** as in (2.28) ;
    $k = k+1$
end

In the Expectation Maximization framework, the basic problem is to find an estimate of the parameter λ that maximizes the likelihood function $f(y|\lambda)$ for a given observation y. That is,

$$\hat{\lambda} = \arg\max_{\lambda} \log f(y|\lambda) \tag{2.31}$$

$$= \arg\max_{\lambda} \log \int p(x,y|\lambda)dx \tag{2.32}$$

$$= \arg\max_{\lambda} \log \int g(x)[\frac{p(x,y|\lambda)}{g(x)}dx] \tag{2.33}$$

Where $p(x,y|\lambda)$ represents the joint distribution of X and the observation Y and $g(x)$ is an arbitrary density.

The EM algorithm alternates between an expectation and a maximization step. The "E step" finds a lower bound, a density $g(x)$, that is equal to the log-likelihood function at the current parameter estimate λ_k. The "M step" generates the next estimate λ_{k+1} as the parameter that maximizes this greatest lower bound.

It can be shown that choosing the posterior distribution $g(x) = p(x|y,\lambda_k)$ maximizes the lower bound in the kth E step. Thus the E step involves taking an expectation with respect to the

posterior distribution of the complete-data log likelihood under the current iterate λ^k:

$$d\log(\lambda) - \lambda E_{\lambda^k}[||x||_1|y] + \mathscr{C} \tag{2.34}$$

where $\mathscr{C}$ represents terms not involving λ.

The M step then maximizes (2.34) with respect to λ to get the next iterate λ^{k+1}. The M-step leads to a simple analytical solution.

$$\lambda^{k+1} = \frac{d}{\sum_{i=1}^{d} E_{\lambda^k}[|x_i|\,|y]} \tag{2.35}$$

Since the expectation is taken with respect to the posterior, we can approximate it with i.i.d. posterior samples from our approach.

For our Bayesian Lasso the steps are as follows:

1. Choose an initial $\lambda^{(0)}$
2. Perform Algorithm 1 with $\lambda = \lambda^{(0)}$ to find S and generate N samples from the posterior distribution as $z_j = S(x_j)$ for $j = 1...N$ where x_j is a sample from the prior in (2.5).
3. (E-Step):Approximate the expected complete data log likelihood by substituting averages for the expectation in (2.34)
4. (M-step) Let λ^{k+1} be the value of λ that maximizes the expected log likelihood of the previous step, namely (2.35)
5. Return to step 2 until convergence.

A setback to this algorithm is that one has to compute a transport map at each iteration which may be computationally challenging. One could alternatively approximate λ with the Empirical formulation presented in [23] and set $\lambda^{(0)}$ to this value, decreasing the number of iterations to approximate λ.

2.5 Comparisons to Gibbs Sampling

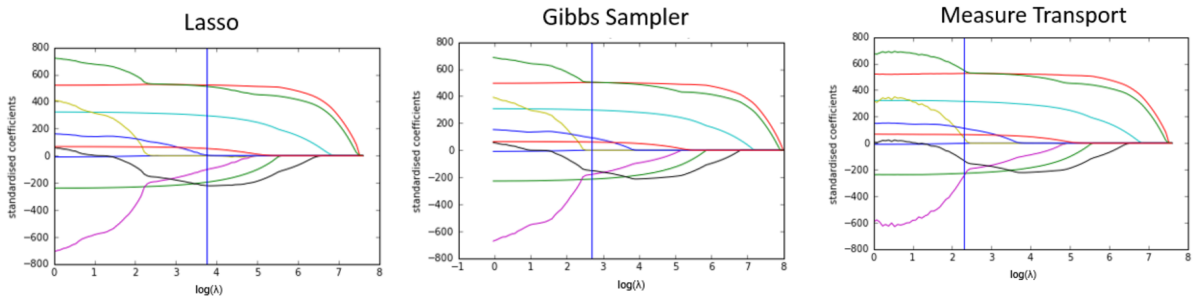
We now present comparisons of our measure transport methodology with a Gibbs sampler for Bayesian Lasso based on the diabetes data of Efron et al [7], which has $d = 10$. We will follow the analysis presented in [23] and compare results with the respective Gibbs sampler. We note that our Bayesian Lasso model differs from that in [23] in that we do not place a prior on σ^2. We compare regression estimates obtained with both methodologies. Then, using a Gibbs sampler that operates in an equivalent model (with prior (2.9)), we show empirically that samples from our methodology outperform samples from a Gibbs sampler in attaining the Bayes Risk.

2.5.1 Analysis on Diabetes Data

We analyze a diabetes data set [7] and compare results when using the Gibbs sampler presented in [23] which utilizes a prior on σ^2 as in (2.10) and when using samples from our transport based methodology. We show that despite our treatment of σ^2 as fixed, we achieve similar results as we capture the complexity of the posterior distribution in this real dataset.

Figure 2.2 compares our measure transport Bayesian Lasso posterior median estimates 2.2c with the ordinary Lasso 2.2a and the Gibbs sampler posterior median estimates 2.2b. We take the vector of posterior medians as the one that minimizes the L_1 norm loss averaged over the posterior. For all three methods, the Lasso and Bayesian Lasso estimates were computed by sweeping over a grid of values for λ. We ran our Bayesian Lasso with a Polynomial Chaos Expansion order of 3, and trained with $N = 500$ prior samples to find a transport map. The specifications for the Gibbs sampler were to use a scale-invariant prior on σ^2 and to run for 10,000 iterations after 1000 iterations of burn-in.

Figure 2.2c shows the resulting optimal λ (depicted with a vertical line) found by the EM algorithm presented in Section 4. The vertical line in Figure 2.2b is the optimal λ found by [23] by running a Monte Carlo EM algorithm corresponding to the particular Gibbs implementation. The vertical line in the Lasso graph 2.2a represents the estimate chosen by n-fold cross validation.



(a) **(b)** **(c)**

Figure 2.2. Comparison of Linear Regression Estimates on Diabetes Data trace plots for estimates of the diabetes data regression parameters for (a) Lasso (b) Gibbs sampler Bayesian Lasso; (c) our measure transport Bayesian Lasso method The vertical line represents the λ estimate.

Despite treating σ^2 as fixed, the L_1 paths are very similar to the Bayesian Lasso imparted with a prior on σ^2. As already noted in previous work, the Bayesian Lasso paths are smoother than the Lasso estimates.

We further compare the 95% credible intervals for the diabetes data obtained with a fixed λ (the optimal λ corresponding to the Gibbs sampler) for the marginal posterior distributions of the Bayesian Lasso estimates. Figure 2.3 shows the corresponding result for the Lasso, Gibbs sampler, and our proposed methodology.

We also show Kernel Density Estimation (KDE) plots (Figure 2.4) of two of the regression variables using both methods. The kernel density estimates are similar in shape and support.

2.5.2 Performance Comparisons

As stated in the introduction, the class of Markov Chain Monte Carlo (MCMC) methods, including Gibbs sampling, are widely used to generate samples from a target distribution. Samples are obtained by iterating through a Markov Chain whose invariant distribution is the the posterior. However, the convergence times and mixing rates of the Markov chain are generally unknown [26]. In practice, one often discards an initial set of samples (burn-in) to avoid biases.

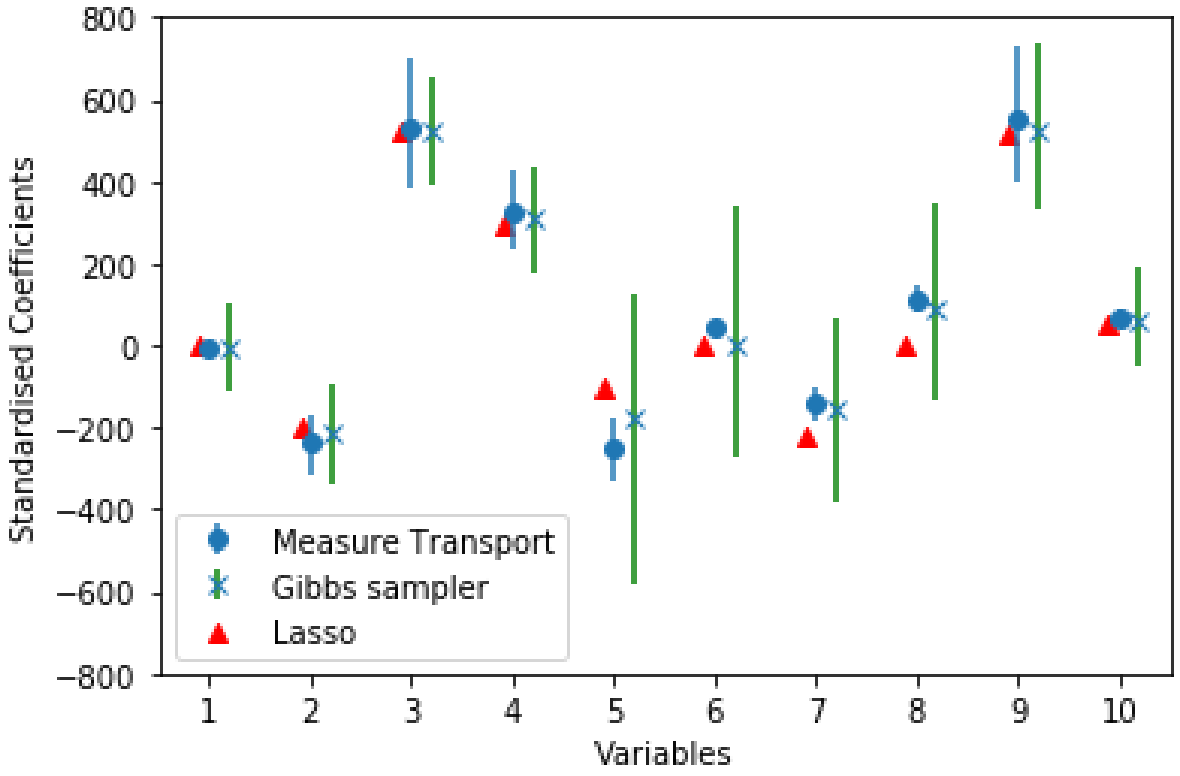
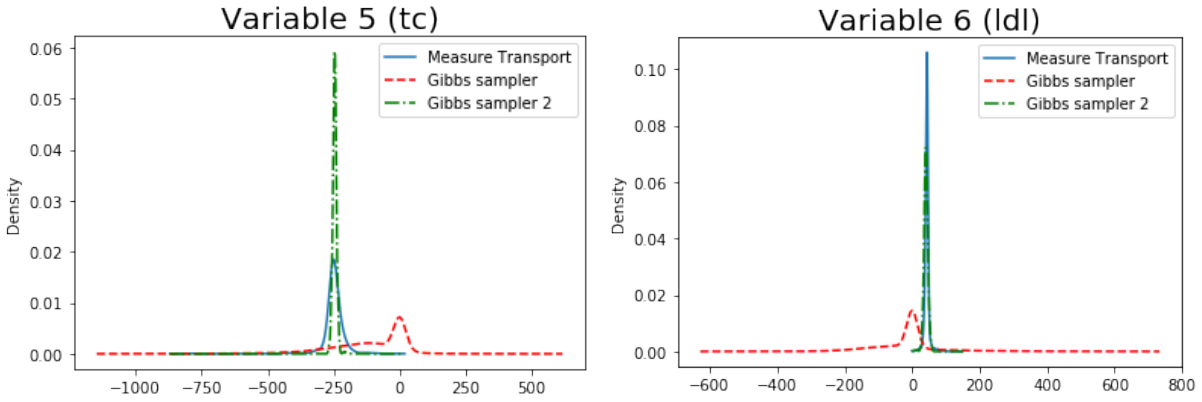


Figure 2.3. Posterior median Bayesian Lasso estimates and corresponding 95 percent credible intervals for a Gibbs sampler and our Measure Transport methodology. Lasso estimates are also shown for comparison.



(a) **(b)**

Figure 2.4. Marginal posterior density estimates for variables 5 and 6 of the Diabetes dataset. Kernel density estimates were constructed using 10,000 samples from a Gibbs sampler or a transport map respectively.

In addition, since adjacent samples are necessarily correlated, the effective sample sizes are reduced when constructing estimators.

A major advantage of our transport-based methodology is that with a good approximation to the transport map we can generate i.i.d. samples from the posterior. However, the accuracy of the approximation to the transport map (and effectively, the accuracy of the samples drawn) , will depend on the set of parameters used to construct such a transport map.

In this section, we formulate a way to compare the performance of samples from a Bayesian Lasso Gibbs sampler utilizing prior (2.9) with that of samples obtained from a Bayesian Lasso transport map utilizing prior (2.5) through empirical risk minimization ideas.

Bayes Risk Comparisons

In a Bayesian setting, a natural question to ask is how well an estimator of the posterior behaves in terms of its risk. Consider a latent random variable $X \in \mathsf{X}$, a measured random variable $Y \in \mathsf{Y}$, and a set of possible decisions $\mathscr{D}$. Let $l : \mathbb{R}^d \times \mathscr{D} \to \mathbb{R}$, be a loss function, so that $l(x,d)$ is the loss incurred when the latent random variable is x and the decision taken is d. The risk is defined as

$$R(P_X, d) = \mathbb{E}[l(X, d(Y))] \tag{2.36}$$

Where the expectation is taken over all (x,y) pairs. We also define the Bayes risk as the minimum possible risk over all possible $d : Y \to \mathscr{D}$

$$R^*(P_X) = \inf_{d \in \mathscr{D}} R(P_X, d) \tag{2.37}$$

As stated in (2.2), the Bayes Optimal decision scheme , $d*$, which attains the Bayes Risk can be approximated with i.i.d. samples from the posterior $Z_1, ..., Z_k$

$$\arg\min_{d \in \mathscr{D}} \mathbb{E}[l(X,d)|Y = y] \simeq \arg\min_{d \in \mathscr{D}} \frac{1}{N} \sum_{k=1}^{K} l(Z_k, d) \tag{2.38}$$

With d^*, for a set of latent variables $X_{1:T}$ inducing observations $Y_{1:T}$, we can approximate (2.37) for Bayesian Lasso with an L_1 loss.

$$R^*(P_X) \simeq \frac{1}{T}\sum_{t=1}^{T} l(X_t, d^*(Y_t)) \tag{2.39}$$

$$= \frac{1}{T}\sum_{t=1}^{T} ||X_t - d^*(Y_t)||_1 \tag{2.40}$$

Here, we formulate a way to compare two posterior sampling schemes, a Gibbs sampler and samples generated from a transport map S by their ability to approximate (2.37).

In Algorithm 1, it is clear that the number of samples N from the prior used to calculate the posterior is a crucial parameter affecting the accuracy of the obtained transport map. Since we can generate as many samples as we wish once we have a transport map, we disadvantage our algorithm by setting the number of training samples from the prior to N and drawing the same number of samples from the posterior. We keep all other parameters fixed using a PCE order of 5. We compare this to drawing N samples from a Gibbs sampler (we do not use burn-in).

Specifically, we generate two sequences of decision making schemes d_N and $\tilde{d}_N$ from i.i.d. samples from our scheme and from samples from a Gibbs sampler respectively and we use these sequences to approximate (2.37) as $R(P_X, d_N)$ and $R(P_X, \tilde{d}_N)$ respectively. As N grows large, both schemes should approach the Bayes Risk. To approximate the expectation as in (2.40), we set T=500 and generate (X_t, Y_t) pairs as in (2.3), where $dim(X) = 3, dim(Y) = 10$, Φ is fixed, and $\varepsilon \sim \mathcal{N}(0,1)$. Using each observation Y_t , and the prior in (2.5), we generate samples from the posterior using both a Gibbs sampler and a transport map computed with N prior samples.

Figure 2.5 shows the Bayes risk approximation attained with a Gibbs sampler for the Bayesian Lasso and that with samples from a transport map. We show that the risk on both methods reaches the same value (the Bayes risk) after N is large enough (to be fixed). Our

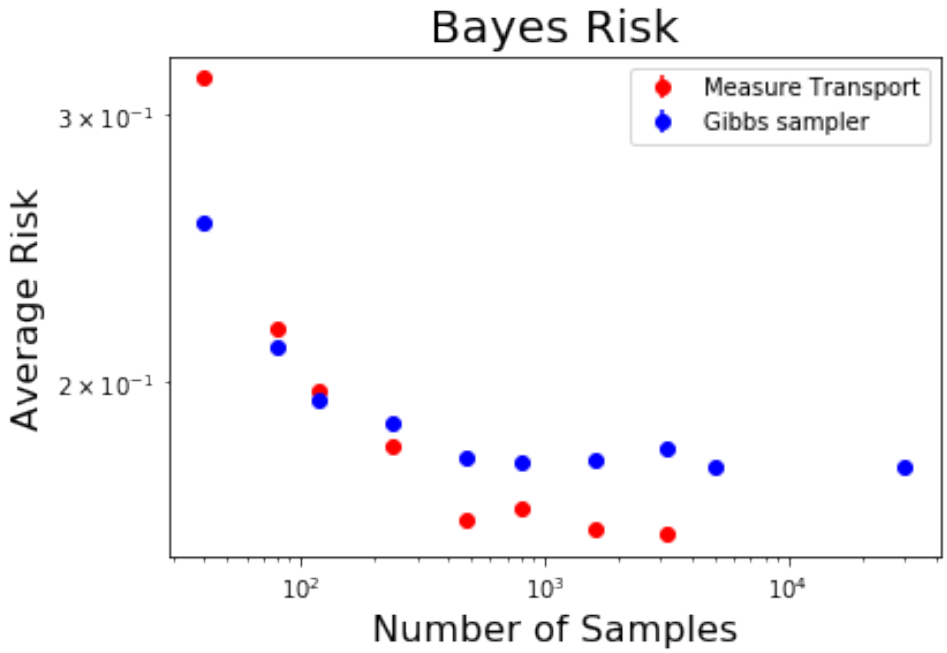


Figure 2.5. Approximation of the Bayes Risk: $R(P_X, d_N)$ generated with Gibbs samples and $R(P_X, \tilde{d}_N)$ generated with Optimal Transport samples plotted against N

methodology attains the Bayes Risk with a fewer number of training samples. Since our methodology produces i.i.d. samples from a distribution that approximates the posterior , it conserves effective sample sizes and approximates the Bayes Risk at a faster rate than Gibbs samples which are correlated. We also note that if N is large enough, we can generate an accurate transport map S with which we generate as many i.i.d. samples from a distribution that approximates the Bayesian Lasso posterior without any additional computation costs per sample. In contrast, the Gibbs sampling computation time is proportional to the number of samples generated.

2.6 Parallelized Implementation and Applications

The fundamentally parallel nature of our Bayesian Lasso formulation allows for solution implementation on a variety of platforms. The fact that our solution relies solely on linear algebra and Lasso solvers allows for it to be deployed in a variety of architectures for parallel computing.

In order to leverage the parallel nature of the algorithm presented above, we here present implementation with a Iterative-Reweighted Least Squares (IRLS) Lasso solver implemented in a Graphics Processing Unit (GPU) solution.

2.6.1 IRLS solver within a GPU Implementation

In the last several years Graphical Processing Units (GPUs) have gained significant attention for their parallel programmability. In this work, we made use of the ArrayFire library that abstracts low-level GPU programming and provides highly parallelized and optimized linear algebra algorithms [18].

We implemented Algorithm 1 using ArrayFire. To solve N Lasso problems of (2.7) we implemented a generalized iterative re-weighted least-squares (GIRLS) [2] algorithm. The GIRLS algorithm requires solving only least-squares sub-problems with linear algebra operations thus facilitating its implementation in ArrayFire.

Figure 2.6 shows execution times of computing a transport map with $N = 500$ and PCE order of 3 running a Python implementation on an Intel Core i7 processor at 2.40 GHz(4 CPUs) and running with the ArrayFire implementation on an NVIDIA GeForce 840M GPU. As the complexity of the problem increases (determined by increasing d), the ArrayFire implementation readily outperforms the Python implementation. This showcases the future possibilities for rapid computation of transport maps on architectures that feature parallelization capabilities.

2.7 Discussion and Conclusion

We have shown that an i.i.d. posterior Bayesian Lasso sampler can be constructed with a measure transport framework with iteratively solving a standard LASSO problem and performing closed-form linear algebra updates. There is a clear advantage to drawing i.i.d. samples from the Bayesian Lasso posterior that is seen in the faster convergence of posterior sampling methods, in comparison to Gibbs samplers, to approach the Bayes Risk. This formulation enables the leverage of the diversity of Lasso solvers to sample from posterior. For example, we show how

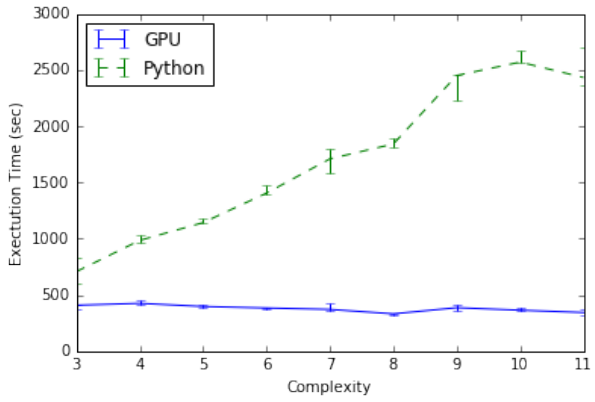


Figure 2.6. Execution times for computing a transport map in Python and using a GPU. The horizontal axis represents the dimension of the latent variable x.

posterior Bayesian Lasso transport samplers can be constructed with a GPU. We also note that this algorithm could be readily implemented in other systems for parallelization such as cloud computing.

Another potential application for inference with this transport-based approach is within the context of the Internet-of-Things (e.g. wearable electronics). In these settings, energy efficiency is of paramount importance, and wireless transmission usually is the most energy-consuming. Developing a framework such as ours where inference is performed on chip, thus obviating the need to transmit collected waveforms, enables only the need to transmit information about the posterior distribution, which is a sufficient statistic for any Bayesian decision making problem. In our case, this boils down to transmission of coefficients of the polynomials representing the transport map. From there, in the cloud for instance, i.i.d. prior samples may be transformed into i.i.d. posterior samples. Figure 2.7 shows a potential use of our posterior transmission scheme and a comparison to current transmission schemes.

Another energy-efficient application could be achieved by implementing our Bayesian Lasso algorithm in analog systems. The Local Competitive Algorithm (LCA) first presented in [27] is an analog dynamical system inspired by neural image processing and exactly solves (2.1). This system has already been implemented in field-programmable analog arrays [29] and integrate-and-fire neurons [12], thus showing promising results for reduced energy in hardware

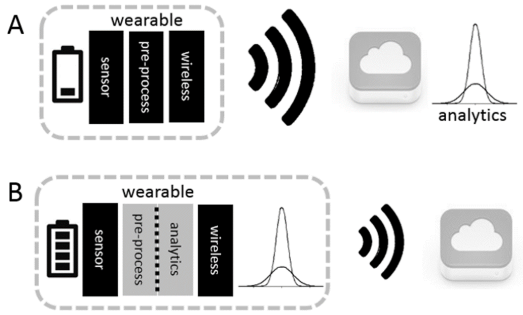


Figure 2.7. (A) shows conventional wireless transmission schemes where signals are acquired and wirelessly transmitted; (B) shows our proposed scheme where inference is performed locally and only the posterior distribution is transmitted.

implementations.

In the LCA, a set of parallel nodes, each associated with an element of the basis $\Phi_m \in \Phi$, compete with each other for representation of the input. The dynamics of LCA are expressed by a set of non-linear ordinary differential equations (ODEs) which represent simple analog components. The system's steady-state is the solution to (2.1). Using the formulation presented in Theorem 2.3.2, we could solve (2.26c) by presenting the LCA dynamics in terms of $\hat{y}$ and $\hat{\Phi}$.

$$\begin{aligned} \dot{u}_m(t) &= \frac{1}{\tau}\left[\langle \hat{\Phi}_m, \hat{y}\rangle - u_m(t) - \sum_{n\neq m} \langle \hat{\Phi}_m, \hat{\Phi}_n\rangle a_n(t)\right] \\ a_n(t) &\triangleq T_\lambda(u_n(t)) = \max(0, u_n(t) - \lambda) \end{aligned}$$

T_λ is a thresholding function that induces local non-linear competition between nodes.

We have presented a framework to find a posterior for the Bayesian Lasso, however this parallelizable formulation could be easily extended to other L_1 priors and sparsity problems. Dynamic formulations for spectrotemporal estimation of time series [28] could be extended to a fully-Bayesian perspective to enable improved statistical inference and decision making.

2.8 Acknowledgments

Chapter 2, in full, has been submitted for publication of the material as it may appear in Bayesian Analysis 2018. Mendoza, Marcela; Allegra, Alexis; Coleman, Todd. The disseration author was the primary investigator and author of this paper.

References

(1) Beck, A.; Teboulle, M. *SIAM journal on imaging sciences* **2009**, *2*, 183–202.

(2) Bissantz, N.; Dümbgen, L.; Munk, A.; Stratmann, B. *SIAM Journal on Optimization* **2009**, *19*, 1828–1845.

(3) Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. *Foundations and Trends in Machine Learning* **2011**.

(4) Brenier, Y. *Communications on pure and applied mathematics* **1991**, *44*, 375–417.

(5) Casella, G. *Biostatistics* **2001**, *2*, 485–500.

(6) Daubechies, I.; Defrise, M.; De Mol, C. *arXiv preprint math/0307152* **2003**.

(7) Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. *The Annals of statistics* **2004**, *32*, 407–499.

(8) El Moselhy, T.; Marzouk, Y. *Journal of Computational Physics* **2012**.

(9) Ernst, O. G.; Mugler, A.; Starkloff, H.-J.; Ullmann, E. *ESAIM: Mathematical Modelling and Numerical Analysis* **2012**, *46*, 317–339.

(10) Ghanem, R. G.; Spanos, P. D., *Stochastic finite elements: a spectral approach*; Courier Corporation: 2003.

(11) Goldstein, T.; Studer, C.; Baraniuk, R. *arXiv eprint* **2014**, *abs/1411.3406*.

(12) Gruenert, G.; Gizynski, K.; Escuela, G.; Ibrahim, B.; Gorecki, J.; Dittrich, P. *International journal of neural systems* **2014**.

(13) Hans, C. *Biometrika* **2009**, *96*, 835–845.

(14) Hastings, W. K. *Biometrika* **1970**, *57*, 97–109.

(15) Kim, S.; Quinn, C. J.; Kiyavash, N.; Coleman, T. P. *Proceedings of the IEEE* **2014**.

(16) Kim, S.; Ma, R.; Mesa, D.; Coleman, T. P. In *ISIT*, 2013.

(17) Lee, A.; Yau, C.; Giles, M. B.; Doucet, A.; Holmes, C. C. *Journal of computational and graphical statistics* **2010**, *19*, 769–789.

(18) Malcolm, J. In *Proc. of SPIE Vol*, 2012; Vol. 8403, 84030A–1.

(19) Marzouk, Y.; Moselhy, T.; Parno, M.; Spantini, A. *arXiv preprint arXiv:1602.05023* **2016**.

(20) Mesa, D.; Kim, S.; Coleman, T. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, 2015, pp 676–680.

(21) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. *The journal of chemical physics* **1953**, *21*, 1087–1092.

(22) Ng, A. Y. In *Proceedings of the twenty-first international conference on Machine learning*, 2004, p 78.

(23) Park, T.; Casella, G. *Journal of the American Statistical Association* **2008**, *103*, 681–686.

(24) Rajaratnam, B.; Sparks, D. *arXiv preprint arXiv:1509.03697* **2015**.

(25) Rajaratnam, B.; Sparks, D. *arXiv preprint arXiv:1508.00947* **2015**.

(26) Robert, C.; Casella, G., *Monte Carlo statistical methods*; Springer Science & Business Media: 2013.

(27) Rozell, C. J.; Johnson, D. H.; Baraniuk, R. G.; Olshausen, B. A. *Neural computation* **2008**, *20*, 2526–2563.

(28) Schamberg, G.; Ba, D.; Coleman, T. P. *arXiv preprint arXiv:1706.04685* **2017**.

(29) Shapero, S.; Charles, A. S.; Rozell, C. J.; Hasler, P. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on* **2012**, *2*, 530–541.

(30) Tibshirani, R. *Journal of the Royal Statistical Society. Series B (Methodological)* **1996**, 267–288.

(31) Villani, C., *Optimal transport: old and new*; Springer: 2008; Vol. 338.

(32) Wang, Z.; Bardsley, J. M.; Solonen, A.; Cui, T.; Marzouk, Y. M. *arXiv preprint arXiv:1607.01904* **2016**.

(33) Xiu, D.; Karniadakis, G. E. *SIAM Journal on Scientific Computing* **2002**, *24*, 619–644.

Chapter 3

L_1-Penalized Distributed Measure Transport

3.1 Introduction

Recently, the ability to find transformations between distributions has been the subject of much research. The general idea is to find a transformation or"transport map" between a reference measure P and a target measure Q. These transport maps have been increasingly useful and popularized in many Machine Learning applications [8]. The mathematical theory of Optimal Transport is a rich one that dates back to the 17th century [13]. Despite recent advances in algorithms to compute Measure Transport maps, the computation of these maps is still challenging, especially when the measures of interest lie in high dimensions. More recently, there has been interest in finding computationally efficient measure transport maps. Kim et al [7] showed that a map from a measure P to Q could be obtained through convex optimization methods and by parametrizing a map through a polynomial basis expansion. More recently, Mesa et al [11] showed that this convex optimization could be solved in a distributed manner by reformulation via an Alternating Direction Method of Multipliers (ADMM). Furthermore, this distributed formulation allows for exploiting parallelizable computational resources to find transport maps where the measures are in relatively high dimensions.

While the ability to exploit computational resources has enabled many applications, the number of parameters needed to accurately approximate a transport map is prohibitive when

these resources are limited. For measure transport maps parametrized with a basis of finite order polynomials O, the number of parameters needed is an exponential function in O and the dimension of the reference measure d [15].

Marzouk et al [10] and Mesa et al [11] have proposed other parametrizations of measure transport maps that are less "expressive" such as using a Knothe-Rosenblatt map [4], which is a lower-triangular map, as well as using radial basis functions. In this work, however, we are concerned with exploring the traditional or"dense" polynomial parametrization of transport maps and exploring more data-driven methods to reduce the number of parameters used to approximate them. In particular, we are interested in exploring whether a well-chosen"sparse" set of parameters will faithfully represent a transport map.

A powerful and well-known statistical model selection procedure is L_1 penalization. In the statistics literature, the Lasso [12] has been explored extensively for selecting pertinent features in regression problems. The Lasso uses L_1 penalization to drive the coefficients of the regressors to zero, providing a method for model selection. However, L_1 penalization is known to systemically shrink the magnitude of the coefficients of all regressors [5], thus, several studies have suggested to fit an OLS solution with the regressors selected by LASSO in order to better estimate their coefficient magnitudes. This is also known as re-fitting [5, 3, 9]. In the applications setting, Wu et al [14] used a similar method for first choosing relevant features for genome-wide association studies by Lasso regression and running an un-penalized solution with the chosen features.

Inspired by Lasso model selection and least-squares refitting techniques in regression, we propose using an L_1 penalized estimator of transport maps to first select a subset of the initial parameters that approximate a transport map. We empirically show that these set of parameters are usually captured by a cap in the order of the polynomials O. Then we run an un-penalized form of the problem with restricted polynomial order O to find a transport map. We also show model-selection algorithms to best choose the parameter of the L_1 regularization which determines the number of parameters that are excluded from the model. We showcase our

methodology with simulated data to show that our algorithms indeed capture the inherent order of a known transport map. We then showcase our algorithm with real data using composition of maps as described in [11].

The remainder of the paper is organized as follows: In Section 2 we provide definitions and notation conventions. In Section 3 we introduce the L_1-Penalized formulation and reformulation technique to find a Measure Transport Map using a relative entropy minimization approach as well as a distributed version of the L_1-Penalization optimization. We also introduce methodology to choose the level of L_1 penalization for transport maps. In Section 4 we showcase results with simulated data. In Section 5 we showcase results with real data. In Section 6 we conclude.

3.2 Definitions

In this section we make some preliminary definitions and provide background information for the rest of this paper.

3.2.1 Definitions and Assumptions

Assume the space for sampling is given by $\mathsf{W} \subset \mathbb{R}^D$, a convex subset of D-dimensional Euclidean space. Define the space of all probability measures on W (endowed with the Borel sigma-algebra) as $\mathsf{P}(\mathsf{W})$. If $P \in \mathsf{P}(\mathsf{W})$ admits a *density* with respect to the Lebesgue measure, we denote it as p.

Assumption 4. *We assume that $P, Q \in \mathsf{P}(\mathsf{W})$ admit densities p, q with respect to the Lebesgue measure.*

Definition 3.2.1 (Push-forward)**.** *Given $P, Q \in \mathsf{P}(\mathsf{W})$ we say that a map $S : \mathsf{W} \to \mathsf{W}$ pushes forward P to Q (denoted as $S_{\#}P = Q$) if a random variable X with distribution P results in $Y \triangleq S(X)$ having distribution Q.*

Of interest to us is the class of invertible and "smooth" push-forwards:

Definition 3.2.2 (Diffeomorphism)**.** *A mapping S is a diffeomorphism on* W *if it is invertible, and both S and S^{-1} are differentiable. Let $\mathscr{D}$ be the space of all diffeomorphisms on* W.

A subclass of these, are those that are "orientation preserving":

Definition 3.2.3 (Monotonic Diffeomorphism)**.** *A mapping $S \in \mathscr{D}$ is orientation preserving, or monotonic, if its Jacobian is positive-definite:*

$$J_S(u) \succeq 0, \quad \forall u \in \mathsf{W}$$

Let $\mathscr{D}_+ \subset \mathscr{D}$ be the set of all monotonic diffeomorphisms on W.

The Jacobian $J_S(u)$ can be thought of as how the map "warps" space to facilitate the desired mapping. Any monotonic diffeomorphism necessarily satisfies the following Jacobian equation:

Lemma 3.2.4 (Monotonic Jacobian Equation)**.** *Let $P, Q \in \mathsf{P}(\mathsf{W})$ and assume they have densities p and q. Any map $S \in \mathscr{M}\mathscr{D}$ for which $S\#P = Q$ satisfies the following Jacobian equation:*

$$p(u) = q(S(u)) \det(J_S(u)) \quad \forall u \in \mathsf{W} \tag{3.1}$$

3.3 L_1-Penalized Measure Transport

3.3.1 Relative Entropy Minimization

We start with the general relative entropy minimization formulation.

We can then cast the transport problem as finding the mapping $S \in \mathscr{D}_+$ that minimizes the relative entropy between P and the induced $\tilde{P}$.

$$S^* = \underset{S \in \mathscr{D}_+}{\arg\min} \; D(P \| \tilde{P}) \tag{3.2}$$

We can expand Eq. (3.2) and combine with (3.1) to write:

$$
\begin{aligned}
S^* &= \underset{S\in\mathscr{D}_+}{\arg\min}\ D(P\|\tilde{P}) \\
&= \underset{S\in\mathscr{D}_+}{\arg\min}\ \mathbb{E}_P\left[\log\frac{p(X)}{\tilde{p}(S(X))}\right] \\
&= \underset{S\in\mathscr{D}_+}{\arg\min}\ -h(p)-\mathbb{E}_P\left[\log\tilde{p}(S(X))\right] \qquad (3.3)\\
&= \underset{S\in\mathscr{D}_+}{\arg\min}\ -\mathbb{E}_P\left[\log\tilde{p}(S(X))\right] \qquad (3.4)\\
&= \underset{S\in\mathscr{D}_+}{\arg\min}\ -\mathbb{E}_P\left[\log q(S(X))+\log\det J_S(X)\right] \qquad (3.5)
\end{aligned}
$$

3.3.2 Parametrization of Transport Maps

To address the infinite dimensional space of functions mentioned above, as in [11, 7, 10] we parameterize the transport map over a space of multivariate polynomial basis functions formed as the product of D-many univariate polynomials of varying degree. That is, given some $\vec{x}=(x_1,\ldots,x_a,\ldots,x_D)\in\mathsf{W}\subset\mathbb{R}^D$, we form a basis function $\phi_{\vec{j}}(\vec{x})$ of multi-index degree $\vec{j}=(j_1,\ldots,j_a,\ldots,j_D)\in\mathscr{J}$ using univariate polynomials ψ_{j_a} of degree j_a as:

$$\phi_{\vec{j}}(\vec{x})=\prod_{a=1}^{D}\psi_{j_a}(x_a)$$

This allows us to represent one component of $S\in\mathscr{D}_+$ as a weighted linear combination of basis functions with weights $w_{d,\vec{j}}$ as:

$$S^d(\vec{x})=\sum_{\vec{j}\in\mathscr{J}} w_{d,\vec{j}}\,\phi_{\vec{j}}(\vec{x})$$

where $\mathscr{J}$ is a set of multi-indices in the representation specifying the order of the polynomials in the associated expansion, and d denotes the d^{th} component of the mapping. In order to make this

problem finite-dimensional, we must *truncate* the expansion to some fixed maximum-order O.

$$\mathscr{J} = \left\{ \vec{j} \in \mathbb{N}^D : \sum_{i=1}^{D} j_i \leq O \right\}$$

We can now approximate any nonlinear function $S \in \mathscr{D}_+$ as:

$$S(\vec{x}) = W\Phi(\vec{x})$$

where $K \triangleq |\mathscr{J}|$ the size of the index-set, $\Phi(\vec{x}) = [\phi_{\vec{j}_1}(\vec{x}), \dots \phi_{\vec{j}_K}(\vec{x})]^T$, and $W \in \mathbb{R}^{D\times K}$ is a matrix of weights.

With this, we can now give a finite-dimensional version of (3.5) as:

$$\begin{aligned} \min_{W\in\mathbb{R}^{D\times K}} \; & -\frac{1}{N}\sum_{i=1}^{N} \left[\log q(W\Phi(X_i)) + \log\det(WJ_\Phi(X_i))\right] \\ \text{s.t.} \quad & WJ_\Phi(X_i) \succeq 0, \quad i = 1,\dots,N \end{aligned} \tag{3.6}$$

with:

$$\begin{aligned} W &= [w_1, \dots, w_K] && D\times K \\ \Phi(\cdot) &= \left[\phi_{\vec{j}_1}(\cdot), \dots, \phi_{\vec{j}_K}(\cdot)\right]^T && K\times 1 \\ J_\Phi(\cdot) &= \left[\frac{\partial \phi_{\vec{j}_i}}{\partial x_j}(\cdot)\right]_{i,j} && K\times D \end{aligned}$$

where we have made explicit the implicit constraint that $\det(J_S) \geq 0$ by ensuring that $WJ_\Phi \succeq 0$. We now provide two important remarks:

Remark 7. *In principle, any basis of polynomials whose finite-dimensional approximations are sufficiently dense over* W *will suffice. In applications where P is assumed known, the basis functions are chosen to be orthogonal with respect to the reference measure P:*

$$\int_{\mathsf{W}} \phi_{\vec{j}}(\vec{x})\ \phi_{\vec{i}}(\vec{x})\ p(x)dx = \mathbb{1}_{\vec{i}=\vec{j}}$$

The total number of expansion terms in this truncated representation K can be calculated, where for a reference measure that is in D dimensional space by the following formula:

$$K = \frac{(D+O)!}{D!O!} \tag{3.7}$$

Thus we see that the number of terms for the parametrization grows exponentially with the dimension and maximum order of the polynomials. This causes problems when we are trying to compute transport maps with computational resources. In general, for high-dimensional and complex datasets, we will need a higher order polynomial representation to achieve an accurate transport map, and thus a high number of parameters to represent the map.

Other works, including our own [11, 10], have looked into parametrizing problem (3.5) with different types of bases and map structures. Specifically using a radial basis and using a Knothe-Rosenblatt transport map, which is a lower-triangular representation. These formulations, although useful, are less expressive than a dense polynomial parametrization as presented above.

In this paper, we present a method of uncovering the inherent maximum order O of a transport map by using a regularization method to select a subset of parameters that faithfully represent a transport map.

3.3.3 L_1 Group Regularization on Parameters

In this section we present the following L_1 regularized optimization problem for measure transport. Since columns of the coefficient matrix W represent coefficients from a particular basis function $\phi_{\vec{j}}(\vec{x})$, we resort to a group shrinkage regularization term. This formulation retains convexity.

Noting that

$$\begin{aligned} &\min_{W\in\mathbb{R}^{D\times K}} -\frac{1}{N}\sum_{i=1}^{N}\left[\log q(W\Phi(X_i)) + \log\det(WJ_\Phi(X_i))\right] + \beta||W||_{1,2} \\ &\text{s.t.} \quad WJ_\Phi(X_i) \succeq 0, \quad i = 1,\ldots,N \end{aligned} \tag{3.8}$$

The $\mathscr{L}_1$ term in (3.8) induces a penalty for the magnitude of the coefficient terms in W and drives some of these terms to zero.

The methodology here proposed is as follows:

1. Find a solution to (3.8) that induces sparse coefficients
2. determine O_{sparse} , the maximum order of non-zero coefficients from solution above
3. run a dense and truncated version of (3.6) with maximal polynomial order $O_{sparse} \leq O$

3.3.4 Distributed Formulation via ADMM

Similar to Mesa et al [11] we conjure a consensus ADMM formulation. This formulation, however , differs from the one in [11] since the regularization term is included.

$$\begin{aligned} &\min_{W_i \in \mathbb{R}^{D\times K}} \; -\frac{1}{N}\sum_{i=1}^{N} [\log q(W_i \Phi(X_i)) + \log\det(W_i J_\Phi(X_i)) + \beta ||W_i||_{1,2}] \\ &\text{s.t.} \quad W_i = W, \quad W_i J_\Phi(X_i) \succeq 0, \quad i = 1,\ldots,N \end{aligned} \tag{3.9}$$

The theory of ADMM enables a formulation where each term W_i can be solved in parallel. Introducing auxiliary variables and concensus variables:

$$
\begin{aligned}
&\min_{\{W,p,Z\}_i,Y,B} \frac{1}{N}\sum_{i=1}^{N} -\log q(p_i) - \log\det(Z_i) \\
&+\frac{1}{2}\rho||W_i - B||_F^2 + \frac{1}{2}||B\Phi_i - p_i||_2^2 \\
&+\frac{1}{2}\rho||BJ_i - Z_i||_F^2 + \frac{1}{2}\rho||B - Y||_F^2 \\
&+\beta||Y||_{1,2} \\
\text{s.t}\quad & B\Phi_i = p_i \qquad\qquad \gamma_i \quad (D\times 1) \\
& W_i - B = 0 \qquad\qquad \alpha_i \quad (D\times K) \\
& BJ_i = Z_i \qquad\qquad \lambda_i \quad (D\times D) \\
& B - Y = 0 \qquad\qquad \eta \quad (D\times K) \\
& Z_i \succeq 0
\end{aligned}
\tag{3.10}
$$

A fully penalized Lagrangian formulation of this problem is introduced and has the following form:

$$
\begin{aligned}
&L_{\rho,\beta}(W,Z,p,B,Y;\gamma,\lambda,\alpha,\eta) \\
&= \frac{1}{N}\sum_{i=1}^{N} -\log q(p_i) - \log\det Z_i \\
&+\frac{1}{2}\rho||W_i - B||_F^2 + \frac{1}{2}\rho||B\Phi_i - p_i||_2^2 \\
&+\frac{1}{2}\rho||BJ_i - Z_i||_F^2 + \frac{1}{2}\rho||B - Y||_F^2 \\
&+\gamma_i^T(p_i - B\Phi_i) + \mathbf{tr}(\alpha_i^T(W_i - B)) \\
&+\mathbf{tr}(\lambda_i^T(Z_i - BJ_i)) \\
&+\mathbf{tr}(\eta^T(B - Y)) + \beta||Y||_{1,2}
\end{aligned}
\tag{3.11}
$$

The updates are as follows

$$B^{(k+1)} = \mathscr{B}_i \cdot \mathscr{B}_s \tag{3.12a}$$

$$W_i^{(k+1)} = -\frac{1}{\rho}\alpha_i^{(k)} + B^{(k+1)} \tag{3.12b}$$

$$Z_i^{k+1} = Q\tilde{Z}_i Q^T \tag{3.12c}$$

$$Y^{(k+1)} = \arg\min_Y \frac{1}{2}\rho||B^{(k+1)} - Y||_F^2 + \mathbf{tr}(\eta^T(B^{(k+1)} - Y)) + \beta||Y||_{1,2} \tag{3.12d}$$

$$\gamma_i^{k+1} = \gamma_i^k + \rho(p_i^{k+1} - B^{k+1}\Phi_i) \tag{3.12e}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(Z_i^{k+1} - B^{k+1}J_i) \tag{3.12f}$$

$$\alpha_i^{k+1} = \alpha_i^k + \rho(W_i^{k+1} - B^{k+1}) \tag{3.12g}$$

$$\eta^{(k+1)} = \eta^{(k)} + \rho(B^{(k+1)} - Y^{(k+1)}) \tag{3.12h}$$

$$p_i^{(k+1)} = \arg\min_{p_i} -\log q(p_i) + \text{pen}(p_i) \tag{3.12i}$$

Theorem 3.3.1. *The Y update above from equation* (3.12d) *is given by a Group LASSO problem which can be solved in closed-form*

Proof.

$$Y^{(k+1)} = \arg\min_Y \frac{1}{2}\rho||B^{(k+1)} - Y||_F^2 + \mathbf{tr}(\eta^T(B^{(k+1)} - Y)) + \beta||Y||_{1,2} \tag{3.13}$$

$$= \arg\min_Y \frac{1}{2}\rho||B^{(k+1)} - Y + \frac{1}{\rho}\eta||_F^2 - \frac{1}{2}\rho||\frac{1}{\rho}\eta|| + \beta||Y||_{1,2} \tag{3.14}$$

$$= \arg\min_Y \frac{1}{2}\rho||B^{(k+1)} - Y + \frac{1}{\rho}\eta||_F^2 + \beta||Y||_{1,2} \tag{3.15}$$

$$= \arg\min_Y \frac{1}{2}||B^{(k+1)} + \frac{1}{\rho}\eta - Y||_F^2 + \frac{\beta}{\rho}||Y||_{1,2} \tag{3.16}$$

$$= \arg\min_Y \frac{1}{2}||\mathbf{I}Y - C||_F^2 + \frac{\beta}{\rho}||Y||_{1,2} \tag{3.17}$$

Where $C = B^{(k+1)} + \frac{1}{\rho}\eta$. Noticing that (3.17) is a group LASSO that satisfies the orthonormal regressor matrix condition (since the identity matrix is orthonormal), the proof

follows. □

3.3.5 Choosing the Level of Sparsity

The method above reduces overfitting by selecting a set of sparse coefficients of a transport map. The level of sparsity is determined by the parameter β. In order to estimate β we can use cross-validation or other model selection techniques. We chose to use a model that penalizes for the number of degrees of freedom as represented by the number of non-zero coefficients. We look at the model of the transport map and induce a penalty as a function of the number of active parameters K:

$$PNNLL(K) = -\frac{1}{m}\log \tilde{P}_K(X;S,\beta) + \text{penalty}(K) \tag{3.18}$$

Where the first term is represented by (3.5), the un-penalized problem. We used the Akaike Criterion [1] and the minimum description length (MDL) [2] as penalties.

$$\text{penalty}(K) = \begin{cases} \frac{K}{m} & \text{AIC} \\ \frac{K\log m}{2m} & \text{MDL} \end{cases} \tag{3.19}$$

β is chosen by finding the local minimum of the corresponding criterion.

3.4 Results with Simulated Data

In order to demonstrate the capability of regularization in uncovering the inherent order of a transport map, we show examples with simulated data. In particular, we consider transformations for which the inherent order O of the transport map is known. For example, we know that if a random variable Z distributed according to a Gaussian distribution then Z^2 will be χ^2 (Chi-square) distributed. Thus there exists a push-forward S that performs a transformation from Gaussian to Chi-square with order $O = 2$.

Table 3.1 shows the list of transformations simulated, where P refers to the reference

Table 3.1. Table of simulated transformations with corresponding polynomial order

Table of Transformations		
P	Q	Order
$Z \sim \mathcal{N}(\mu, \Sigma)$	$\mathcal{N}(\mu + B, A\Sigma)$	1
$Z \sim \mathcal{N}(\mu, \Sigma)$	χ^2	2
$Z \sim$ Pareto($a = 3$)	Unif$[0,1]$	3

distribution and Q refers to the target distribution. Thus we expect that an L_1 penalized map would uncover the order of the map.

We first obtain a dense transport map (3.6) with a relative high order $O = 5$. Then we obtain a penalized map for the above transformations by solving (3.8)with β chosen either by AIC and MDL criterion.

Figure 3.1 shows that the penalized maps do not choose coefficients which are higher than the maximum order of the map , as opposed to the regular maps. We see this as "uncovering" the maximum order of polynomials needed to represent a transport map S. However, we also see that the L_1 term systemically shrinks all coefficients. Therefore finding a map with (3.6) and a capped order $O_s parse$ performs better than the L_1 maps. We observed this by measuring and comparing the KL divergence between samples from an initial map and those of either a penalized map or a truncated map.

3.5 Results with Real Data

We showcase our algorithm within the context of density estimation. As shown in [11] the framework for learning a transport map can be used for density estimation of real datasets. We build on this work and show that we can learn the inherent transport map order of the dataset presented in [6] and learn a truncated version of the map that estimates the density of this dataset comparably good to the higher order map.

Note that a transport map $\hat{S}$ can arrive at a density estimate $\hat{P}$ utilizing the Jacobian

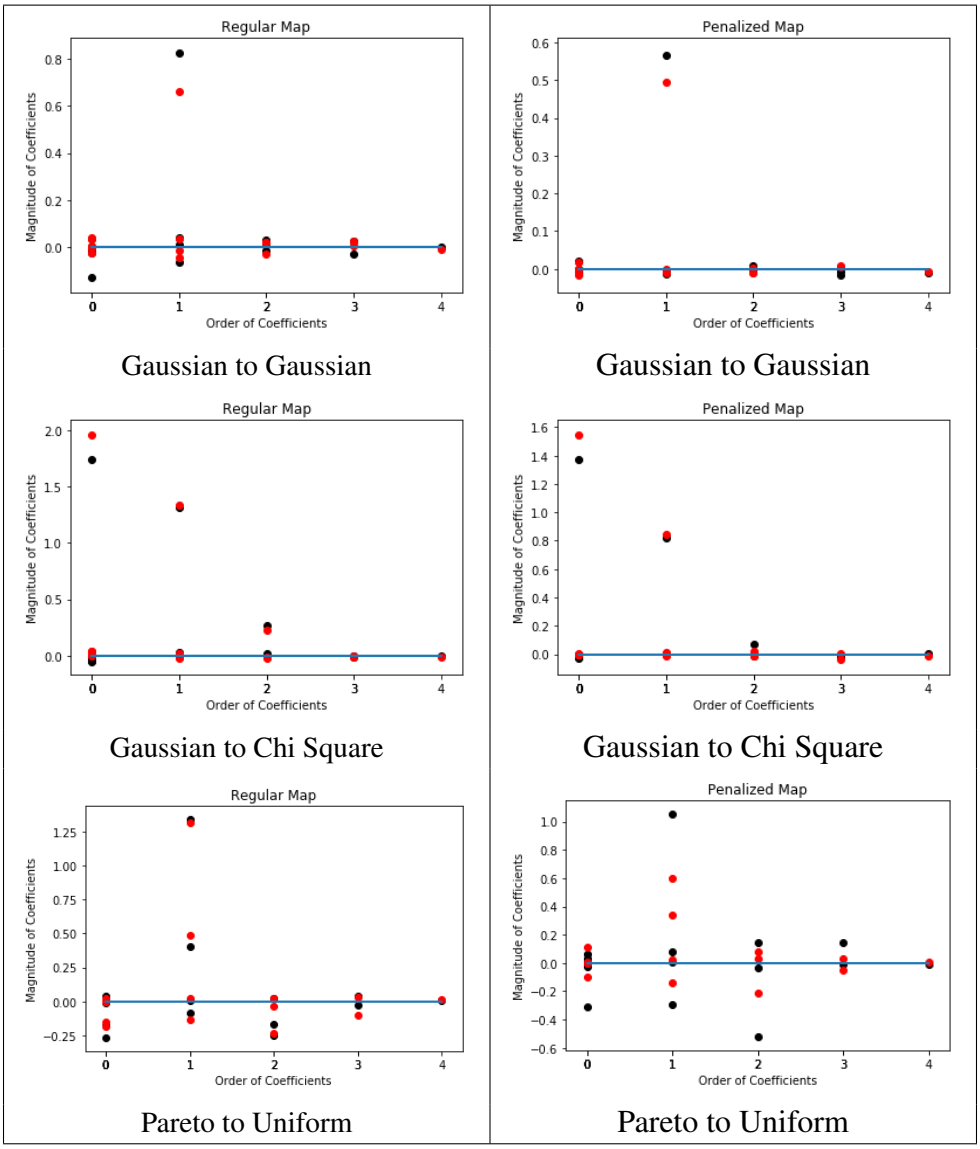


Figure 3.1. Magnitude of coefficients for varying known transformations.

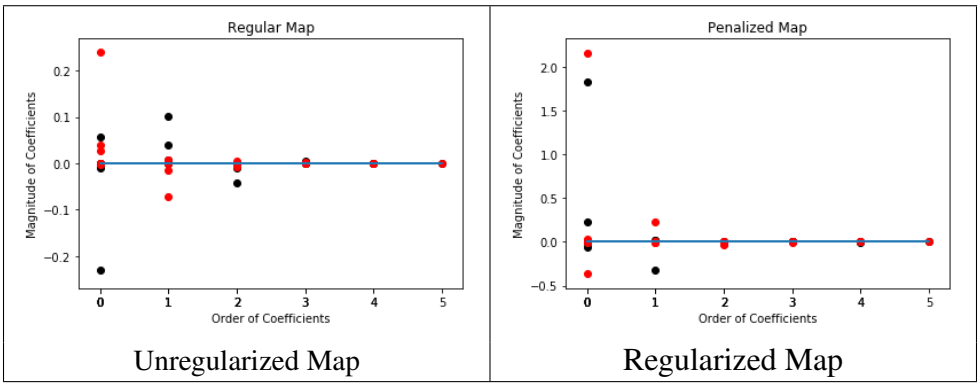


Figure 3.2. Magnitude of coefficients of transport map for light sleep data

equation.

$$\hat{p}(x) = q(\hat{S}(x)) \det J_{\hat{S}}(x) \tag{3.20}$$

We here look at the power of two electroencephalography (EEG) frequency bands when a patient is in "light" sleep and we wish to estimate the density associated with that stage of sleep P_{light}. We obtained real data from electroencephalography (EEG) analysis taken from a window of light sleep. P_{light} in $\mathbb{R}^2$ thus is the distribution over two important EEG frequency bands in determining light sleep. The problem then becomes constructing a transport map S^* that pushes P_{light} to a known $Q = \mathcal{N}(\mu, \Sigma)$. In this scenario, Q is a "dummy" density whose only purpose is to be known and well-behaved.

We first construct a transport map S^*_{dense} with a set basis order of 5 and estimate P_{light} with it. Then we construct a penalized map S^*_{pen} to uncover the inherent order which in this case appeared to be 2 (see Figure 3.2). Finally, we construct a truncated map S^*_{trun} of PCE order 2.

Figure 3.3 shows the density estimation estimates for the truncated and the original dense map. The density estimation is preserved by a lower order map without compromising the complexity of the density.

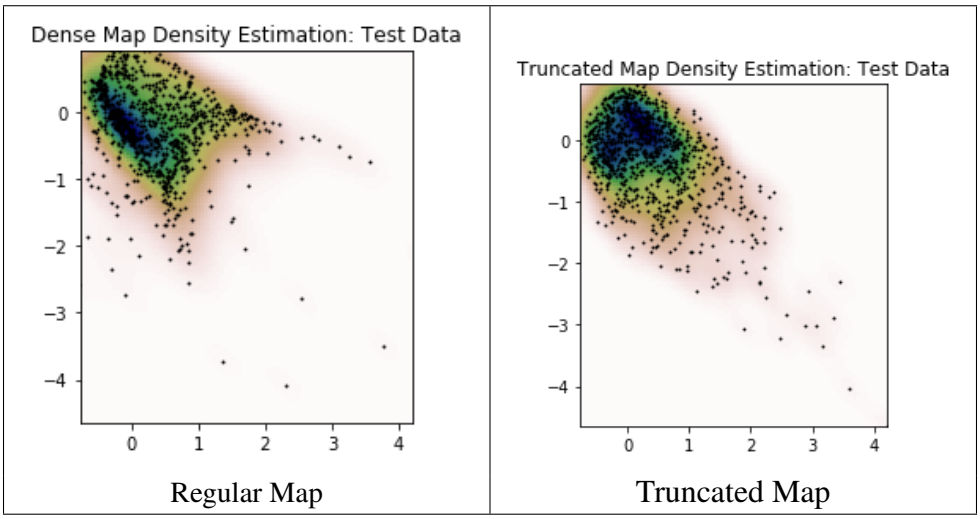


Figure 3.3. Density estimation of alpha and beta frequency bands in light sleep using a regular and a truncated polynomial order map

3.6 Conclusion

We have shown that using L_1 group regularization , we can find transport maps parametrized by polynomials that choose a sparse set of polynomial bases represented by a lower polynomial order O_{sparse} than that of the un-penalized problem. By this we show that a truncated map retains the expressivity of the original map.

3.7 Acknowledgements

Chapter 3, in part, is currently being prepared for submission for publication of the material. Mendoza, Marcela; Coleman, Todd P. The dissertation author was the primary investigator and author of this material.

References

(1) Akaike, H. *IEEE transactions on automatic control* **1974**, *19*, 716–723.

(2) Barron, A.; Rissanen, J.; Yu, B. *IEEE Transactions on Information Theory* **1998**, *44*, 2743–2760.

(3) Belloni, A.; Chernozhukov, V. *Bernoulli* **2013**, *19*, 521–547.

(4) Bonnotte, N. *SIAM Journal on Mathematical Analysis* **2013**, *45*, 64–87.

(5) Chzhen, E.; Hebiri, M.; Salmon, J. *arXiv preprint arXiv:1707.05232* **2017**.

(6) Coleman, T. P.; Tantiongloc, J.; Allegra, A.; Mesa, D.; Kang, D.; Mendoza, M. In *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*, 2016, pp 1330–1334.

(7) Kim, S.; Mesa, D.; Ma, R.; Coleman, T. P. *arXiv preprint arXiv:1509.08582* **2015**.

(8) Kolouri, S.; Park, S. R.; Thorpe, M.; Slepcev, D.; Rohde, G. K. *IEEE Signal Processing Magazine* **2017**, *34*, 43–59.

(9) Lederer, J. *arXiv preprint arXiv:1306.0113* **2013**.

(10) Marzouk, Y.; Moselhy, T.; Parno, M.; Spantini, A. *arXiv preprint arXiv:1602.05023* **2016**.

(11) Mesa, D. A.; Tantiongloc, J.; Mendoza, M.; Coleman, T. P. *arXiv preprint arXiv:1801.08454* **2018**.

(12) Tibshirani, R. *Journal of the Royal Statistical Society. Series B (Methodological)* **1996**, 267–288.

(13) Villani, C., *Optimal transport: old and new*; Springer: 2008; Vol. 338.

(14) Wu, T. T.; Chen, Y. F.; Hastie, T.; Sobel, E.; Lange, K. *Bioinformatics* **2009**, *25*, 714–721.

(15) Xiu, D.; Karniadakis, G. E. *SIAM Journal on Scientific Computing* **2002**, *24*, 619–644.