arXiv:1705.07443v1 [cs.LG] 21 May 2017

Parallel Streaming Wasserstein Barycenters

Matthew Staib, Sebastian Claici, Justin Solomon, and Stefanie Jegelka

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{mstaib, sclaici, jsolomon, stefje}@mit.edu

Abstract

Efficiently aggregating data from different sources is a challenging problem, particularly when samples from each source are distributed differently. These differences can be inherent to the inference task or present for other reasons: sensors in a sensor network may be placed far apart, affecting their individual measurements. Conversely, it is computationally advantageous to split Bayesian inference tasks across subsets of data, but data need not be identically distributed across subsets. One principled way to fuse probability distributions is via the lens of optimal transport: the Wasserstein barycenter is a single distribution that summarizes a collection of input measures while respecting their geometry. However, computing the barycenter scales poorly and requires discretization of all input distributions and the barycenter itself. Improving on this situation, we present a scalable, communication-efficient, parallel algorithm for computing the Wasserstein barycenter of arbitrary distributions. Our algorithm can operate directly on continuous input distributions and is optimized for streaming data. Our method is even robust to nonstationary input distributions and produces a barycenter estimate that tracks the input measures over time. The algorithm is semi-discrete, needing to discretize only the barycenter estimate. To the best of our knowledge, we also provide the first bounds on the quality of the approximate barycenter as the discretization becomes finer. Finally, we demonstrate the practical effectiveness of our method, both in tracking moving distributions on a sphere, as well as in a large-scale Bayesian inference task.

1 Introduction

A key challenge when scaling up data aggregation occurs when data comes from multiple sources, each with its own inherent structure. Sensors in a sensor network may be configured differently or placed far apart, but each individual sensor simply measures a different view of the same quantity. Similarly, user data collected by a server in California will differ from that collected by a server in Europe: the data samples may be independent but are not identically distributed.

One reasonable approach to aggregation in the presence of multiple data sources is to perform inference on each piece independently and fuse the results. This is possible when the data can be distributed randomly, using methods akin to distributed optimization [Zhang et al., 2013, 2015]. However, when the data is *not* split in an i.i.d. way, Bayesian inference on different subsets of observed data yields slightly different “subset posterior” distributions for each subset that must be combined [Minsker et al., 2014]. Further complicating matters, data sources may be nonstationary.

How can we fuse these different data sources for joint analysis in a consistent and structure-preserving manner?

We address this question using ideas from the theory of *optimal transport*. Optimal transport gives us a principled way to measure distances between measures that takes into account the underlying space on which the measures are defined. Intuitively, the optimal transport distance between two distributions measures the amount of *work* one would have to do to move all mass from one distribution to the other. Given J input measures $\{\mu_j\}_{j=1}^J$, it is natural, in this setting, to ask for a measure ν that minimizes the total squared distance to the input measures. This measure ν is called the *Wasserstein barycenter* of the input measures [Agueh and Carlier, 2011], and should be thought of as an aggregation of the input measures which preserves their geometry. When the measures are discrete, their barycenter can be computed relatively efficiently via either a sparse linear program [Anderes et al., 2016], or regularized projection-based methods [Cuturi and Doucet, 2014; Benamou et al., 2015; Ye et al., 2017; Cuturi and Peyré, 2016]. However, when the input measures are continuous, to the best of our knowledge the only option is to discretize them via sampling.

Given sample access to J potentially continuous distributions μ_j, we propose a communication-efficient, parallel algorithm to estimate their barycenter. Our method can be parallelized to J worker machines, and the messages sent between machines are merely single integers. We require a discrete approximation of the barycenter itself, making our algorithm *semi-discrete*, but our algorithm scales well to fine approximations (e.g. $n \sim 10^6$). In contrast to previous work, we provide guarantees on the quality of the approximation as n increases. These rates apply to the general setting in which the μ_j's are defined on manifolds, with applications to directional statistics [Sra, 2016]. Our algorithm is based on stochastic gradient descent as in [Genevay et al., 2016] and hence is robust to gradual changes in the distributions: as the μ_j's change over time, we maintain a moving estimate of their barycenter, a task which is not possible using current methods without solving a large linear program in each iteration.

We emphasize that we aggregate the input distributions into a summary, the barycenter, which is itself a distribution. Instead of performing any single domain-specific task such as clustering or estimating an expectation, we can simply compute the barycenter of the inputs and process it later any arbitrary way. This generality coupled with the efficiency and parallelism of our algorithm yields immediate applications in fields from large scale Bayesian inference to e.g. streaming sensor fusion.

Contributions. **1.** We give a communication-efficient and fully parallel algorithm for computing the barycenter of a collection of distributions. Though our algorithm is semi-discrete, we stress that the input measures can be *continuous*, and even *nonstationary*. **2.** We give bounds on the quality of the recovered barycenter as our discretization becomes finer. These are the first such bounds which we are aware of, and they apply to measures on arbitrary compact and connected manifolds. **3.** We demonstrate the practical effectiveness of our method, both in tracking moving distributions on a sphere, as well as in a real large-scale Bayesian inference task.

1.1 Related work

Optimal transport. A comprehensive treatment of optimal transport and its many applications is beyond the scope of our work. We refer the interested reader to the detailed monographs by Villani [2009] and Santambrogio [2015]. Fast algorithms for optimal transport have been developed in recent years via Sinkhorn's algorithm [Cuturi, 2013] and in particular stochastic gradient methods [Genevay et al., 2016], which we build off of in this work. These algorithms have enabled several applications of optimal transport and Wasserstein metrics to machine learning, for example in supervised learning [Frogner et al., 2015], unsupervised learning [Montavon et al., 2016; Arjovsky et al., 2017], and domain adaptation [Courty et al., 2016]. Wasserstein barycenters in particular have been applied to a wide variety of problems including fusion of subset posteriors [Srivastava et al., 2015a], distribution clustering [Ye et al., 2017], shape and texture interpolation [Solomon et al., 2015; Rabin et al., 2011], and multi-target tracking [Baum et al., 2015].

When the distributions μ_j are discrete, transport barycenters can be computed relatively efficiently via either a sparse linear program [Anderes et al., 2016] or regularized projection-based methods [Cuturi and Doucet, 2014; Benamou et al., 2015; Ye et al., 2017; Cuturi and Peyré, 2016]. In settings like posterior inference, however, the distributions μ_j are likely continuous rather than discrete, and the most obvious viable approach requires discrete approximation of each μ_j. The resulting discrete barycenter converges to the true, continuous barycenter as the approximations become finer [Boissard et al., 2015; Kim and Pass, 2017], but the rate of convergence is not well-understood, and finely approximating each μ_j yields a very large linear program.

Scalable Bayesian inference. Scaling Bayesian inference to large datasets has become an important topic in recent years. There are many approaches to this, ranging from parallel Gibbs sampling [Newman et al., 2008; Johnson et al., 2013] to stochastic and streaming algorithms [Welling and Teh, 2011; Chen et al., 2014; Hoffman et al., 2013; Broderick et al., 2013]. For a more complete picture, we refer the reader to the survey by Angelino et al. [2016].

One promising method is via subset posteriors: instead of sampling from the posterior distribution given by the full data, the data is split into smaller tractable subsets. Performing inference on each subset yields several subset posteriors, which are biased but can be combined via their Wasserstein barycenter [Srivastava et al., 2015a], with provable guarantees on approximation quality. This is in contrast to other methods which rely on summary statistics to estimate the true posterior [Minsker et al., 2014; Neiswanger et al., 2914] and that require additional assumptions. In fact, our algorithm works with arbitrary measures and on manifolds.

2 Background

Let $(\mathcal{X}, d)$ be a metric space. Given two probability measures $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{X})$ and a cost function $c : \mathcal{X} \times \mathcal{X} \to [0, \infty)$, the Kantorovich problem asks for a solution to

$$\inf \left\{ \int_{\mathcal{X}\times\mathcal{X}} c(x,y) d\gamma(x,y) : \gamma \in \Pi(\mu,\nu) \right\} \tag{1}$$

where $\Pi(\mu, \nu)$ is the set of measures on the product space $\mathcal{X} \times \mathcal{X}$ whose marginals evaluate to μ and ν respectively.

Under mild conditions on the cost function (lower semi-continuity) and the underlying space (completeness and separability), problem (1) admits a solution Santambrogio [2015]. Moreover, if the cost function is of the form $c(x, y) = d(x, y)^p$, the optimal transportation cost is a distance metric on the space of probability measures. This is known as the *Wasserstein distance* and is given by

$$W_p(\mu,\nu) = \left(\inf_{\gamma\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^p d\gamma(x,y) \right)^{1/p}. \tag{2}$$

Optimal transport has recently attracted much attention in machine learning and adjacent communities [Frogner et al., 2015; Montavon et al., 2016; Courty et al., 2016; Peyré et al., 2016; Rolet et al., 2016; Arjovsky et al., 2017]. When μ and ν are discrete measures, problem (2) is a linear program, though faster regularized methods based on Sinkhorn iteration are used in practice Cuturi [2013]. Optimal transport can also be computed using stochastic first-order methods Genevay et al. [2016].

Now let $\mu_1, \ldots, \mu_J$ be measures on $\mathcal{X}$. The Wasserstein barycenter problem, introduced by Agueh and Carlier [2011], is to find a measure $\nu \in \mathcal{P}(\mathcal{X})$ which minimizes the functional

$$F[\nu] := \frac{1}{J} \sum_{j=1}^{J} W_2^2(\mu_j, \nu). \tag{3}$$

Finding the *barycenter* ν is the primary problem we address in this paper. When each μ_j is a discrete measure, the exact barycenter can be found via linear programming [Anderes et al., 2016], and many of the regularization techniques apply for approximating it [Cuturi and Doucet, 2014; Cuturi and Peyré, 2016]. However, the problem size grows quickly with the size of the support. When the measures μ_j are truly continuous, we are aware of only one strategy: sample from each μ_j in order to approximate it by the empirical measure, and then solve the discrete barycenter problem.

We directly address the problem of computing the barycenter when the input measures can be continuous. We solve a semi-discrete problem, where the target measure is a finite set of points, but we do not discretize any other distribution.

3 Algorithm

We first provide some background on the dual formulation of optimal transport. Then we derive a useful form of the barycenter problem, provide an algorithm to solve it, and prove convergence guarantees. Finally, we demonstrate how our algorithm can easily be parallelized.

3.1 Mathematical preliminaries

The primal optimal transport problem (2) admits a dual problem [Santambrogio, 2015]:

$$OT_c(\mu, \nu) = \sup_{v \text{ 1-Lipschitz}} \left\{ \mathbb{E}_{Y\sim\nu}[v(Y)] + \mathbb{E}_{X\sim\mu}[v^c(X)] \right\}, \tag{4}$$

where $v^c(x) = \inf_{y\in\mathcal{X}}\{c(x,y) - v(y)\}$ is the *c-transform* of v [Villani, 2009]. When $\nu = \sum_{i=1}^n w_i \delta_{y_i}$ is discrete, problem (4) becomes the *semi-discrete* problem

$$OT_c(\mu, \nu) = \max_{v\in\mathbb{R}^n} \left\{ \langle w, v\rangle + \mathbb{E}_{X\sim\mu}[h(X, v)] \right\}, \tag{5}$$

where we define $h(x, v) = v^c(x) = \min_{i=1,\ldots,n}\{c(x, y_i) - v_i\}$. Semi-discrete optimal transport admits efficient algorithms [Lévy, 2015; Kitagawa et al., 2016]; Genevay et al. [2016] in particular observed that given sample oracle access to μ, the semi-discrete problem can be solved via stochastic gradient ascent.

3.2 Deriving the optimization problem

Absolutely continuous measures can be approximated arbitrarily well by discrete distributions with respect to Wasserstein distance [Kloeckner, 2012]. Hence one natural approach to the barycenter problem (3) is to approximate the true barycenter via discrete approximation: fixing n support points $\{y_i\}_{i=1}^n \in \mathcal{X}$, we wish to find the discrete distribution $\nu_n = \sum_{i=1}^n w_i \delta_{y_i}$ with support on n points which optimizes

$$\min_{w \in \Delta_n} F(w) = \min_{w \in \Delta_n} \frac{1}{J} \sum_{j=1}^J W_2^2(\mu_j, \nu_n) \tag{6}$$

$$= \min_{w \in \Delta_n} \left\{ \frac{1}{J} \sum_{j=1}^J \max_{v^j \in \mathbb{R}^n} \left\{ \langle w, v^j \rangle + \mathbb{E}_{X_j \sim \mu_j}[h(X_j, v^j)] \right\} \right\}. \tag{7}$$

where we have defined $F(w) := F[\nu_n] = F[\sum_{i=1}^n w_i \delta_{y_i}]$ and used the dual formulation from equation (5). We will discuss in Section 4 the effect of different choices for the support points $\{y_i\}_{i=1}^n$.

Noting that the variables v^j are uncoupled, we can rearrange to get the following problem:

$$\min_{w \in \Delta_n} \max_{v^1, \ldots, v^J} \frac{1}{J} \sum_{j=1}^J \left[\langle w, v^j \rangle + \mathbb{E}_{X_j \sim \mu_j}[h(X_j, v^j)] \right]. \tag{8}$$

Problem (8) is convex in w and jointly concave in the v^j, and we can compute an unbiased gradient estimate for each by sampling $X_j \sim \mu_j$. Hence, we could solve this saddle-point problem via simultaneous (sub)gradient steps as in Nemirovski and Rubinstein [2005]. Such methods are simple to implement, but in the current form we must project onto the simplex Δ_n at each iteration. This requires only $O(n \log n)$ time [Held et al., 1974; Michelot, 1986; Duchi et al., 2008] but makes it hard to decouple the problem across each distribution μ_j. Fortunately, we can reformulate the problem in a way which avoids projection entirely. By strong duality, Problem (8) can be written as

$$\max_{v^1, \ldots, v^J} \min_{w \in \Delta_n} \left\{ \left\langle \frac{1}{J} \sum_{j=1}^J v^j, w \right\rangle + \frac{1}{J} \sum_{j=1}^J \mathbb{E}_{X_j \sim \mu_j}[h(X_j, v^j)] \right\} \tag{9}$$

$$= \max_{v^1, \ldots, v^J} \left\{ \min_i \left\{ \frac{1}{J} \sum_{j=1}^J v_i^j \right\} + \frac{1}{J} \sum_{j=1}^J \mathbb{E}_{X_j \sim \mu_j}[h(X_j, v^j)] \right\}. \tag{10}$$

Note how the variable w disappears: for any fixed vector b, minimization of $\langle b, w \rangle$ over $w \in \Delta_n$ is equivalent to finding the minimum element of b. The optimal w can also be computed in closed form when the barycentric cost is entropically regularized as in [Bigot et al., 2016], which may yield better convergence rates but requires dense updates that, e.g. need more communication. In either case, we are left with a concave maximization problem in $v^1, \ldots, v^J$, to which we can directly apply stochastic gradient ascent. Unfortunately the gradients are still not sparse and decoupled, so we turn problem (10) into the constrained problem

$$\max_{s, v^1, \ldots, v^J} \frac{1}{J} \sum_{j=1}^J \left[\frac{1}{J} \min_i s_i + \mathbb{E}_{X_j \sim \mu_j}[h(X_j, v^j)] \right] \quad \text{s.t.} \quad s = \sum_{j=1}^J v^j. \tag{11}$$

3.3 Algorithm and convergence

Algorithm 1 Subgradient Ascent

```
$s, v^1, \dots, v^J \leftarrow 0_n$
loop
    Draw $j \sim \text{Unif}[1, \dots, J]$
    Draw $x \sim \mu_j$
    $i_W \leftarrow \text{argmin}_i \{c(x, y_i) - v_i^j\}$
    $i_M \leftarrow \text{argmin}_i \, s_i$
    $v_{i_W}^j \leftarrow v_{i_W}^j - \gamma$              ▷ Gradient update
    $s_{i_M} \leftarrow s_{i_M} + \gamma/J$              ▷ Gradient update
    $v_{i_W}^j \leftarrow v_{i_W}^j + \gamma/2$              ▷ Projection
    $v_{i_M}^j \leftarrow v_{i_M}^j + \gamma/(2J)$              ▷ Projection
    $s_{i_W} \leftarrow s_{i_W} - \gamma/2$              ▷ Projection
    $s_{i_M} \leftarrow s_{i_M} - \gamma/(2J)$              ▷ Projection
end loop
```

We can now solve this problem via stochastic projected subgradient ascent. This is described in Algorithm 1; note that the sparse adjustments after the gradient step are actually projections onto the constraint set with respect to the ℓ_1 norm. Derivation of this sparse projection step is given rigorously in Appendix A. Not only do we have an optimization algorithm with sparse updates, but we can even recover the optimal weights w from standard results in online learning [Freund and Schapire, 1999]. Specifically, in a zero-sum game where one player plays a no-regret learning algorithm and the other plays a best-response strategy, the average strategies of both players converge to optimal:

Theorem 3.1. *Perform T iterations of stochastic subgradient ascent on $u = (s, v^1, \dots, v^J)$ as in Algorithm 1, and use step size $\gamma = \frac{R}{4\sqrt{T}}$, assuming $\|u_t - u^*\|_1 \leq R$ for all t. Let i_t be the minimizing index chosen at iteration t, and write $\overline{w}_T = \frac{1}{T}\sum_{t=1}^{T} e_{i_t}$. Then we can bound*

$$\mathbb{E}[F(\overline{w}_T) - F(w^*)] \leq 4R/\sqrt{T}. \tag{12}$$

The expectation is with respect to the randomness in the subgradient estimates g_t.

Theorem 3.1 is proved in Appendix B. The proof combines the zero-sum game idea above, which itself comes from [Freund and Schapire, 1999], with a regret bound for online gradient descent [Zinkevich, 2003; Hazan, 2016].

3.4 Parallel Implementation

The key realization which makes our barycenter algorithm truly scalable is that the variables $s, v^1, \dots, v^J$ can be separated across different machines. In particular, the "sum" or "coupling" variable s is maintained on a master thread which runs Algorithm 2, and each v^j is maintained on a worker thread running Algorithm 3. Each projected gradient step requires first selecting distribution j. The algorithm then requires computing only $i_W = \text{argmin}_i\{c(x_j, y_i) - v_i^j\}$ and $i_M = \text{argmin}_i \, s_i$, and then updating s and v^j in only those coordinates. Hence only a small amount of information (i_W and i_M) need pass between threads.

Where are the bottlenecks? When there are n points in the discrete approximation, each worker's task of computing $\text{argmin}_i\{c(x_j, y_i) - v_i^j\}$ requires $O(n)$ computations of $c(x, y)$. The master must iteratively find the minimum element s_{i_M} in the vector s, then update s_{i_M}, and decrease element s_{i_W}. These can be implemented respectively as the "find min", "delete min" then "insert," and "decrease min" operations in a Fibonacci heap. All these operations together take amortized $O(\log n)$ time. Hence, it takes $O(n)$ time it for all J workers to each produce one gradient sample in parallel, and only $O(J \log n)$ time for the master to process them all. Of course, communication is not free, but the messages are small and our approach should scale up well for $J \ll n$.

This parallel algorithm is particularly well-suited to the Wasserstein posterior (WASP) [Srivastava et al., 2015b] framework for merging Bayesian subset posteriors. In this setting, we split the dataset $X_1, \dots, X_k$ into J subsets $S_1, \dots, S_J$ each with k/J data points, distribute those subsets to J different machines, then each machine runs Markov Chain Monte Carlo (MCMC) to sample from $p(\theta|S_i)$, and we aggregate these posteriors via their barycenter. The most expensive subroutine in the worker thread is actually sampling from the posterior, and everything else is cheap in comparison. In particular, the machines need not even share samples from their respective MCMC chains.

One subtlety is that selecting worker j truly uniformly at random each iteration requires more synchronization, hence our gradient estimates are not actually independent as usual. Selecting worker threads as they are available will fail to yield a uniform distribution over j, as at the moment worker j finishes one gradient step, the probability that worker j is the next available is much less than $1/J$: worker j must resample and recompute i_W, whereas other threads would have a head start. If workers all took precisely the same amount of time, the ordering of worker threads would be deterministic, and guarantees for without-replacement sampling variants of stochastic gradient ascent would apply [Shamir, 2016]. In practice, we have no issues with our approach.

Algorithm 2 Master Thread

```
Input: index $j$, distribution $\mu$, atoms $\{y_i\}_{i=1,\dots,N}$,
number $J$ of distributions, step size $\gamma$
Output: barycenter weights $w$
$c \leftarrow 0_n$
$s \leftarrow 0_n$
$i_M \leftarrow 1$
loop
    $i_W \leftarrow$ message from worker $j$
    Send $i_M$ to worker $j$
    $c_{i_M} \leftarrow c_{i_M} + 1$
    $s_{i_M} \leftarrow s_{i_M} + \gamma/(2J)$
    $s_{i_W} \leftarrow s_{i_W} - \gamma/2$
    $i_M \leftarrow \mathrm{argmin}_i\, s_i$
end loop
return $w \leftarrow c/(\sum_{i=1}^n c_i)$
```

Algorithm 3 Worker Thread

```
Input: index $j$, distribution $\mu$, atoms $\{y_i\}_{i=1,\dots,N}$,
number $J$ of distributions, step size $\gamma$
$v \leftarrow 0_n$
loop
    Draw $x \sim \mu$
    $i_W \leftarrow \mathrm{argmin}_i\{c(x, y_i) - v_i\}$
    Send $i_W$ to master
    $i_M \leftarrow$ message from master
    $v_{i_M} \leftarrow v_{i_M} + \gamma/(2J)$
    $v_{i_W} \leftarrow v_{i_W} - \gamma/2$
end loop
```

4 Consistency

Prior methods for estimating the Wasserstein barycenter ν^* of continuous measures $\mu_j \in \mathcal{P}(\mathcal{X})$ involve first approximating each μ_j by a measure $\mu_{j,n}$ which has finite support on n points, then computing the barycenter ν_n^* of $\{\mu_{j,n}\}$ as a surrogate for ν^*. This approach is consistent, in that if $\mu_{j,n} \to \mu_j$ as $n \to \infty$, then also $\nu_n^* \to \nu^*$. This holds even if the barycenter is not unique, both in the Euclidean case [Boissard et al., 2015, Theorem 3.1] as well as when $\mathcal{X}$ is a Riemannian manifold [Kim and Pass, 2017, Theorem 5.4]. However, it is not known how fast the approximation ν_n^* approaches the true barycenter ν^*, or even how fast the barycentric distance $F[\nu_n^*]$ approaches $F[\nu_n]$.

In practice, not even the approximation ν_n^* is computed exactly: instead, support points are chosen and ν_n^* is constrained to have support on those points. There are various heuristic methods for choosing these support points, ranging from mesh grids of the support, to randomly sampling points from the convex hull of the supports of μ_j , or even optimizing over the support point locations. Yet we are unaware of any rigorous guarantees on the quality of these approximations.

While our approach still involves approximating the barycenter ν^* by a measure ν_n^* with fixed support, we are able to provide bounds on the quality of this approximation as $n \to \infty$. Specifically, we bound the rate at which $F[\nu_n^*] \to F[\nu_n]$. The result is intuitive, and appeals to the notion of an ϵ-cover of the support of the barycenter:

Definition 4.1 (Covering Number)**.** The *ϵ-covering number* of a compact set $K \subset \mathcal{X}$, with respect to the metric g, is the minimum number $\mathcal{N}_\epsilon(K)$ of points $\{x_i\}_{i=1}^{\mathcal{N}_\epsilon(K)} \in K$ needed so that for each $y \in K$, there is some x_i with $g(x_i, y) \leq \epsilon$. The set $\{x_i\}$ is called an ϵ-covering.

Definition 4.2 (Inverse Covering Radius)**.** Fix $n \in \mathbb{Z}^+$. We define the *n-inverse covering radius* of compact $K \subset \mathcal{X}$ as the value $\epsilon_n(K) = \inf\{\epsilon > 0 : \mathcal{N}_\epsilon(K) \leq n\}$, when n is large enough so the infimum exists.

Suppose throughout this section that $K \subset \mathbb{R}^d$ is endowed with a Riemannian metric g, where K has diameter D. In the specific case where g is the usual Euclidean metric, there is an ϵ-cover for K with at most $C_1\epsilon^{-d}$ points, where C_1 depends only on the diameter D and dimension d [Shalev-Shwartz and Ben-David, 2014]. Reversing the inequality, K has an n-inverse covering radius of at most $\epsilon \leq C_2 n^{-1/d}$ when n takes the correct form.

We now present and then prove our main result:

Theorem 4.1. *Suppose the measures μ_j are supported on K, and suppose μ_1 is absolutely continuous with respect to volume. Then the barycenter ν^* is unique. Moreover, for each empirical approximation size n, if we choose support points $\{y_i\}_{i=1,\dots,n}$ which constitute a $2\epsilon_n(K)$-cover of K, it follows that $F[\nu_n^*] - F[\nu^*] \leq O(\epsilon_n(K) + n^{-1/d})$, where $\nu_n^* = \sum_{i=1}^n w_i^* \delta_{y_i}$ for w^* solving Problem (8).*

Proof. For any two measures η, η' supported on K, we can bound $W_2(\eta, \eta') \leq D$: the worst-case η, η' are point masses distance D apart, so that the transport plan sends all the mass a distance of D.

It follows that $|W_2(\mu, \nu_n) + W_2(\mu, \nu)| \leq 2D$ and therefore

$$|W_2^2(\mu, \nu_n) - W_2^2(\mu, \nu)| \leq 2D \cdot |W_2(\mu, \nu_n) - W_2(\mu, \nu)| \tag{13}$$
$$\leq 2D \cdot W_2(\nu_n, \nu) \tag{14}$$

by the triangle inequality. Summing over all $\mu = \mu_j$, we find that

$$|F[\nu_n] - F[\nu]| \leq \frac{1}{J} \sum_{j=1}^{J} |W_2^2(\mu_j, \nu_n) - W_2^2(\mu_j, \nu)| \tag{15}$$
$$\leq \frac{1}{J} \sum_{j=1}^{J} 2D \cdot W_2(\nu_n, \nu) = 2D \cdot W_2(\nu_n, \nu), \tag{16}$$

completing the proof. □

Remark 4.1. Absolute continuity is only needed to reason about approximating the barycenter with an N point discrete distribution. If the input distributions are themselves discrete distributions, so is the barycenter, and we can strengthen our result. For large enough n, we actually have $W_2(\nu_n^*, \nu^*) \leq 2\epsilon_n(K)$ and therefore $F[\nu_n^*] - F[\nu^*] \leq O(\epsilon_n(K))$.

Corollary 4.1 (Convergence to ν^*)**.** *Suppose the measures μ_j are supported on K, with μ_1 absolutely continuous with respect to volume. Let ν^* be the unique minimizer of F. Then we can choose support points $\{y_i\}_{i=1,\dots,n}$ such that some subsequence of $\nu_n^* = \sum_{i=1}^n w_i^* \delta_{y_i}$ converges weakly to ν^*.*

Proof. By Theorem 4.1, we can choose support points so that $F[\nu_n^*] \to F[\nu^*]$. By compactness, the sequence ν_n^* admits a convergent subsequence $\nu_{n_k}^* \to \nu$ for some measure ν. Continuity of F allows us to pass to the limit $\lim_{k\to\infty} F[\nu_{n_k}^*] = F[\lim_{k\to\infty} \nu_{n_k}^*]$. On the other hand, $\lim_{k\to\infty} F[\nu_{n_k}^*] = F[\nu^*]$, and F is strictly convex Kim and Pass [2017], thus $\nu_{n_k}^* \to \nu^*$ weakly. □

Before proving Theorem 4.1, we need smoothness of the barycenter functional F with respect to Wasserstein-2 distance:

Lemma 4.1. *Suppose we are given measures $\{\mu_j\}_{j=1}^J$, ν, and $\{\nu_n\}_{n=1}^\infty$ supported on K, with $\nu_n \to \nu$. Then, $F[\nu_n] \to F[\nu]$, with $|F[\nu_n] - F[\nu]| \leq 2D \cdot W_2(\nu_n, \nu)$.*

Proof of Theorem 4.1. Uniqueness of ν^* follows from Theorem 2.4 of [Kim and Pass, 2017]. From Theorem 5.1 in [Kim and Pass, 2017] we know further that ν^* is absolutely continuous with respect to volume.

Let $N > 0$, and let ν_N be the discrete distribution on N points, each with mass $1/N$, which minimizes $W_2(\nu_N, \nu^*)$. This distribution satisfies $W_2(\nu_N, \nu^*) \leq CN^{-1/d}$ [Kloeckner, 2012], where C depends on K, the dimension d, and the metric. With our "budget" of n support points, we can construct a $2\epsilon_n(K)$-cover as long as n is sufficiently large. Then define a distribution $\nu_{n,N}$ with support on the $2\epsilon_n(K)$-cover as follows: for each x in the support of ν_N, map x to the closest point x' in the cover, and add mass $1/N$ to x'. Note that this defines not only the distribution $\nu_{n,N}$, but also a transport plan between ν_N and $\nu_{n,N}$. This map moves N points of mass $1/N$ each a distance at most $2\epsilon_n(K)$, so we may bound $W_2(\nu_{n,N}, \nu_N) \leq N \cdot 1/N \cdot 2\epsilon_n(K) = 2\epsilon_n(K)$. Combining these two bounds, we see that

$$\begin{aligned} W_2(\nu_{n,N}, \nu^*) &\leq W_2(\nu_{n,N}, \nu_N) + W_2(\nu_N, \nu^*) && (17) \\ &\leq 2\epsilon_n(K) + CN^{-1/d}. && (18) \end{aligned}$$

For each n, we choose to set $N = n$, which yields $W_2(\nu_{n,n}, \nu^*) \leq 2\epsilon_n(K) + Cn^{-1/d}$. Applying Lemma 4.1, and recalling that ν^* is the minimizer of J, we have

$$F[\nu_{n,n}] - F[\nu^*] \leq 2D \cdot (2\epsilon_n(K) + Cn^{-1/d}) = O(\epsilon_n(K) + n^{-1/d}). \tag{19}$$

However, we must have $F[\nu_n^*] \leq F[\nu_{n,n}]$, because both are measures on the same n point $2\epsilon_n(K)$-cover, but ν_n^* has weights chosen to minimize J. Thus we must also have

$$F[\nu_n^*] - F[\nu^*] \leq F[\nu_{n,n}] - F[\nu^*] \leq O(\epsilon_n(K) + n^{-1/d}).$$

□

The high-level view of the above result is that choosing support points y_i to form an ϵ-cover with respect to the metric g, and then optimizing over their weights w_i via our stochastic algorithm, will give us a consistent picture of the behavior of the true barycenter. Also note that the proof above requires an ϵ-cover only of the support of v^*, not all of K. In particular, an ϵ-cover of the convex hull of the supports of μ_j is sufficient, as this must contain the barycenter. Other heuristic techniques to efficiently focus a limited budget of n points only on the support of ν^* are advantageous and justified.

While Theorem 4.1 is a good start, ideally we would also be able to provide a bound on $W_2(\nu_n^*, \nu^*)$. This would follow readily from sharpness of the functional $F[\nu]$, or even the discrete version $F(w)$, but it is not immediately clear how to achieve such a result.

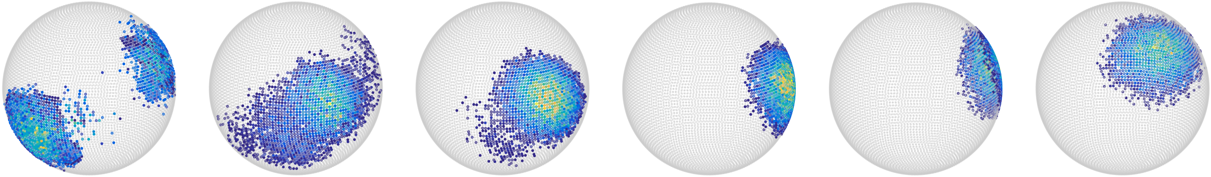

Figure 1: The Wasserstein barycenter of four von Mises-Fisher distributions on the unit sphere S^2. From left to right, the figures show the initial distributions merging into the Wasserstein barycenter. As the input distributions are moved along parallel paths on the sphere, the barycenter accurately tracks the new locations as shown in the final three figures.

5 Experiments

We demonstrate the applicability of our method on two experiments, one synthetic and one performing a real inference task. Together, these showcase the positive traits of our algorithm: speed, parallelization, robustness to non-stationarity, applicability to non-Euclidean domains, and immediate performance benefit to Bayesian inference. We implemented our algorithm in C++ using MPI, and our code will be made available on Github. Full experiment details are given in Appendix C.

5.1 Von Mises-Fisher Distributions with Drift

We demonstrate computation and tracking of the barycenter of four drifting von Mises-Fisher distributions on the unit sphere S^2. Note that W_2 and the barycentric cost are now defined with respect to geodesic distance on S^2.

The distributions are randomly centered, and we move the center of each distribution 3×10^{-5} radians (in the same direction for all distributions) each time a sample is drawn. A snapshot of the results is shown in Figure 1. Our algorithm is clearly able to track the barycenter as the distributions move.

5.2 Large Scale Bayesian Inference

We run logistic regression on the UCI skin segmentation dataset Bhatt and Dhall. There are 245057 datapoints which are colors represented in $\mathbb{R}^3$, each with a binary label determing whether that color is a skin color. We split consecutive blocks of the dataset into 127 subsets, and due to locality in the dataset, the data in each subsets is *not* identically distributed. Each subset is assigned one thread of an InfiniBand cluster on which we simultaneously sample from the subset posterior via MCMC and optimize the barycenter estimate. This is in contrast to [Srivastava et al., 2015a], where the barycenter can be computed via linear program (LP) only after all the samplers are run.

Since the full dataset is tractable, we can compare the two methods via W_2 distance to the posterior of the full dataset, which we can estimate via the large-scale optimal transport algorithm in [Genevay et al., 2016]. For each method, we fix n barycenter support points on a mesh determined by samples from the subset posteriors. After 283 seconds, or about 3000 iterations per subset posterior, our algorithm has produced a barycenter on $n = 10^6$ support points with W_2 distance about 22 from the full posterior. Moreover, no individual 16 thread node used more than 2GB of memory.

Table 1: Number of support points n versus computation time and W_2 distance to the true posterior. Compared to prior work, our algorithm handles much finer meshes, producing much better estimates.

	Linear program from [Srivastava et al., 2015a]							This paper
n	20	30	50	100	200	300	500	10^6
time (s)	0.5	1.5	3.1	16	67	169	out of memory	283
W_2	67.9	34.1	34.1	64.8	47.1	34.1	out of memory	22.2

In comparsion, in Table 1 we attempt to compute the barycenter LP as in Srivastava et al. [2015a] via Mosek [ApS, 2017], for varying values of n. Even $n = 500$ is not possible on a system with 16GB of memory, and feasible values of n result in meshes too sparse to accurately and reliably approximate the barycenter. Specifically, when n increases from 50 to 100, the approximation quality actually decreases: the subset posteriors are spread far apart, and the barycenter is so small relative to the required grid size that covering the barycenter well is a matter of luck. Entropy regularized methods may have faired better than the LP for finer meshes but would still not give the same result as our method. Note also that the LP timings include only optimization time, whereas in 283 seconds our algorithm produces samples *and* optimizes.

6 Conclusion and Future Directions

We have proposed an original algorithm for computing the Wasserstein barycenter of arbitrary measures given a stream of samples. Our algorithm is communication-efficient, highly parallel, easy to implement, and has immediate impact in large-scale Bayesian inference and sensor fusion tasks. For Bayesian inference in particular, we obtain far finer estimates of the Wasserstein-averaged subset posterior (WASP) [Srivastava et al., 2015a] than was possible before, enabling faster and more accurate inference.

There are many directions for future work: we have barely scratched the surface in terms of new applications of large-scale Wasserstein barycenters, and there are still many possible algorithmic improvements. One implication of Theorem 3.1 is that a faster algorithm for solving the concave problem (11) immediately yields faster convergence to the barycenter. Incorporating variance reduction [Defazio et al., 2014; Johnson and Zhang, 2013] is a promising direction, provided we maintain communication-efficiency. Recasting problem (11) as distributed consensus optimization [Nedic and Ozdaglar, 2009; Boyd et al., 2011] would further help scale up the barycenter computation to huge numbers of input measures.

References

M. Agueh and G. Carlier. Barycenters in the Wasserstein Space. *SIAM J. Math. Anal.*, 43(2): 904–924, January 2011. ISSN 0036-1410. doi: 10.1137/100805741.

Ethan Anderes, Steffen Borgwardt, and Jacob Miller. Discrete Wasserstein barycenters: Optimal transport for discrete data. *Math Meth Oper Res*, 84(2):389–409, October 2016. ISSN 1432-2994, 1432-5217. doi: 10.1007/s00186-016-0549-x.

Elaine Angelino, Matthew James Johnson, and Ryan P. Adams. Patterns of scalable bayesian inference. *Foundations and Trends® in Machine Learning*, 9(2-3):119–247, 2016. ISSN 1935-8237. doi: 10.1561/2200000052. URL http://dx.doi.org/10.1561/2200000052.

MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.0.0.53.*, 2017. URL http://docs.mosek.com/8.0/toolbox/index.html.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. 2017.

M. Baum, P. K. Willett, and U. D. Hanebeck. On Wasserstein Barycenters and MMOSPA Estimation. *IEEE Signal Process. Lett.*, 22(10):1511–1515, October 2015. ISSN 1070-9908. doi: 10.1109/LSP.2015.2410217.

J. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman Projections for Regularized Transportation Problems. *SIAM J. Sci. Comput.*, 37(2):A1111–A1138, January 2015. ISSN 1064-8275. doi: 10.1137/141000439.

Rajen Bhatt and Abhinav Dhall. Skin segmentation dataset. UCI Machine Learning Repository.

Jérémie Bigot, Elsa Cazelles, and Nicolas Papadakis. Regularization of barycenters in the Wasserstein space. *arXiv:1606.01025 [math, stat]*, June 2016.

Emmanuel Boissard, Thibaut Le Gouic, and Jean-Michel Loubes. Distribution's template estimate with Wasserstein metrics. *Bernoulli*, 21(2):740–759, May 2015. ISSN 1350-7265. doi: 10.3150/13-BEJ585.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. Streaming Variational Bayes. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1727–1735. Curran Associates, Inc., 2013.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/cheni14.html.

N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal Transport for Domain Adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, PP(99):1–1, 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2615921.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

Marco Cuturi and Arnaud Doucet. Fast Computation of Wasserstein Barycenters. pages 685–693, 2014.

Marco Cuturi and Gabriel Peyré. A Smoothed Dual Approach for Variational Wasserstein Problems. *SIAM J. Imaging Sci.*, 9(1):320–343, January 2016. doi: 10.1137/15M1032600.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pages 272–279. ACM, 2008.

Yoav Freund and Robert E. Schapire. Adaptive Game Playing Using Multiplicative Weights. *Games and Economic Behavior*, 29(1):79–103, October 1999. ISSN 0899-8256. doi: 10.1006/game.1999.0738.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a Wasserstein Loss. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2053–2061. Curran Associates, Inc., 2015.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic Optimization for Large-scale Optimal Transport. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3440–3448. Curran Associates, Inc., 2016.

Elad Hazan. Introduction to Online Convex Optimization. *OPT*, 2(3-4):157–325, August 2016. ISSN 2167-3888, 2167-3918. doi: 10.1561/2400000013.

Michael Held, Philip Wolfe, and Harlan P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88, December 1974. ISSN 0025-5610, 1436-4646. doi: 10.1007/BF01580223.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Matthew Johnson, James Saunderson, and Alan Willsky. Analyzing hogwild parallel gaussian gibbs sampling. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2715–2723. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5043-analyzing-hogwild-parallel-gaussian-gibbs-sampling.pdf.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Young-Heon Kim and Brendan Pass. Wasserstein barycenters over Riemannian manifolds. *Advances in Mathematics*, 307:640–683, February 2017. ISSN 0001-8708. doi: 10.1016/j.aim.2016.11.026.

Jun Kitagawa, Quentin Mérigot, and Boris Thibert. Convergence of a Newton algorithm for semi-discrete optimal transport. *arXiv:1603.05579 [cs, math]*, March 2016.

Benoît Kloeckner. Approximation by finitely supported measures. *ESAIM Control Optim. Calc. Var.*, 18(2):343–359, 2012. ISSN 1292-8119.

Bruno Lévy. A Numerical Algorithm for L2 Semi-Discrete Optimal Transport in 3D. *ESAIM Math. Model. Numer. Anal.*, 49(6):1693–1715, November 2015. ISSN 0764-583X, 1290-3841. doi: 10.1051/m2an/2015055.

C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of ∝n. *J Optim Theory Appl*, 50(1):195–200, July 1986. ISSN 0022-3239, 1573-2878. doi: 10.1007/BF00938486.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David Dunson. Scalable and Robust Bayesian Inference via the Median Posterior. In *PMLR*, pages 1656–1664, January 2014.

Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein Training of Restricted Boltzmann Machines. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3718–3726. Curran Associates, Inc., 2016.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Willie Neiswanger, Chong Wang, and Eric P. Xing. Asymptotically exact, embarrassingly parallel mcmc. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 623–632, Arlington, Virginia, United States, 2914. AUAI Press. ISBN 978-0-9749039-1-0. URL http://dl.acm.org/citation.cfm?id=3020751.3020816.

Arkadi Nemirovski and Reuven Y. Rubinstein. An Efficient Stochastic Approximation Algorithm for Stochastic Saddle Point Problems. In Moshe Dror, Pierre L'Ecuyer, and Ferenc Szidarovszky, editors, *Modeling Uncertainty*, number 46 in International Series in Operations Research & Management Science, pages 156–184. Springer US, 2005. ISBN 978-0-7923-7463-3 978-0-306-48102-4. doi: 10.1007/0-306-48102-2_8.

David Newman, Padhraic Smyth, Max Welling, and Arthur U. Asuncion. Distributed inference for latent dirichlet allocation. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1081–1088. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3330-distributed-inference-for-latent-dirichlet-allocation.pdf.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In *PMLR*, pages 2664–2672, June 2016.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein Barycenter and Its Application to Texture Mixing. In *Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, Berlin, Heidelberg, May 2011. doi: 10.1007/978-3-642-24785-9_37.

Sasha Rakhlin and Karthik Sridharan. Optimization, Learning, and Games with Predictable Sequences. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3066–3074. Curran Associates, Inc., 2013.

Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast Dictionary Learning with a Smoothed Wasserstein Loss. In *PMLR*, pages 630–638, May 2016.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Springer International Publishing, Cham, 2015. ISBN 978-3-319-20827-5 978-3-319-20828-2. doi: 10.1007/978-3-319-20828-2.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge university press, 2014.

Ohad Shamir. Without-replacement sampling for stochastic gradient methods. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 46–54. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6245-without-replacement-sampling-for-stochastic-gradient-methods.pdf.

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains. *ACM Trans Graph*, 34(4):66:1–66:11, July 2015. ISSN 0730-0301. doi: 10.1145/2766963.

Suvrit Sra. Directional Statistics in Machine Learning: A Brief Review. *arXiv:1605.00316 [stat]*, May 2016.

Sanvesh Srivastava, Volkan Cevher, Quoc Dinh, and David Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 912–920, San Diego, California, USA, 09–12 May 2015a. PMLR. URL http://proceedings.mlr.press/v38/srivastava15.html.

Sanvesh Srivastava, Cheng Li, and David B. Dunson. Scalable Bayes via Barycenter in Wasserstein Space. *arXiv:1508.05880 [stat]*, August 2015b.

Cédric Villani. *Optimal Transport: Old and New*. Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009. ISBN 978-3-540-71049-3. OCLC: ocn244421231.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

J. Ye, P. Wu, J. Z. Wang, and J. Li. Fast Discrete Distribution Clustering Using Wasserstein Barycenter With Sparse Support. *IEEE Trans. Signal Process.*, 65(9):2317–2332, May 2017. ISSN 1053-587X. doi: 10.1109/TSP.2017.2659647.

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015. URL http://jmlr.org/papers/v16/zhang15d.html.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

A Sparse Projections

Our algorithms for solving the barycenter problem in the parallel setting relied on the ability to efficiently project the matrix $A = (s, v^1, \ldots, v^J)$ back onto the constraint set $s = \sum_{j=1}^J v^j$. For the sake of completion, we include a proof that our sparse updates actually result in projection with respect to the ℓ_1 norm.

At any given iteration of gradient ascent, we start with some iterate $A = (s, v^1, \ldots, v^J)$ which does satisfy the constraint. Suppose we selected distribution j. The gradient estimate is a sparse $n \times (J+1)$ matrix M which has $M_{u1} = 1/J$ and $M_{vj} = -1$, for some indices u and v, with column 1 corresponding to s and column j corresponding to v^j. After the gradient step with stepsize γ, we have $A + \gamma M$. Now, our constraint can be written in matrix form as $Az = 0$, where

$$z = \begin{bmatrix} -1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{20}$$

and so the problem of projecting $A + \gamma M$ onto this constraint set can be written as

$$\begin{array}{ll} \min_B & \|A + \gamma M - B\|_1 \\ \text{s.t.} & Bz = 0. \end{array} \tag{21}$$

Equivalently, we want to find the matrix C solving

$$\begin{array}{ll} \min_C & \|C\|_1 \\ \text{s.t.} & (A + \gamma M + C)z = 0. \end{array} \tag{22}$$

Note that

$$(A + \gamma M + C)z = 0 \Leftrightarrow Cz = -\gamma Mz = \gamma\left(\frac{1}{J}e_u + e_v\right). \tag{23}$$

Consider the sparse matrix C given by $C_{u1} = -\gamma/(2J)$, $C_{uj} = \gamma/(2J)$, $C_{v1} = \gamma/2$, and $C_{vj} = -\gamma/2$. Define a sparse vector $\lambda \in \mathbb{R}^n$ by $\lambda_u = \lambda_v = -1$. We wish to show that the primal dual pair (C, λ) solves problem (22). We can do this directly by looking at the Karush–Kuhn–Tucker conditions. It is easy to check that C is primal feasible, so it remains only to show that

$$0 \in \partial_C(\|C\|_1 + \lambda^T Cz) \Leftrightarrow -z\lambda^T \in \partial_C(\|C\|_1). \tag{24}$$

The subgradients of the ℓ_1 norm at C are matrices G which satisfy $\|G\|_\infty \le 1$ and $\langle G, C\rangle = \|C\|_1$. It is easy to check that $\|z\lambda^T\|_\infty = 1$. Finally,

$$\begin{aligned} \langle -z\lambda^T, C\rangle = -\lambda^T Cz = -\gamma\lambda^T\left(\frac{1}{J}e_u + e_v\right) \quad (25) \\ = \gamma \cdot \left(\frac{1}{J} + 1\right) \quad (26) \\ = \|C\|_1. \quad (27) \end{aligned}$$

Hence after the gradient step we can project onto the feasible set with respect to ℓ_1, simply by adding the sparse matrix C.

B Stochastic Gradient Bound

We first need a lemma which gives a regret bound for online gradient ascent:

Lemma B.1 (Adapted from [Hazan, 2016, Theorem 3.1])**.** *Run online gradient ascent on concave functions f_t with subgradients $g_t \in \partial f_t(x_t)$. Assume $\|x_t - x^*\| \leq R$ for some optimizer x^* of $\sum_{t=1}^T f_t$, and assume $\mathbb{E}[\|g_t\|] \leq G$. Using stepsize $\gamma = \frac{R}{G\sqrt{T}}$, the expected regret after T iterations is bounded by $2RG\sqrt{T}$.*

Proof of Theorem 3.1. This is adapted from [Freund and Schapire, 1999; Rakhlin and Sridharan, 2013].

Define $f(s, v, w) = \langle s, w\rangle + \frac{1}{J}\sum_{j=1}^J \mathbb{E}_{X_j\sim\mu_j}[h(X_j, v^j)]$ as in (11). For simplicity, concatenate s and v into a vector u, with $f(u, w) = f(s, v, w)$. Write $w^*(u) = \operatorname{argmin}_{w\in\Delta_n}\langle s, w\rangle$ and note that our objective in Equation (11) is $f(u) := f(u, w^*(u))$.

Recall the online optimization setup: at time step t we play u_t, then receive f_t and reward $f_t(u_t)$, then update u_t and repeat. Note that if f_t is given by $f(u_t, w^*(u_t))$, then online gradient ascent on f_t is effectively subgradient ascent on f. Suppose we play online subgradient ascent and achieve average expected regret $\varepsilon(T)$ after T timesteps, where the expectation is with respect to the gradient estimates in the learning algorithm. Then by the definition of expected regret,

$$\varepsilon(T) \geq \mathbb{E}\left[\sup_u \frac{1}{T}\sum_{t=1}^T f_t(u) - \frac{1}{T}\sum_{t=1}^T f_t(u_t)\right] = \mathbb{E}\left[\sup_v \frac{1}{T}\sum_{t=1}^T f(u, w_t) - \frac{1}{T}\sum_{t=1}^T f(u_t, w_t)\right]. \tag{28}$$

where we write $w_t = w^*(u_t)$. Simultaneously, we have

$$\frac{1}{T}\sum_{t=1}^T f(u_t, w_t) - \inf_w \frac{1}{T}\sum_{t=1}^T f(u_t, w) \leq \frac{1}{T}\sum_{t=1}^T f(u_t, w_t) - \frac{1}{T}\sum_{t=1}^T f(u_t, w_t) = 0 \tag{29}$$

because w_t are each chosen optimally. Summing, we have

$$\mathbb{E}\left[\sup_u \frac{1}{T}\sum_{t=1}^T f(u, w_t) - \inf_w \frac{1}{T}\sum_{t=1}^T f(u_t, w)\right] \leq \varepsilon(T). \tag{30}$$

Now we merely need combine this with the standard bound:

$$\inf_w \frac{1}{T}\sum_{t=1}^T f(u_t, w) \leq \inf_w f(\overline{u}_T, w) \leq \sup_v \inf_w f(u, w) \tag{31}$$

$$\leq \inf_w \sup_u f(u, w) \leq \sup_u f(u, \overline{w}_T) \leq \sup_u \frac{1}{T}\sum_{t=1}^T f(u, w_t). \tag{32}$$

The extreme bounds on either side of this chain of inequalities are within $\varepsilon(T)$, hence we also have

$$\mathbb{E}\left[\sup_u f(u, \overline{w}_T) - \inf_w \sup_u f(u, w)\right] \leq \varepsilon(T). \tag{33}$$

By definition of f, the left hand side is precisely $\mathbb{E}[F(\overline{w}_T) - F(w^*)]$. Now, noting that our gradient estimates g are always sparse (we always have two elements of magnitude 1, so $\|g\|_1 = 2$), we simply replace $\varepsilon(T)$ with the particular regret bound of Lemma B.1 for online gradient ascent. □

C Experiment details

C.1 Von Mises-Fisher Distributions with Drift

The distributions are randomly centered with concentration parameter $\kappa = 30$. To verify that the barycenter accurately tracks when the input distributions are non-stationary, we move the center of each distribution 3×10^{-5} radians (in the same direction for all distributions) each time a sample is drawn. A snapshot of the results is shown in Figure 1.

We use a sliding window of $T = 10^5$ timesteps with step size $\gamma = 1$ and on $N = 10^4$ evenly-distributed support points. Each thread is run for 5×10^5 iterations on a separate thread of an 8 core workstation. The total time is roughly 80 seconds, during which our algorithm has processed a total of 2×10^6 samples. Clearly our algorithm is efficient and is able to perform the specified task.

C.2 Large Scale Bayesian Inference

Subset assignment. The skin segmentation dataset is given with positive samples grouped all together, then negative samples grouped together. To ensure even representation of positive and negative samples across all subsets, while simulating the non-i.i.d data setting, each subset is composed of a consecutive block of positive samples and one of negative samples.

MCMC chains. We used a simple Metropolis-Hastings sampler with Gaussian proposal distribution $\mathcal{N}(0, \sigma^2 I)$, for $\sigma = 0.05$. We used a very conservative 10^5 burn-in iterations, and afterwards took every fifth sample.

Mesh selection. During the burn-in phase, we compute a minimum axis-aligned bounding box containing all samples from all MCMC chains. Then, for a desired mesh size of n, we cut each axis into $n^{1/3}$ evenly-spaced values.

Optimization. We chose an aggressive step size of $\gamma = 500$ because of the large mesh size ($n = 10^6$). The 283 seconds value corresponds to 3×10^5 iterations total, or about 3000 per sampler. The barycenter estimate $\overline{w}_T = \frac{1}{T}\sum_{t=1}^{T} e_{i_t}$ was maintained over a sliding window of $T = 10^5$ timesteps.

Error metric. We stored 10^4 samples from the true posterior, and computed the W_2 distance between these samples and each candidate barycenter.