Optimization Course Project I:

Computing Wasserstein Barycenter via Linear Programming

In this project, we study the computation the Wasserstein barycenter of a set of discrete probability measures. Given support points of probability measures in a metric space and a transportation cost function (e.g. the Euclidean distance), Wasserstein distance defines a distance between two measures as the minimal transportation cost between them. Given a set of measures in the same space, the p-Wasserstein barycenter is defined as the measure minimizing the sum of p-Wasserstein distances to all measures in the set. Note that computing the barycenter of a set of discrete measures can be formulated by linear programming.

In this project, we focus on the case of $p = 2$ and compare the performance of different linear programming methods in solving the barycenter problem. We refer the notations and model setup to [2]. Instead of running experiments on MNIST dataset, we first restrict our attention to the algorithmic side and consider the following way in specifying the distributions $\mathcal{P}^{(t)}$.

- Generate m_t samples from normal distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ and construct $\mathcal{P}^{(t)}$ as the empirical distribution on the m_t samples.
- In the following experiments, you should vary the choice of the number of samples m_t, the number of distributions N, and the parameters (μ_t, σ_t^2).

In this way, the objective becomes finding the Wasserstein barycenter of N normal distributions. Additionally, we first focus on the case of *Pre-specified Support Problem* (See [2]) and choose the support of the barycenter distribution $\mathcal{P}$ be the union of the supports of $\mathcal{P}^{(t)}$'s.

- **Approach 1**: Implement any linear programming method for solving this problem, such as the PDLP method [1, 3]. Clearly state the linear program, include pseudo-code for your implementation, and explore/tune the parameters in the optimization algorithm such as step sizes.
- **Approach 2:** Implement the single low-rank regularization method (SLRM) and double low-rank regularization method (DLRM) developed in [2]. This method aims to reduce the cost of solving

the Newton equations in interior point method. How does this algorithm compare to the vanilla implementation of interior point method in Question 1? Plot the barycenter distribution $\mathcal{P}$ for two to three problem instances (specifications of N, m_t, etc.).

Now we consider the general *Free Support Problem* where the support of distribution $\mathcal{P}$ is also a decision variable. In [4] and [5], the authors proposed an entropy-smoothed version of Wasserstein distance for both regularization and computation purpose. The new distance replaces the summand $\langle D^{(t)}, \Pi^{(t)} \rangle$ in (3) and (4) in [2] with

$$\langle D^{(t)}, \Pi^{(t)} \rangle - \frac{1}{\lambda} h(\Pi^{(t)})$$

where $h(\cdot)$ is the entropy function. Intuitively, the entropy function will encourage the dispersion of the distribution $\Pi^{(t)}$ and avoid concentrations on a few points. Computationally, this new formulation enables a cheap computation of the gradients with respect to both the probability distribution parameters $(a_1, ..., a_m)$ and the support $\boldsymbol{X} = (\boldsymbol{q}_1, ..., \boldsymbol{q}_m)$ (in the language of [2]).

- **Approach 3**: Implement Algorithm 3 in [5] with different choice of the regularization parameter γ. How does the resultant barycenter distribution compare with the ones obtained from interior-point methods? Note that Algorithm 3 in [5] considers a free-support setting while the two proceeding questions consider a pre-specified support. For a fair comparison, you may implement the free-support version of the interior-point method in [2]. Specifically, it will alternate between solving a linear program for the distribution parameter $(a_1, ..., a_m)$ and solving a quadratic program for the support $\boldsymbol{X} = (\boldsymbol{q}_1, ..., \boldsymbol{q}_m)$. The quadratic program features for an analytical solution as (7) in [2].

  As noted in these papers, the free support problem is then a non-convex problem. Now we are interested in how the gradient-based algorithm (Algorithm 3 in [5]) compares to the interior point method in respect with escaping saddle points and local minima.

- **Approach 4**: Implement MAAIPM algorithm in [2] and compare it against Algorithm 3 in [5] in respect with the original objective function value

$$\sum_{t=1}^{N} \langle D^{(t)}, \Pi^{(t)} \rangle.$$

  While implementing Algorithm 3 in [5], you may want to periodically increase the regularization parameter γ to mitigate the effect of the additional penalty term. Please report the runtime, the number of iterations, and the objective value under both algorithms.

- **Approach 6**: Now you may migrate the experiments to the MNIST dataset[1] and Fashion MNIST dataset[2]. The advantage of using these datasets is that it can provide a more meaningful visualization of the barycenter distribution. The computational experiments can be on CPU and/or GPU [3].

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.kaggle.com/zalando-research/fashionmnist

References

[1] David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. Mathematical Programming, 201(1-2):133–184, September 2023. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-022-01901-9. URL https://link.springer.com/10.1007/s10107-022-01901-9.

[2] Dongdong Ge, Haoyue Wang, Zikai Xiong, Yinyu Ye. Interior-Point Methods Strike Back: Solving the Wasserstein Barycenter Problem. https://arxiv.org/abs/1905.12895.

[3] Haihao Lu, Jinwen Yang, Haodong Hu, Qi Huangfu, Jinsong Liu, Tianhao Liu, etc. cuPDLP-C: A Strengthened Implementation of cuPDLP for Linear Programming by C language. arXiv preprint arXiv:2312.14832, 2023.

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems, 2013.

[5] Marco Cuturi, Arnaud Doucet. Fast computation of Wasserstein barycenters. Proceedings of the International Conference on Machine Learning 2014.