Efficient Approximation of p-Wasserstein distance Using the Sinkhorn Algorithm

Jorge A. Martinez

1. Introduction

1.1. Motivation

In optimal transport theory, a key goal is to define some notion of distance between two probability measures μ and ν, which would represent some minimum cost to transform one to the other (useful in areas such as computer vision, imagine comparing the similarity between two images). One such notion is the Wasserstein metric, expressed with finite p-moments as $W_p(\mu, \nu)$.

1.2. Description

The Sinkhorn algorithm approximates $W_p(\mu, \nu)$ by solving an entropy-regularized optimal transport problem. For context, consider the discretized measures μ and ν into vectors x and y respectively such that $|x| = n$ and $|y| = m$. Let M be the $n \times m$ matrix such that $M_{ij} = \|x_i - y_j\|^p$. This is the cost of moving probability mass from the x_i to the y_j. We want to find a transport plan P that minimizes the overall cost

$$C = P \cdot M$$

while satisfying the marginal constraints. Current methods for calculating this exactly are computational expensive, due to the need to solve high order LP problems. [1] instead redefines C by introducing an entropy term.

$$C = P \cdot M + \frac{1}{\lambda} \sum_{i,j} P_{ij} \log P_{ij}$$

where $\lambda > 0$ is the regularization parameter. This allows us to express P as...

$$P = \operatorname{diag}(u) e^{-\lambda M} \operatorname{diag}(v)$$

Here, u and v are iteratively updated. The updates use LSE to ensure numerical stability and to enforce the marginal constraints. At each iteration, the algorithm adjusts x and y until the changes fall below some desired δ, indicating that the approximate transport plan P adheres to the marginals μ and ν. In the end, C serves an entropy-regularized approximation of $W_p(\mu, \nu)$.

2. Validation Scheme

I mainly plan to generate discrete probability measures with simple analytical properties to start. For example, I'll work with low-dimensional tensors with simple histograms (such as Gaussian or uniform) so I can calculate an exact value for C to compare to. This will directly allow us to verify convergence, that the marginal constraints are respected, and that C converges to the expected value as the regularization parameter λ is tuned. As for data that's similar to real-world data, I'm still now sure what to pull from (perhaps MNIST).

3. Implementation

The implementation in Python using PyTorch or NumPy. The main goal will be to replicate the process outlined in [1]. Key steps include:

- Constructing the cost matrix M.
- Initializing $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$, which will eventually enforce the marginal constraints.
- Update u and v using LSE operations of the form

$$u \leftarrow \varepsilon \left(\log \mu - \mathrm{LSE}\left(\frac{-M + v}{\varepsilon} \right) \right)$$

$$v \leftarrow \varepsilon \left(\log \nu - \mathrm{LSE}\left(\frac{-M + u}{\varepsilon} \right) \right)$$

  where $\varepsilon > 0$ is a parameter related to the regularization strength. These updates are repeated until the maximum changes in u and v fall below a desired δ.

- Compute the transport plan P as described in the introduction.
- Approximate $W_p(\mu,\nu) \approx P \cdot M$.

4. Alternate Approaches

I plan to compare my algorithm against several methods, including traditional algorithms for classical LP problems, and other approximation methods, such as the Greenkhorn algorithm (or any method that can be trivially computed or imported).

5. Comparisons

For comparison, I'll mainly assess accuracy (evaluate how closely the entropy-regularized distance approximates $W_p(\mu,\nu)$ as the regularization parameter changes), wall clock time performance, and CPU time performance (likely just FLOPs). The expectation is that the Sinkhorn algorithm will demonstrate significant speed-ups while maintaining acceptable accuracy for larger, higher dimensional problems, but lack for smaller problems.

6. Bibliography

References

1. M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. https://papers.nips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf, 2013.