# Optimal Transport Based Domain Adaptation

**Yihan He**
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
yihanhe@andrew.cmu.edu

**Jiayin Xia**
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
jiayinx@andrew.cmu.edu

**Kerou Zhang**
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
kerouz@andrew.cmu.edu

**Tianxiang Lin**
Information Networking Institute
Carnegie Mellon University
Pittsburgh, PA 15213
tianxian@andrew.cmu.edu

**Jiacheng Zhu***
Mechanical Engineering, Ph.D
Carnegie Mellon University
PIttsburgh, PA 15213
jzhu4@andrew.cmu.edu

**Hanjiang Hu**†
Mechanical Engineering, Ph.D
Carnegie Mellon University
PIttsburgh, PA 15213
hanjianghu@andrew.cmu.edu

## Abstract

Recently, adapting or transferring knowledge across different datasets or domains has gained increasing interests. Different learning methods include domain adaptation, transfer learning, meta learning and few/zero-show learning. Among the many strategies proposed to adapt a domain to another, finding a common representation has shown excellent properties: by finding a common representation for both domains, a single classifier can be effective in both and use labelled samples from the source domain to predict the unlabelled samples of the target domain. In this project, we focus on a large-scale optimal transportation model to perform the alignment of the representations in the source and target domains. First, we learn an optimal transport (OT) plan. To that end, we use a stochastic dual approach of regularized OT, which enables OT scale to large datasets. Second, we estimate a *Monge map* as a deep neural network learned by approximating the barycentric projection of the previously-obtained OT plan. This parameterization allows generalization of the mapping outside the support of the input measure. The source code for this project can be obtained by this github repo: `https://github.com/yihhhh/OT-for-Domain-Adaptation`.

## 1 Introduction

In modern data analytics, data used for learning a decision function and those used for inference do not necessarily follow the same distribution based on various applications. For example, in robotic and autonomous driving applications, domain drifts may occur during the navigation data collection because of lighting conditions, dynamic obstacle changes, and device changes. Domain adaptation problems are developed to learn the discrepancy context from a source domain to a target domain which has different data PDFs [7].

*Mentor of the project
†Mentor of the project

This project works on unsupervised domain adaptation[1] that data lables are only available in the source domain. It is performed under the assumption that the domain drifts can be reduced if data undergo a phase of non-linear mapping where both domains are more similar. The project focuses on the approach that has minimal transformation cost or metric. It aims at computing a transformation of input data which matches the distributions of source and target while learning a new classifier from the transformed source. The algorithms we use is mainly proposed by this paper[9].

## 2 Literature Review

### 2.1 Background on Optimal Transport

**Monge problem** Let $\mathcal{X} \subseteq \mathbb{R}^{d_s}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_t}$ be two complete and separable metric spaces and $\mathcal{M}(\mathcal{Z})$ denote the space of probability measures over spaces $\mathcal{Z}$. Given two probability measures $\mu \in \mathcal{M}(\mathcal{X})$, $\nu \in \mathcal{M}(\mathcal{Y})$ and a cost function $c : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}^+$, the Monge problem consists in finding a Borel map, $T : \mathcal{X} \mapsto \mathcal{Y}$ between $\mu$ and $\nu$ that realizes the infimum

$$\inf_T \int_\Omega c(x, T(x))d\mu(x) \text{ subject to } T_\#\mu = \nu, \tag{1}$$

where $T_\#\mu$ denotes the push forward operator of $\mu$ by $T$. The existence of the optimal transport map T is not always guaranteed: the Monge problem is non-convex and often unfeasible, for example, when the support $\mu$ and $\nu$ are different number of Diracs. Monge's formulation can be improved by the following relaxation problem by Kantorovich [6].

**Kantorovich relaxation** Given $\mu \in \mathcal{M}(\mathcal{X})$, $\nu \in \mathcal{M}(\mathcal{Y})$ and a cost function $c : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}^+$, Kantorovich OT seeks a joint measure $\pi \in \Pi$ minimizing

$$W(\mu, \nu) := \inf_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y)d\pi(x, y). \tag{2}$$

Here, $\Pi$ is the set of couplings of $\mu$ and $\nu$ denoted by :

$$\Pi = \{\pi : \gamma^{\mathcal{X}}_\#\pi = \mu, \gamma^{\mathcal{Y}}_\#\pi = \nu\}, \tag{3}$$

where $\gamma^{\mathcal{X}}$, $\gamma^{\mathcal{Y}}$ are functions that project onto $\mathcal{X}$ and $\mathcal{Y}$ respectively. Note that the optimal coupling always exists, and the conditional probability distributions $\pi_{y|x}$ gives stochastic maps from $\mathcal{X}$ to $\mathcal{Y}$ and is considered as "one-to-many" version of the deterministic map of the Monge map.

The computation in high dimensions of the optimal transport is typically computationally intensive. A faster approximate solution was proposed by [4] as follows:

$$\inf_{\pi \in \Pi} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y)d\pi(x, y) + \lambda H(\pi). \tag{4}$$

Here, $H(\pi) = \int \pi(x, y) \log(\pi(x, y))\mathrm{d}x\mathrm{d}y$ is the negative entropy function of $\pi$.

**Regularized OT** To improve the computational efficiency, regularization is included in the calculation of OT. It is achieved by adding a negative-entropy penalty to the Kantorovich Relaxation problem.

$$\inf_\pi \mathbb{E}_{(X,Y)\sim\pi}[c(X, Y)] + \varepsilon R(\pi) \text{ subject to } X \sim \mu, Y \sim \nu. \tag{5}$$

### 2.2 Domain adaptation as a transportation problem

Let $\Omega \in \mathbb{R}^d$ be an input measurable space of dimension $d$ and $\mathcal{C}$ the set of possible labels. $\mathcal{P}(\Omega)$ denotes the set of all probability measures over $\Omega$.
In a standard learning paradigm, one assumes a set of training data $X_s = \{x^s_i\}^N_{i=1s}$ is associated with a set of class labels $Y_s = \{y^s_i\}^N_{i=1s}$, with $y^s_i \in \mathcal{C}$, and a testing set $X_t = \{x^t_i\}^N_{i=1t}$ with unknown labels. In order to infer the set of labels $Y_t$ associated with $X_t$, one usually relies on an empirical estimate of the joint probability distribution $P(x, y) \in \mathcal{P}(\Omega \times \mathcal{C})$ from $(X_s, Y_s)$ and assumes that $X_s$ and $X_t$ are drawn from the same distribution $P(x) \in \mathcal{P}(\Omega)$. However, domain drifts between $X_s$ and $X_t$ exists for multiple reasons in the real world applications.
In domain adaptation problems, we denote a source domain and a target domain as $\Omega_s$ and $\Omega_t$. In some studies, domain drift is considered to be caused by an unknown, possibly nonlinear transformation of

the input space $\mathrm{T} : \Omega_s \to \Omega_t$. This transformation may have a physical interpretation (e.g. sensor drifts, thermal noise. etc). Additionally, an assumption are made that the transformation preserves the conditional distribution, i.e.

$$P_s(y|x^s) = P_t(y|T(x^s))$$

This means that the label information is preserved by the transformation. Assume the existence of two distinct joint probability distributions $P_s(x^s, y)$ and $P_t(x^t, y)$ that are related to the source and target domains and represent their corresponding marginal distributions over X are $\mu_s$ and $\mu_t$, a principled way to solve the adaptation problem can be formulated as follows:
1) Estimate $\mu_s$ and $\mu_t$ from $\mathrm{X}_s$ and $\mathrm{X}_t$;
2) Find a transport map T from $\mu_s$ to $\mu_t$;
3) Use T to transport labeled samples $\mathrm{X}_s$ and train a classifier from them.
Searching for T in the space of all possible transformations is intractable, and some restrictions need to be imposed. Here, we propose that T should be chosen so as to minimize a transportation cost $C(\mathrm{T})$ expressed as:

$$C(\mathrm{T}) = \int_{\Omega_s} c(\mathrm{x}, \mathrm{T}(\mathrm{x}))d\mu(\mathrm{x}),$$

where the cost function $c : \Omega_s \times \Omega_t \to \mathbb{R}^+$ is a distance function over the metric space $\Omega$. $C(\mathrm{T})$ can be interpreted as the energy required to move a probability mass $\mu(\mathrm{x})$ from x to $\mathrm{T}(\mathrm{x})$.
The problem of finding such a transportation of minimal cost has already been investigated in the literature, for example, Sinkhorn's algorithm.

### 2.3 Large-Scale Optimal Transport

**Regularized OT dual** To apply OT to large-scale data, a dual stochastic approach is used on the regularized Kantorovich problem by letting $X \sim \mu$ and $Y \sim \nu$. The dual of regularized OT problems is computed through the Fenchel-Rockafellar's duality theorm, and both entropy regularization and L2 regularization are considered.

$$\sup_{u,v} \mathbb{E}_{(X,Y)\sim\mu\times\nu} \left[u(X) + v(Y) + F_\varepsilon(u(X), v(Y))\right] \tag{6}$$

where

$$F_\varepsilon(u(x), v(y)) = \begin{cases} -\varepsilon e^{\frac{1}{\varepsilon}(u(x)+v(y)-c(x,y))} & \text{(entropy reg.)} \\ -\frac{1}{4\varepsilon}(u(x) + v(y) - c(x,y))^2 + \left(L^2 \text{ reg. }\right) \end{cases}. \tag{7}$$

The regularization relaxes the hard constraint with a strictly convex regularizer R since it is hard to satisfy the hard constraint on u and v along gradient iterations. It is enforced smoothly through a penalty term which is concave with respect to (u, v).

**Primal-Dual Relationship** To recover the solution of the regularized primal problem (5), first-order optimality conditions of the Fenchel-Rockafellar's duality theorem can be used.

$$d\pi^\varepsilon(x,y) = H_\varepsilon(x,y)d\mu(x)d\nu(y) \text{ where } H_\varepsilon(x,y) = \begin{cases} e^{\frac{u(x)}{\varepsilon}} e^{-\frac{c(x,y)}{\varepsilon}} e^{\frac{v(y)}{\varepsilon}} & (\text{ entropy reg. }) \\ \frac{1}{2\varepsilon}(u(x) + v(y) - c(x,y))_+ & \left(L^2 \text{ reg. }\right) \end{cases} \tag{8}$$

**Algorithm** The Stochastic OT computation algorithm sums up large-scale OT. By stochastic gradient methods sampling from $\mu \times \nu$, the relaxed dual problem can be maximized. The dual variable is a n-dimensional vector under the discrete case so that optimization can be computed. When $\mu$ has a density, u is a function that must be parametrized for optimization. Hence, deep neural networks are used since they are universal function approximators.

### 2.4 Optimal Mapping Estimations

**Barycentric Projection** Barycentric projection $\bar{\pi}$, in which $\pi$ is a solution of the regularized OT problem, with respect to a convex cost $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is defined as

$$\bar{\pi}(x) = \arg\min_z \mathbb{E}_{Y\sim\pi(\cdot|x)}[d(z, Y)] \tag{9}$$

**Algorithm 1** Stochastic OT computation

**Inputs:** input measures $\mu$, $\nu$; cost function $c$; batch size $p$; learning rate $\gamma$.
Discrete case: $\mu = \sum_i a_i \delta_{x_i}$ and $u$ is a finite vector: $u(x_i) \overset{\text{def.}}{=} u_i$ (similarly for $\nu$ and $v$)
Continuous case: $\mu$ is a continuous measure and $u$ is a neural network (similarly for $\nu$ and $v$)
$\nabla$ indicates the gradient w.r.t. the parameters
**while** not converged **do**
  sample a batch $(x_1, \cdots, x_p)$ from $\mu$
  sample a batch $(y_1, \cdots, y_p)$ from $\nu$
  update $u \leftarrow u + \gamma \sum_{ij} \nabla u(x_i) + \partial_u F_\varepsilon(u(x_i), v(y_j)) \nabla u(x_i)$
  update $v \leftarrow v + \gamma \sum_{ij} \nabla v(y_j) + \partial_v F_\varepsilon(u(x_i), v(y_j)) \nabla v(y_j)$
**end while**

**Algorithm 2** Optimal map learning with SGD

**Inputs:** input measures $\mu$, $\nu$; cost function $c$; dual optimal variables $u$ and $v$; map $f_\theta$ parameterized as a deep NN; batch size $n$; learning rate $\gamma$.
**while** not converged **do**
  sample a batch $(x_1, \cdots, x_n)$ from $\mu$
  sample a batch $(y_1, \cdots, y_n)$ from $\nu$
  update $\theta \leftarrow \theta - \gamma \sum_{ij} H_\varepsilon(x_i, y_j) \nabla_\theta d(y_j, f_\theta(x_i))$
**end while**

With respect to the squared Euclidean cost, the Barycentric Projection is usually served to recover optimal maps from optimal transport plans [8] (9) calculates a pointwise value. It indicates that a finite number of points is needed for mapping estimations if $\mu$ is discrete. The Barycentric Projection is defined as a deep neural network since we'd like the map to be defined everywhere.

**Optimal Map Learning** The objective function below with respect to the parameter $\theta$ can be minimized by training a deep neural network. An estimation of the Barycentric Projection of a regularized plan that generalizes outside the support of $\mu$ can also be obtained.

$$\begin{aligned} \mathbb{E}_{X\sim\mu}\left[\mathbb{E}_{Y\sim\pi^\varepsilon(\cdot|X)}\left[d\left(Y, f_\theta(X)\right)\right]\right] &= \mathbb{E}_{(X,Y)\sim\pi^\varepsilon}\left[d\left(Y, f_\theta(X)\right)\right] \\ &= \mathbb{E}_{(X,Y)\sim\mu\times\nu}\left[d\left(Y, f_\theta(X)\right) H_\varepsilon(X, Y)\right] \end{aligned} \tag{10}$$

This objective function is minimized by stochastic gradient descent in Algorithm 2. The opposite Barycentric Projection can also be computed with respect to a convex cost by minimizing $\mathbb{E}_{(X,Y)\sim\mu\times\nu}\left[d(g(Y), X) H_\varepsilon(X, Y)\right]$ since the OT map is symmetric.

## 3 Methodology

The computational framework is illustrated in 1 and follows two basic step: 1) learn an optimal map between the source and target distribution with Alg. 1 and Alg. 2; 2) map the source samples and train a classifier on them in the target domain. Specifically, in the first step, an optimal transport plan is parameterized as neural networks and learned from seen data (Alg. 1). Then, the obtained solution are used for learning an estimate of the barycentric projection from source domain to target domain that generalizes to unseen data using a neural network (Alg. 2).

## 4 Numerical Examples

To illustrate the behavior of the above optimal transport base method applied to DA problem, we use two toy examples, double moons and double circles. In the double moons case, the simulated datasets consists of two domains: the standard two entangled moons data, where each moon is associated to a special class. In the two circles case, the simulated datasets

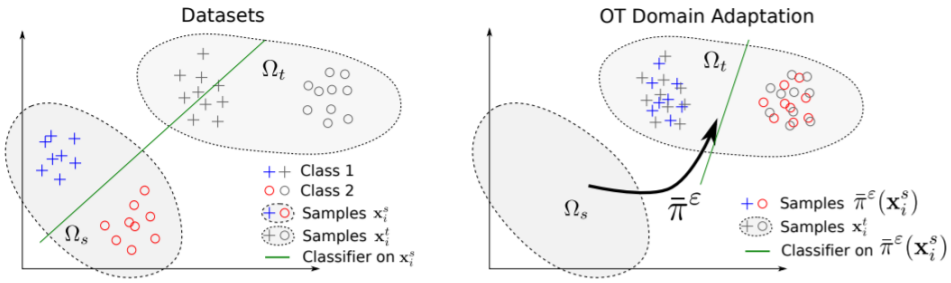
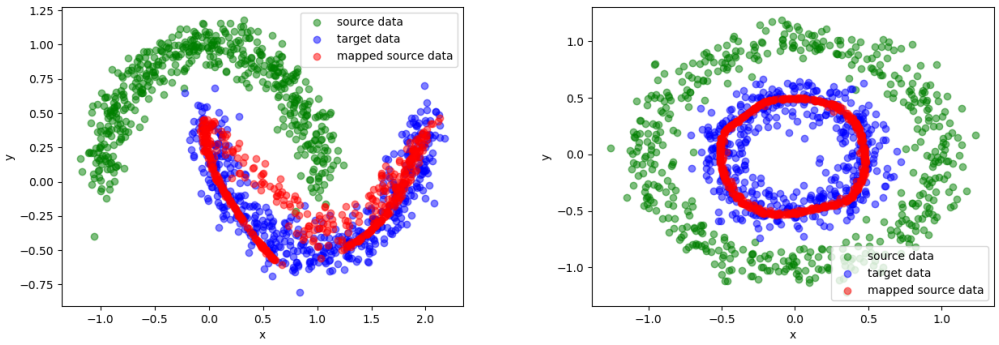



Figure 1: Pipeline of Optimal Transport based Domain Adaptation. [3]

consists of two domains: the circles data, where the two circles share the same center while have different radius. This two toy examples are notably interesting because the input dimensionality is small, 2, which leads to poor performance when applying methods based on subspace alignment.




(a) The standard two entangled moons data

(b) The circles data

Figure 2: Visualization of two Numerical Examples

For both examples, source domain and target domain are consist of 500 samples each. Besides, Gaussian noises with the standard deviation of 0.1 are added to data to simulate real world data. Results are shown in Fig. 2. We can observe that the mapped source data well preserves the shape of the target domain, which proves that the aforementioned method have the basic ability for domain adaptation task. In the following section, we use large datasets to evaluate its scalibility.

## 5 Experimental Setups

We apply the above method on an unsupervised domain adaptation (UDA) task. Our goal here is not to compete with the state-of-the-art methods in domain adaptation but to assess that this kind of computation framework allows to scale to optimal transport based domain adaptation (OTDA) to large datasets.

### 5.1 Datasets

We used two cross-domain digit image datasets MNIST and USPS. Both datasets have 10 classes of digits from 0 tos 9. 60000 samples in the MNIST domain and 7291 samples in the USPS domain are used for the adaptation. To keep consistency with USPS images, digit images from MNIST are

resized from the size of $28 \times 28$ to $16 \times 16$ by simple interpolation and downsampling. We consider MNIST dataset as the source domain and USPS as the target domain.

### 5.2 Baseline

Adaptation performance is evaluated using a 1-nearest neighbor (1-NN) classifier because it has the advantage of being parameter free and allows better assessment of the quality of the adapted representation (Courty et al., 2017b). In our experiments, we consider the 1-NN classification as a baseline, which is trained with the adapted source data and evaluated over the target data to provide a classification accuracy score. Here, 60000 samples from MNIST training set are used for training and xxx samples from USPS are used for testing.

### 5.3 Hyper-parameters

There are several sets of important hyper-parameters in the experiments. To narrow down our search space, we simply use multi-layer perceptron for both dual variable estimation and barycentric projection estimation networks so that the only parameter we need to tune is the number of hidden layers. Note that the size of the hidden layer are set to be symmetrical that first expand the dimension of the feature space and then gradually scale it down, both with a factor of 2. For instance, the hidden sizes of a 4-hidden-layer network are $(d \to 2d \to 2d \to d)$, the hidden sizes of a 5-hidden-layer network are $(d \to 2d \to 4d \to 2d \to d)$, where $d$ is the input dimension. ReLU are used as the activation function here. The value for the number of hidden layers of dual variable estimation is set in $\{3, 4, 5, 6, 7, 8, 9\}$, and that of barycentric projection estimation is set in $\{4, 5, 6, 7, 8\}$. Adam optimizer with batch size 1024 is used to optimize the network. The initial learning rate for Alg. 1 is varied in $\left\{2e^{-6}, 1e^{-5}, 2e^{-5}, 1e^{-4}, 2e^{-4}, 3e^{-4}\right\}$, while that for Alg. 2 is set to be $2e^{-4}$. Learning rate will be reduced by factor of 10 for every 100 epochs.
The regularization type and its corresponding scalar are also crucial hyper-parameters. For entropy-regularization, the scalar is chosen from $\{0.01, 0.1, 1, 10, 100\}$. For L2-regularization, the scalar is chosen from $\left\{10, 5, 1, 5e^{-1}, 1e^{-1}, 5e^{-2}, 1e^{-2}, 5e^{-3}, 1e^{-3}, 5e^{-4}, 1e^{-4}, 5e^{-5}, 1e^{-5}\right\}$. While we are tuning one hyper-parameter, other hyper-parameters keep fixed.

## 6 Results

Results on domain adaptation varied with number of hidden layers are provided in Table 1 and 2. We choose the network and learning rate with the best performance to carry on experiments with different regularization settings. Table 3 shows the experiment results with different regularization settings. From the above results, we can see that the test accuracy raise from 61.23% without adaptation to 70.45% by using the large-scale OTDA method.

We calculate the mean of digit images in each category from source data, mapped source data and target data. By visualize the mean images as presented in Fig. 3, we observe that for L2-regularization, with smaller scalar thus larger regularization value, the patterns of mapped source data become more distinct. Table 4 shows the sparsity of optimal transport plan, indicate that the regularization actually adds the sparsity of the optimal transport plan $\pi$. Note that the OT plan itself does not necessarily need to be sparse. However, when we apply it to applications like domain adaptation, we desire it to be sparse (or at least group-sparse) to enhance the interpretability, for that intuitively, samples in the source domain should be transferred towards samples in the target domain with the same labels.[2]

Moreover, the sparsity of the plan $\pi$ is positively correlated with the clarity of the mapped data pattern. However, a more distinct pattern in mapped source data does not lead to a better performance in adaption. In our case, the digit 4 and 7 are mapped in a way that is increasingly like 9 as regularization value grows. That may impede classifier from correctly learn the feature of 9, since inputs with features that supposed to be 9 are labeled with 4 or 7. A more detailed look into accuracy for each category provided in Figure 4 further confirm the above points. The accuracy 9 are much lower than that of other digits. One of the reason why 4 and 7 cannot be well adapted may be the inherent similarity among 4, 7 and 9 in the source domain. Digit 5 also shows bad performance in Figure

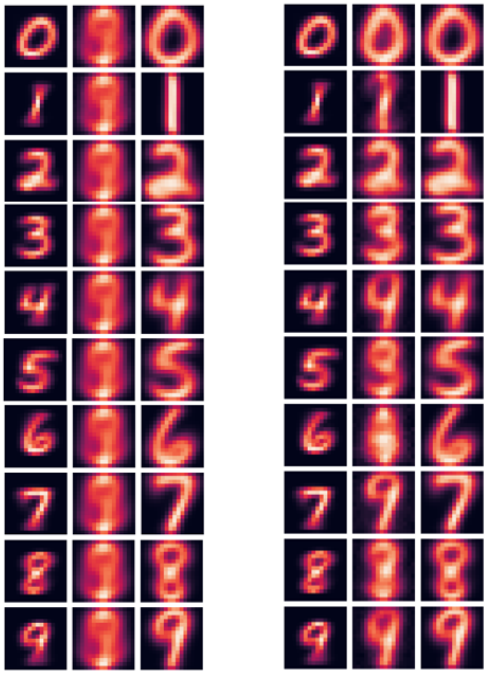



Figure 3: Visualization of mean images of each category in source data (the left column), mapped source data (the middle column) and target data (the right column) with two different L2 regularization setting

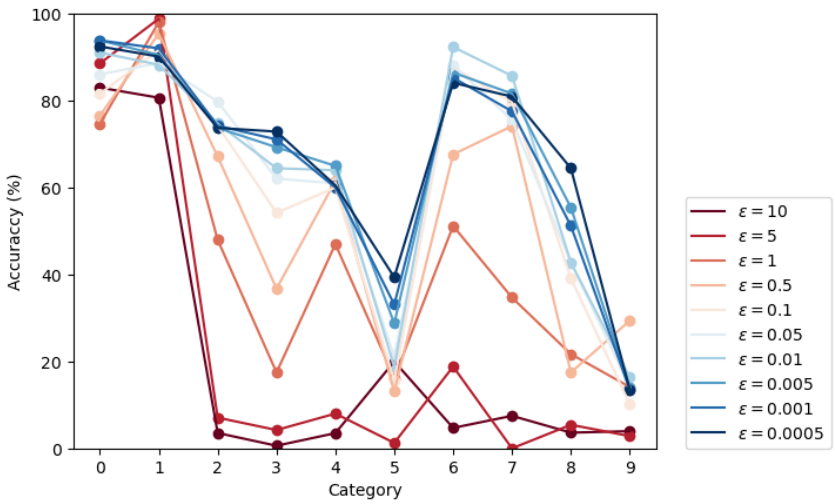



Figure 4: Test Accuracy for each digit category under different L2-regularization settings

Figure 4, this may also due to its inherent unclear feature pattern. Figure 5 is a visualization of the two dimensional features of different digits in MNIST extracted by PCA, which shows the little divergence among 4, 7 and 9, as well as the unclear feature pattern of digit 5. In this project, we did not do many feature processing, we believe extra feature extraction, i.e. using an autoencoder, would help address this issue.

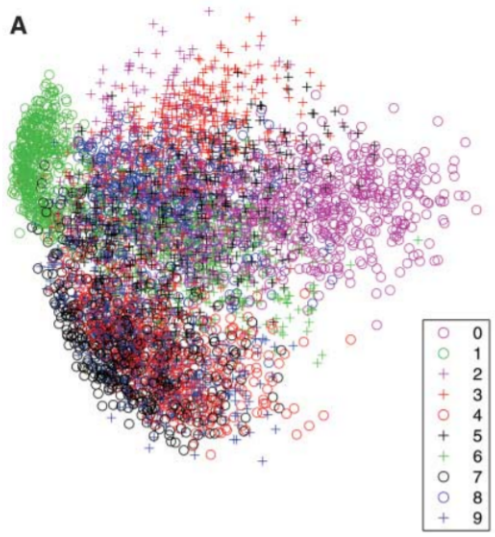



Figure 5: The two dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. [5]

Table 1: Results (accuracy in %) on domain adaptation from MNIST to USPS dataset with different number of hidden layers for dual variable estimation(Alg. 1). In the corresponding experiments, 5-hidden-layer MLP and a learning rate 2e-4 are used for barycentric projection estimation (Alg. 2). Learning rate for Alg. 1 is set to be 2e-4 here.

| Experiment | Test Accuracy (%) |
|---|---|
| w/o DA | 61.24 |
| # layers = 3 | 59.59 |
| # layers = 4 | 62.03 |
| # layers = 5 | 61.78 |
| # layers = 6 | 63.13 |
| # layers = 7 | **66.02** |
| # layers = 8 | 65.07 |
| # layers = 9 | 63.83 |

Table 2: Results (accuracy in %) on domain adaptation from MNIST to USPS dataset with different number of hidden layers for barycentric projection estimation (Alg. 2). In the corresponding experiments, 7-hidden-layer MLP and a learning rate 2e-5 are used for dual variable estimation (Alg. 1). Learning rate for Alg. 2 is set to be 2e-4 here.

| Experiment | Test Accuracy (%) |
|---|---|
| w/o DA | 61.24 |
| # layers = 4 | **69.06** |
| # layers = 5 | 68.96 |
| # layers = 6 | 63.02 |
| # layers = 7 | 55.3 |
| # layers = 8 | 68.46 |

Table 3: Results on domain adaptation from MNIST to USPS with entropy and L2 regularizations. In the corresponding experiments, 7-hidden-layer MLPs and a learning rate 2e-5 are used for Alg. 1, 4-hidden-layer MLPs and a learning rate 2e-4 are used for Alg. 2.

| **Experiment** | | **Test Accuracy(%)** |
|---|---|---|
| w/o DA | | 61.24 |
| L2-Regularization | $\varepsilon = 10$ | 29.40 |
| | $\varepsilon = 5$ | 33.08 |
| | $\varepsilon = 1$ | 48.33 |
| | $\varepsilon = 0.5$ | 58.30 |
| | $\varepsilon = 0.1$ | 63.08 |
| | $\varepsilon = 0.05$ | 65.62 |
| | $\varepsilon = 0.01$ | 67.56 |
| | $\varepsilon = 0.005$ | 69.61 |
| | $\varepsilon = 0.001$ | 69.06 |
| | $\varepsilon = 0.0005$ | **70.45** |
| | $\varepsilon = 0.0001$ | 63.98 |
| | $\varepsilon = 0.00005$ | 64.28 |
| | $\varepsilon = 0.00001$ | 56.70 |
| Entropy-regularization | $\varepsilon = 0.1$ | 32.74 |
| | $\varepsilon = 1$ | **67.91** |
| | $\varepsilon = 10$ | 41.31 |
| | $\varepsilon = 100$ | 16.89 |

Table 4: The sparsity of the optimal transport plan varied with different L2 regularizations. Here, the sparsity of a matrix is defined as the percetange of zero elements in a matrix.

| $\varepsilon$ | **Sparsity** |
|---|---|
| 10 | 0.0000 |
| 5 | 0.0000 |
| 1 | 0.1554 |
| 0.5 | 0.4020 |
| 0.1 | 0.8121 |
| 0.05 | 0.8958 |
| 0.01 | 0.9721 |
| 0.005 | 0.9847 |
| 0.001 | 0.9961 |
| 0.0005 | 0.9978 |
| 0.0001 | 0.9995 |
| 0.00005 | 0.9997 |
| 0.00001 | 1.0000 |

## 7 Conclusion

We implement two algorithms that allow for large-scale computation of regularized optimal transport and learning an optimal map that moves one probability distribution onto another. Experiments to investigate the effects of different hyper-parameters are carried out. This approach is the first tractable algorithms for computing both the regularized OT objective and optimal maps in large-scale or continuous settings. Though can be applied on domain adaptation, it is not currently adequate to compare with the state-of-the-art domain adaptation methods. To dig deeper, a lot of further work could be done, such as how to add known label information of target domain into current scheme to make fully use of the prior knowledge, how to adapt different type of data using this approach.

## References


[1] David Alvarez-Melis and Nicolò Fusi. "Geometric dataset distances via optimal transport". In: *arXiv preprint arXiv:2002.02923* (2020).

[2] Mathieu Blondel, Vivien Seguy, and Antoine Rolet. "Smooth and sparse optimal transport". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 880–889.

[3] Nicolas Courty et al. “Optimal transport for domain adaptation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.9 (2016), pp. 1853–1865.

[4] Marco Cuturi. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport”. In: *Advances in Neural Information Processing Systems*. 2013.

[5] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[6] Leonid Kantorovitch. “On the translocation of masses”. In: *Management Science* 5.1 (1958), pp. 1–4.

[7] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[8] Sebastian Reich. “A nonparametric ensemble transform method for Bayesian inference”. In: *SIAM Journal on Scientific Computing* 35.4 (2013), A2013–A2024.

[9] Vivien Seguy et al. “Large-scale optimal transport and mapping estimation”. In: *arXiv preprint arXiv:1711.02283* (2017).