Label Propagation via Diffusion on a Graph

Julian Schmitt, Stefano Ferioli, Camilla Beneventano

January 2024

Introduction

In many machine learning applications, labeled data is required to estimate a target function. However, obtaining such data is often time-consuming and very expensive. Semi-supervised learning methods aim to combine a small amount of labeled data (supervised), followed by a large amount of unlabeled data (unsupervised). In this project, we revisit a semi-supervised learning method presented in "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions" (Zhu et. al, 2003) [2] and provide further analysis of its performance compared to other "classical" methods.

In Section 1, we start by describing the *Harmonic Energy Minimization* (HEM) approach from [2]. Our primary objective is to clarify how the geometry of the data is exploited to profit from a large amount of unlabeled datapoints. Further, we explain how an external classifier can be incorporated and how hyperparameters can be optimized to improve the overall performance of the method.

In Section 2, we measure the accuracy of the predictions made by HEM in various classification problems. In particular, we compare its performance with other methods and investigate the impact of incorporating an external classifier. Last, we optimize hyperparameters graphically and via gradient descent for a binary classification problem.

1 The method

1.1 Harmonic Energy Minimization

We consider a binary classification problem with l labeled data points $\{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled data points $\{x_{l+1}, ..., x_{l+u}\}$ where $x_i \in \mathbb{R}^m \ \forall i \in [l+u]$ and $y_i \in \{0, 1\} \ \forall i \in [l]$. The unlabeled data points are classified using a weighted complete graph $G = (V, E)$ with $V = L \cup U$, $L = \{1, ..., l\}$ and $U = \{l+1, ..., l+u\}$. The weights w_{ij} are given by

$$w_{ij} = \exp\left(-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}\right) \quad \forall i, j \in [n], \tag{1}$$

where $n = l + u$ and $\sigma_1, ..., \sigma_m$ are length-scale hyperparameters for each dimension. This $n \times n$ symmetric weight matrix W is used to describe the geometric structure of the data.

The idea of the method is to find a real-valued function $f : V \to \mathbb{R}$ which can be used to assign labels to U. The function f is defined through the following optimization problem

$$f = \underset{f|_L = y}{\operatorname{argmin}} \ E(f) \quad \text{with} \quad E(f) := \frac{1}{2} \sum w_{ij}(f(i) - f(j))^2, \tag{2}$$

where $E(f)$ is a quadratic energy function. Intuitively, the known labels propagate on the graph similarly to how heat propagates on a conductive structure. In this analogy, the "temperature" is fixed on the labeled datapoints. As proven in Appendix A, (2) is equivalent to the Dirichlet problem for Laplace's equation

$$\begin{cases} f_i = y_i & \text{for } i \in L, \\ (\Delta f)_i = 0 & \text{for } i \in U, \end{cases}$$

where Δ is the combinatorial Laplacian. It holds $\Delta = D - W$, with $D := \operatorname{diag}(d)$ and $d_i = \sum_j w_{ij}\ \forall i \in [n]$. Therefore, the function f is harmonic and it holds

- Mean value property, i.e. $f_j = \frac{1}{d_j} \sum w_{ij} f_i = \sum P_{ij} f_i$ with $P = D^{-1}W$,
- Maximum principle, i.e. $f \in [\min y, \max y] \subseteq [0, 1]$.

Using these results, the solution of (2) is given by (for the proof see Appendix A)

$$f_U = (D_{uu} - W_{uu})^{-1} W_{ul} f_L = (I - P_{uu})^{-1} P_{ul} f_L \qquad \text{with} \qquad W = \left[\begin{array}{c|c} W_{ll} & W_{lu} \\ \hline W_{ul} & W_{uu} \end{array}\right].$$

In the following, we refer to the procedure above as *Harmonic Energy Minimization* (HEM). To translate the soft labels given by f into hard-labels, we present two methods:

- Harmonic Threshold (Thresh): $y_i = \mathbb{1}\{f_i > \frac{1}{2}\}$,
- Class Mass Normalization (CMN): $y_i = \mathbb{1}\left\{q \frac{f_i}{\sum_{j \in U} f_j} > (1-q) \frac{1-f_i}{\sum_{j \in U} (1-f_j)}\right\}$, where q and $1-q$ are desirable proportions for classes 1 and 0, respectively.

1.2 Extension to Multiclass Classification

Like most methods for binary classification, HEM can be extended for multiclass classification by using an one-vs-rest approach. For K sets of possible classes, we define $\tilde{y}^{(k)}$ as $\tilde{y}_i^{(k)} = \mathbb{1}\{y_i = k\}\ \forall k \in K, i \in [l]$. Based on these redefined labels, we obtain $f^{(k)}$ via HEM, as in the previous section. The methods to assign hard labels are generalized as follows:

- Harmonic Threshold (Thresh): $y_i = \underset{k \in K}{\operatorname{argmax}}\ f_i^{(k)}$,
- Class Mass Normalization (CMN): $y_i = \underset{k \in K}{\operatorname{argmax}}\ q_k \frac{f_i^{(k)}}{\sum_{j \in U} f_j^{(k)}}$.

1.3 Incorporation of External Classifiers

One can combine soft- or hardlabels $h_i\ \forall i \in [n]$ given via an external classifier with HEM by modifying the graph G as follows. We attach a "dongle" node i' labeled with h_i to each unlabeled node i. Further, we set a transition probability η from i' to i, and discount every other transition P_{ij} from i by $1-\eta$. From the mean value property, we have $f_j = \sum_i (1-\eta) P_{ij} f_i + \eta h_i$, that is equivalent to $f_U = (1-\eta)(P_{uu} f_U + P_{ul} f_L) + \eta h_U$ in matrix from. We obtain the solution formula $f_U = (I - (1-\eta) P_{uu})^{-1} ((1-\eta) P_{ul} f_L + \eta h_U)$. The hyperparameter η could be optimized via cross-validation on the labeled data.

1.4 Hyperparameter selection of σ_d

As suggested in [2], we use the minimum *average label entropy* as a heuristic criterion to obtain optimal hyperparameters $\sigma_1, ..., \sigma_m$ from the data. This entropy is defined as

$$H(f) = \frac{1}{u} \sum_{i \in U} H_i(f_i), \tag{3}$$

where $H_i(f_i) = -f_i \log f_i - (1-f_i) \log(1-f_i)$. Intuitively, $H(f)$ rewards values of f that are close to 0 or 1 and formalizes the idea of *confident* labeling. One problem arises from H reaching a minimum as $\sigma_d \to 0$ $\forall d \in [m]$. As the σ_d's approach zero, the weight function (1) is increasingly sensitive to the distance and therefore label predictions of unlabeled examples are dominated by their nearest neighbor's label. To exclude this undesired behaviour, we replace P with the smoothed $\tilde{P} = \varepsilon \mathcal{U} + (1-\varepsilon) P$ where $\mathcal{U}_{ij} = \frac{1}{n}$. This avoids the minimum to be selected in 0. The hyperparameter ε can be selected depending on the context, i.e. the behaviour of H with regard to the given data, as in Appendix D.

Note that the average label entropy does not consider the method used to obtain hard labels from f. As we will see in Section 2, the hyperparameters obtained via the minimum average label entropy are near-optimal only when using Thresh. This is not the case when labeling with CMN.

The average label entropy H can be minimized via gradient descent or grid search. Although gradient descent offers more guaranties, it turns out that $\frac{\partial H}{\partial \sigma_d}$ has a very "nested" structure and it is challenging to compute in a reasonable amount of time. In Appendix B, we give a detailed explanation how this can be achieved.

2 Experiments on the MNIST dataset

In this section, we evaluate the performance of HEM on the MNIST data set [1] which is a collection of 70,000 labeled 28x28 pixel grayscale images of handwritten digits, more experiments are presented in Appendix D. In particular, we solve several classification tasks and compare the predictions made by HEM to the following three other classification algorithms that do not make use of the geometric structure of the data.

- One-nearest-neighbour (1NN) labeling each test point based on its closest point in the training set.
- Radial-basis-function (RBF), that classifies each test point as $y_i = \mathbb{1}\{(W_{ul} f_l)_i > (W_{ul}(1 - f_l))_i\}$.
- Nadaraya-Watson with the Gaussian kernel (NW), that soft-labels each test point in $[0, 1]$ using a Gaussian kernel to weight nearby observations. To convert the soft-labels into hard-labels we use the the same methods we used for HEM: Thresh and CMN.

We remark that NW paired with Thresh is exactly the same as RBF when we use a single hyperparameter σ for all the dimensions. The proof is given in Appendix C. It is however still interesting, to see how NW performs when paired with CMN.

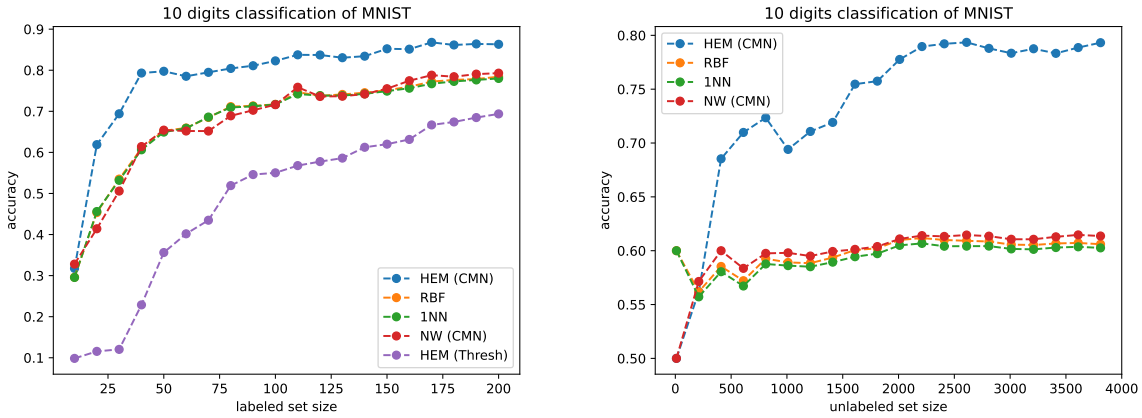

Figure 1: **10 digits classification.** (left) Varying labeled set size l with total set size $n = 4000$ and $\sigma_d = 2\ \forall d$. (right) Varying unlabeled set size u with labeled set size $l = 40$ and $\sigma_d = 2\ \forall d$.

In Figure 1 (left), we consider the multi-class problem of classifying digits "0" through "9" with $\sigma_d = 2\ \forall d$ for computing the weight matrix. First, we vary the labeled set size and compare the accuracies reached by HEM using Thresh as well as CMN with 1NN, RBF and NW. We report the results on a data set with intentionally unbalanced class sizes of 359, 460, 382, 429, 373, 408, 381, 426, 391, 391. For methods that incorporate class priors q, we estimate these from the corresponding proportion in the labeled set. HEM with Thresh performs poorly while HEM with CMN clearly outperforms RBF, 1NN and NW. Furthermore, it is noticeable that HEM with CMN reaches an accuracy of more than 80% already for a very small labeled set size of $l = 40$.

Although the setup of a varying labeled set size is useful as a baseline benchmark, it does not allow HEM to demonstrate its full potential. Recall that the strength of HEM is the incorporation of the geometry of the data and therefore we expect to perform better the larger the (unlabeled) set size. In contrast, the accuracy of 1NN, RBF as well as NW depends only on labeled data and does not change with the unlabeled set size. In Figure 1 (right), we fix the labeled set size to $l = 40$ with class sizes 3, 4, 3, 3, 7, 5, 3, 4, 4, 4 and vary the

unlabeled set size. The illustration confirms the theory. While the accuracy of HEM with CMN is increasing with the unlabeled set size, the results for 1NN, RBF and NW stay more or less constant.

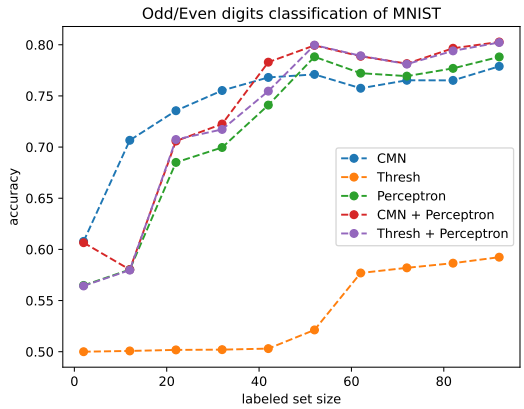

Figure 2: **Odd/Even digits classification.** Accuracy incorporating perceptron as an external classifier with labeled set size $l = 40$, total set size $n = 4000$, transition probability $\eta = 0.1$ and $\sigma_d = 2.5\ \forall d$.

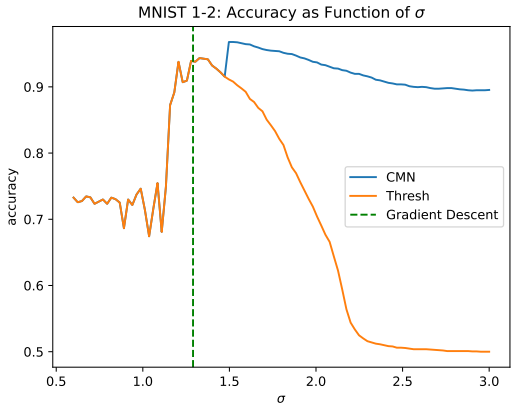

Figure 3: **Optimizing σ for 1-2 classification.** Accuracy of HEM w.r.t. σ for labeled set size $l = 100$ and total set size $n = 2200$. Green bar indicates optimal σ determined via gradient descent.

Next, we consider the problem of classifying odd vs. even digits, i.e. "1,3,5,7,9" and "2,4,6,8,0" are grouped into two classes, and incorporate a single-layer perceptron as an external classifier. In Figure 2, a balanced data set is used. The perceptron is trained for 10 epochs on the labeled set and then applied to the unlabeled set. As expected, the performance of the perceptron increases with the labeled set size and for $l \geq 50$ it gives better results than HEM. In contrast, HEM with CMN yields the best results for small labeled sets. HEM with Thresh once again reaches only low accuracies and is clearly outperformed by every other classifier. Both combinations, HEM with Thresh + perceptron and HEM with CMN + perceptron, show higher accuracies for large labeled set sizes than either method alone, suggesting there is complementary information used by each. Furthermore, we remark that the effect of complementary information was observed in [2] even for small labeled set sizes when using the voted perceptron instead of a single-layer perceptron.

Last, we use gradient descent to optimize the hyperparameter σ_d as suggested in Section 1.4. In Figure 3, we illustrate the accuracy of HEM for a fixed balanced set w.r.t. a single σ_d for all dimensions. We executed gradient descent for 1000 iterations with fixed learning rate $\gamma = 0.01$, starting value $\sigma_d = 2$ as well as $\varepsilon = 0.0001$ for smoothing and obtained $\sigma_d = 1.28$ as the optimal parameter (indicated by the green bar). For HEM with Thresh, this value is indeed near optimal. For HEM with CMN however, $\sigma_d = 1.28$ is only useful as an approximation of the best σ_d. Indeed, the entropy function uses f_u as label probabilities directly and does not incorporate class prior information.

3 Conclusion and final considerations

In summary, the Harmonic Energy Minimization (HEM) method from [2] enables an efficient utilization of the geometry of the data by introducing a Gaussian field matrix W, which captures pairwise distances of data points. Then, soft labels are determined by minimizing an energy function over W. The method naturally extends from binary to multiclass classification via a one-vs-rest approach and hyperparameters can be optimized using gradient descent.

Our experiments, conducted on the MNIST dataset [1], revealed the superiority of HEM compared to the traditional classifiers 1NN, RBF and NW. In particular, they demonstrated HEM's ability to profit from the geometric structure of the data and a large amount of unlabeled datapoints. Moreover, combining HEM with an external classifier can significantly improve results. Finally, we verified that optimizing the hyperparameters σ_d using an entropy function does lead to near-optimal results.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[2] X. Zhu, Z. Ghahramani, and J. Lafferty, *Semi-supervised learning using gaussian fields and harmonic functions*, Proceedings of the Twentieth International Conference on International Conference on Machine Learning, (2003), p. 912–919.

Appendix

A Solution of the minimization problem

We want to find $f : V \to \mathbb{R}$ that solves the constrained minimization problem

$$f = \underset{f|_L = y}{\operatorname{argmin}} E(f) \quad \text{with} \quad E(f) := \frac{1}{2} \sum w_{ij} (f(i) - f(j))^2.$$

Since V is finite, finding f is the same as finding a vector $f \in \mathbb{R}^{|V|}$, such that $f(i) \equiv f_i$ for $i \in V$. Having $f = \binom{f_L}{f_U}$, we can then rewrite the energy function as

$$\begin{aligned}
\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (f_i - f_j)^2 &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i f_j + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_j^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} f_i^2 \sum_{j=1}^{n} w_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i f_j + \frac{1}{2} \sum_{j=1}^{n} f_j^2 \sum_{i=1}^{n} w_{ij} \\
&= \sum_{i=1}^{n} f_i^2 d_i - \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i f_j \\
&= f^\top D f - f^\top W f \\
&= f^\top (D - W) f \\
&= \begin{bmatrix} f_L^\top & f_U^\top \end{bmatrix} \begin{bmatrix} D_{ll} - W_{ll} & -W_{lu} \\ -W_{ul} & D_{uu} - W_{uu} \end{bmatrix} \begin{bmatrix} f_L \\ f_U \end{bmatrix} \\
&= f_L^\top (D_{ll} - W_{ll}) f_L - f_L^\top W_{lu} f_U - f_U^\top W_{ul} f_L + f_U^\top (D_{uu} - W_{uu}) f_U,
\end{aligned}$$

that is convex since $D - W$ is a combinatorial Laplacian and therefore positive semi-definite. Setting the derivative w.r.t. f_U to 0, we find

$$-W_{ul} f_L + (D_{uu} - W_{uu}) f_U = 0 \quad \iff \quad (\Delta f)_i = 0 \quad \text{for } i \in U.$$

It follows that

$$f_U = (D_{uu} - W_{uu})^{-1} W_{ul} f_L.$$

B Finding the Hyperparameters σ_d via Gradient Descent

In this section, we explain how to compute the gradient of the average label entropy

$$H(f) = \frac{1}{u} \sum_{i \in U} -f_i \log f_i - (1 - f_i) \log(1 - f_i).$$

The d-th component of the gradient of H is computed as

$$\frac{\partial H}{\partial \sigma_d} = \frac{1}{u} \sum_{i=l+1}^{l+u} \log \left(\frac{1 - f(i)}{f(i)} \right) \frac{\partial f_i}{\partial \sigma_d},$$

where $\frac{\partial f_i}{\partial \sigma_d}$ is read from the vector $\frac{\partial f_U}{\partial \sigma_d}$. We have

$$\frac{\partial f_U}{\partial \sigma_d} = (I - \tilde{P}_{uu})^{-1} \frac{\partial \tilde{P}_{ul}}{\partial \sigma_d} f_L + (I - \tilde{P}_{uu})^{-1} \frac{\partial \tilde{P}_{uu}}{\partial \sigma_d} \underbrace{(I - \tilde{P}_{uu})^{-1} \tilde{P}_{ul} f_L}_{f_U} = (I - \tilde{P}_{uu})^{-1} \left(\frac{\partial \tilde{P}_{uu}}{\partial \sigma_d} f_U + \frac{\partial \tilde{P}_{ul}}{\partial \sigma_d} f_L \right).$$

Furthermore, $\frac{\partial \tilde{P}}{\partial \sigma_d} = (1-\varepsilon)\frac{\partial P}{\partial \sigma_d}$ and, since $P_{ij} = \frac{w_{ij}}{\sum_i w_{ij}}$, we have that

$$\frac{\partial P_{ij}}{\partial \sigma_d} = \frac{\frac{\partial w_{ij}}{\partial \sigma_d} - P_{ij}\sum_{k=1}^{l+u}\frac{\partial w_{ik}}{\partial \sigma_d}}{\sum_{k=1}^{l+u} w_{ik}}.$$

Last, it holds $\frac{\partial w_{ij}}{\partial \sigma_d} = 2w_{ij}\frac{(x_{id}-x_{jd})^2}{\sigma_d^3}$, since $w_{ij} = \exp\left(-\sum_{k=1}^{m}\frac{(x_{ik}-x_{jk})^2}{\sigma_k^2}\right)$.

Note that $\frac{\partial H}{\partial \sigma_d}$ is expressed via f and $\frac{\partial f_i}{\partial \sigma_d}$ which both require determining the matrix W. Therefore, W has to be computed in every step of the gradient descent when optimizing $\sigma_d\ \forall d$ and an efficient implementation is crucial for a reasonable performance. Since we executed gradient descent only for the case of a single σ for all dimensions, we focus on this case. Listing 1 illustrates a fast solution for computing W in Python. Observe that we rewrote $(X_{id}-X_{jd})^2 = (X_{id})^2 + (X_{jd})^2 - 2X_{id}X_{jd}\ \forall i,j \in [n]$ to leverage NumPy's broadcasting of arrays.

```python
import numpy as np

# Compute W with a single sigma for all dimensions
def W(X, sigma):
    X_norm = np.sum(X**2, axis=-1)
    return np.exp(-(1/sigma**2) * (X_norm[:,None] + X_norm[None,:] - 2 * np.dot(X,X.T)))
```

Listing 1: Computation of W

Another possible improvement would be to compute only one-half of W because the matrix is symmetric. This could further reduce the runtime by almost 50% (the diagonal is included in both the upper and lower halves of the matrix).

C Radial-basis-function VS Nadaraya-Watson

We prove here that the Nadaraya-Watson classifier with Gaussian kernel and Harmonic Thresh labeling is equivalent to the Radial-basis-function classifier from Section 2 when using a single hyperparamenter σ for all dimensions. Note that

$$w_{ij} = \exp\left(-\sum_{d=1}^{m}\frac{(x_{id}-x_{jd})^2}{\sigma^2}\right) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) = s(x_i, x_j).$$

It follows that

$$(W_{ul}f_L)_i > (W_{ul}(1-f_L))_i \quad \iff \quad \sum_{j=1}^{l} w_{ij}f_j > \sum_{j=1}^{l} w_{ij}(1-f_j)$$

$$\iff \quad \sum_{j=1}^{l} w_{ij}f_j > \frac{1}{2}\sum_{j=1}^{l} w_{ij} \quad \iff \quad \sum_{j=1}^{l}\frac{s(x_i,x_j)}{\sum_{j=1}^{l} s(x_i,x_j)}f_j > \frac{1}{2}.$$

D Additonal Experiments

In addition to Section 2, we present further experiments to investigate the properties of HEM. We start with Figure 4 in which we labeled two synthetic data sets using HEM. Observe that the algorithm gives the same labels to data points from the same point cloud (left) and the same spirals (right). In particular, HEM follows the geometry of the two data sets while "classical" methods such as 1NN, RBF and NW would fail to do so.

Next, we repeat the experiment from Figure 1 for the binary classification problem of distinguishing between the digit "1" and "2" on the MNIST data set [1]. Overall, we obtain very similar results to the 10 digits classification. For a varying labeled set size, HEM with Thresh gives bad results while HEM with CMN outperforms every other classifier. For a varying unlabeled set size, the performance of HEM with CMN

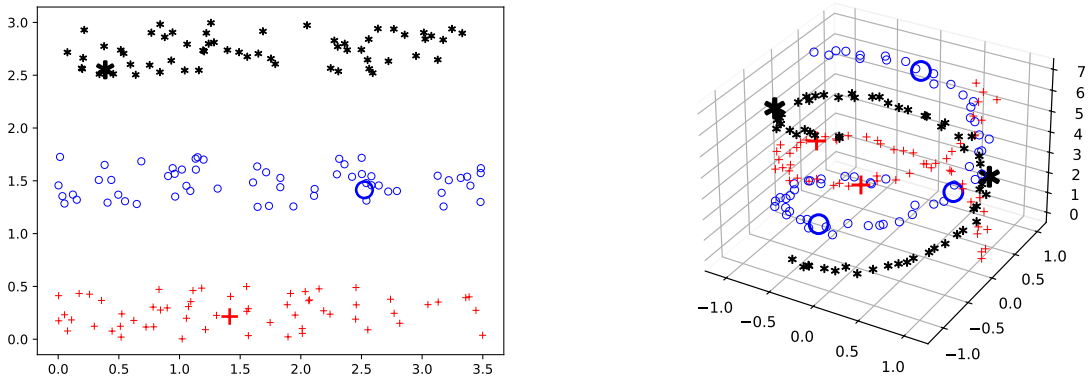

Figure 4: **3 sets classification.** HEM on synthetic data sets. Large symbols indicate labeled data, other points are unlabeled. (left) Three bands with $l = 3, u = 178$ and $\sigma_d = 0.22\ \forall d$. (right) Three spirals with $l = 7, u = 184$ and $\sigma_d = 0.43\ \forall d$.

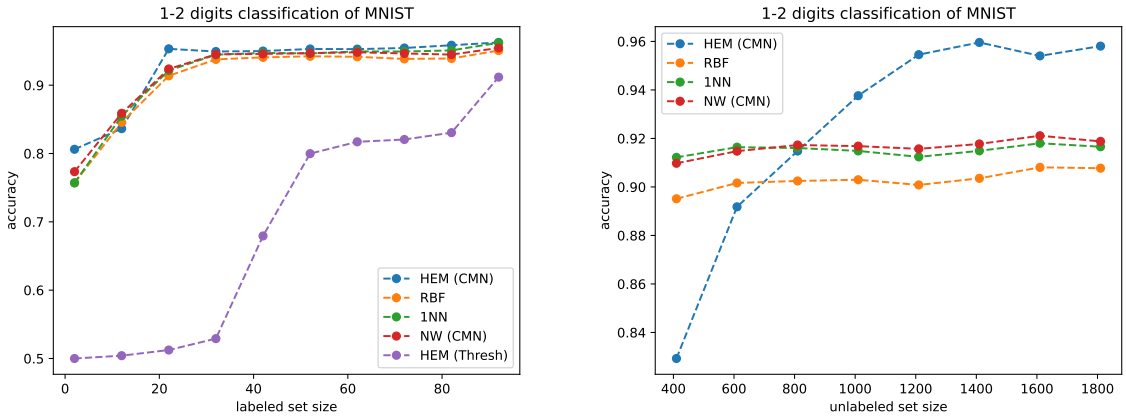

Figure 5: **1-2 digit classification.** (left) Varying labeled set size l of balanced set with size $n = 2200$ and $\sigma_d = 1.8\ \forall d$. (right) Varying unlabeled set size u with labeled set size $l = 40$ and $\sigma_d = 1.8\ \forall d$.

increases with the amount of data. In contrast, the accuracy of 1NN, RBF and NW with CMN stays the same no matter how large the unlabeled set is. Compared to the 10 digits classification, the differences between HEM and the other methods are significantly smaller and the obtained accuracies are generally higher. This is no surprise, as the 1-2 classification problem is much simpler.

Finally, we revisit the experiment from Figure 3 and investigate the effect of the smoothing parameter ε. Recall that we considered the 1-2 binary classification problem with a balanced data set ($l = 100, n = 2200$) obtained from MNIST [1] for optimizing a single hyperparamter σ which is used among all dimensions. In Figure 6, we plot the average label entropy $H(f)$ (3) w.r.t. σ. As mentioned in Section 1, the unsmoothed $H(f) \to 0$ for $\sigma \to 0$. However, as the smoothing factor ε increases, the "nuisance minimum" at 0 vanishes and $\varepsilon = 10^{-4}$ is already sufficient. Lastly, regarding the starting value for gradient descent, $\sigma = 2$ has proved to be a good choice for the smoothed $H(f)$ with $\varepsilon = 10^{-4}$ because it makes sure that the algorithm iterates towards the global minimum. For larger values of σ, gradient descent could get stuck in the local minimum around 3.

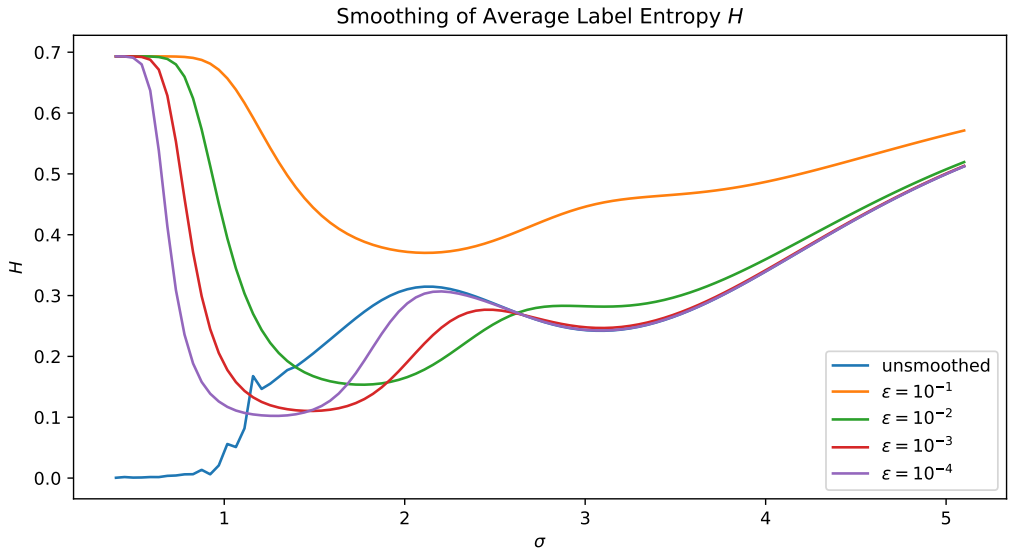

Figure 6: **Smoothing of Average Label Entropy.** $H(f)$ w.r.t. σ for $l = 100, n = 2200$ and different smoothing values ε.