Statistical Machine Learning
Label Propagation on a Graph

EPFL Department of Mathematics

Julia Bierent, Luca Nyckees, Blerton Rashiti

***Abstract*—In this project, we investigate a type of semi-supervised learning algorithm designed to efficiently tackle binary and multi-classification problems with a very low proportion of labeled data. More precisely, we introduce and study the functioning of label propagation via diffusion on a graph. We evaluate the introduced model by interpreting new results based on the problem of classifying hand-written digits. This work will strengthen the *a priori* supposition that this label propagation method proves promising when compared to other models, in the case where only a small amount of labeled data is available.**

I. Introduction

Various semi-supervised learning methods have proven to be very efficient in machine learning classification problems when facing a deficit in labeled data. Here, we study the case of label propagation via diffusion on a graph, for which the search for a classifier is based on a notion of energy minimization. We naturally build a weighted complete graph from our labeled and unlabeled data sets, where weights are assigned following a normalized multivariate Gaussian distribution principle. High weights (close to 1) mimic a high resemblance between points, and those are the ones we draw as shown in Figure 1. Moreover, we choose to make resembling points close to each other in the graph, to obtain Euclidean clusters. Energy minimization on the resulting graph is equivalent to having a harmonic solution, allowing us to express our classifier as an explicit function of labeled and unlabeled data under closed form.

The reason why label propagation on a graph works well when facing a deficit in labeled data is its ability to effectively make use of both unlabeled and labeled data. The richness of the model comes from the fact that the forming of the weighted graph takes into account the nature of geometric features appearing in the raw data, so that the importance of each feature is modularizable.

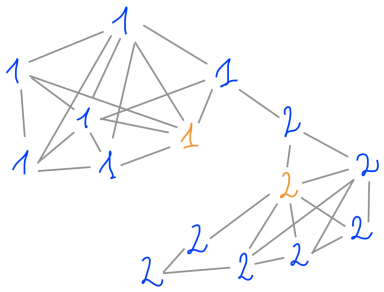


Figure 1: Representation weighted graph of hand-written *ones* and *twos* with only two labeled nodes. Blue nodes are unlabeled, whereas orange nodes are labeled.

In Section II, we mention our main reference, which is the paper initially proposing the method we investigate. In Section III, we introduce the main idea of the method, with the basic tools and concepts leading to the closed form solution, expressed in Section IV. This is followed by Section VI, where we propose refinements of the main method. More precisely, we talk about how to improve the classification step in the first subsection, and how to make use of an external classifier to increase model performance in the second one.

II. Related Works

The model we study is entirely based on the label propagation method initially introduced in the publication *Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions* of Xiaojin Zhu, Zoubin Ghahraman and John Lafferty. The complete reference can be found at [1]. The presented results are entirely computed on our part.

III. Mathematical Framework

The method takes as initial input a data set of l labeled and u unlabeled data, say respectively given by families $\{(x_i, y_i)\}_{i=1}^{l}$ and $\{x_i\}_{i=l+1}^{l+u}$, where $x_i \in \mathbb{R}^m$ are data points and y_i are labels, which are assumed to be binary, *i.e.* $y \in \mathbb{Z}/2\mathbb{Z}$. Let $n := l + u$ be cardinality of the full data set, and consider the graph (V, E) where $V = \{x_i | i = 1, ..., n\}$ so that nodes naturally decompose into a labeled subgraph L and an unlabeled subgraph U. The only *a priori* constraint on E is that the graph must be connected, but we choose it to be complete, *i.e.* $|E| = 2^{|V|}$. However, we usually only show edges whose assigned weight respects a certain threshold of significance when representing data with a graph, as in Figure 1. Those weights are determined via the expression below, where w_{ij} is the weight assigned to the edge connecting nodes i and j.

$$w_{ij} = \exp(-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}) \in]0, 1]$$

Here, node i represents a data point whose information is contained in the vector $x_i = (x_{i1}, ..., x_{im}) \in \mathbb{R}^m$, and the σ_d are parameters to be estimated so as to regulate the importance given to each direction $1 \leq d \leq m$. It is clear from the expression above that two very *similar* points will share a weight close to 1, whereas two *non-similar* points will share a weight close to 0. It is good to note that the method we present works with any other appropriate weight matrix, *i.e.* $(w_{ij})_{i,j}$ doesn't necessarily have to simulate Gaussian weights.

The main idea is to train a predictor $f : (V, E) \to \mathbb{R}$ (or $f : (V, E) \to [0, 1]$ in our case) that assigns a continuous value to each node, so that the final prediction can be made by combining this continuous prediction with a simple classification step, such as a threshold criterion. We decide to give f some obvious behavioral constraints. First, we impose that it agrees on the labeled set with the corresponding values y_i. Then, we follow the key idea of the method that clusters in the graph (when nearby points mean heavy shared weights) should contain points that are similar to each other, and thus should be assigned the same label. This motivates the classic energy minimization problem formulated below.

$$f = \text{argmin}_{f|_L = f_l} E(f), \text{ where}$$

$$E(f) = \sum_{i<j} w_{ij}(f(i) - f(j))^2$$

IV. Closed-form Solution

We define the *discrete Laplacian operator* as $\Delta = D - W$ where W is the previous weight matrix $(w_{ij})_{i,j} \in]0, 1]^{n \times n}$ and D is the diagonal matrix with diagonal entries $d_i = \sum_j w_{ij}$. The solution to the problem above satisfies the so-called *harmonic property* $\Delta f = 0$ on unlabeled points. (Proof in the Appendix.) Now, this condition means that $Df = Wf$ on U, *i.e.* $f_j = \frac{1}{d_j} \sum_{i=1}^{n} w_{ij} f_i$ for all $j \in \{l + 1, ..., l + u\}$. Since we attribute low weights to edges between distant points, we see that predictions on unlabeled points are obtained as approximations of local averages, which is exactly the type of property we desire.

The closed-form solution is expressed as

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l$$

where we consider the block structure given by

$$W = \begin{pmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{pmatrix}, f = \begin{pmatrix} f_l \\ f_u \end{pmatrix},$$

(similarly for the matrix D) with W_{uu} being the upper-left $u \times u$ submatrix of W and so on. (Proof in the Appendix.)

Finally, the simplest final classification step is a $\frac{1}{2}$ threshold criterion : node i is classified as 0 if $f(i) < \frac{1}{2}$ and as 1 otherwise.

The functioning of this solution is shown in Figures 2 and 3, where the synthetic data are two cosines separated by a shift, we want to classify the membership to one cosine to blue and to the other to red. We emphasize the fact that the method works best if the data are well separated.

V. Random Walk Analogy

One can get an intuition from the context of random walks. More precisely, we can imagine a particle moving accross the graph (V, E). Then it moves from a node i to an adjacent node j with a certain probability related to the energy of the edge. The probability to encounter a node labeled 1 first is then $f(i)$, *i.e.* the particle moves along one of the shortest paths leading to a labeled 1 node with probability $f(i)$.

VI. Refining the Model

We present two ways of improving model performance. The first one improves the execution of the final classification step, replacing the classic threshold criterion. The second one is a way of taking into consideration the opinion of another classifier, so as to take the best of both predictions and whose action is fairly intuitive.

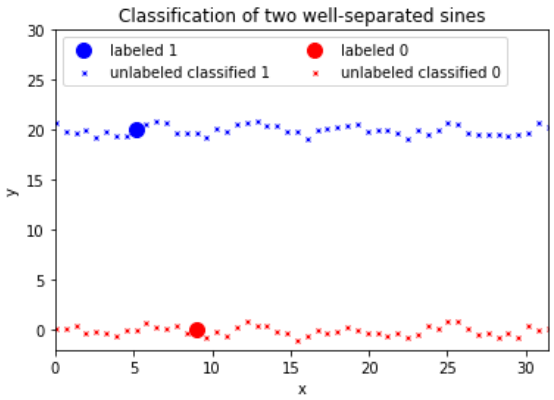


Figure 2: Demonstration of harmonic energy minimization on a synthetic data set (20-shifted cosine wave with noise). Crosses : labeled data points.

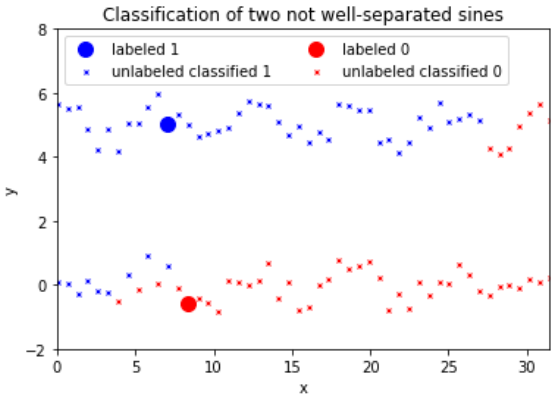


Figure 3: Demonstration of harmonic energy minimization on a synthetic data set (5-shifted cosine wave with noise). Crosses : labeled data points.

A. Class Mass Normalization

In Section V, the $\frac{1}{2}$ threshold means that the particle at node i is more likely to reach a node labeled as 1 first if $f(i) > \frac{1}{2}$. But if the data does not form separated clusters, then this method performs poorly. The following is a way to remedy this. We define the *mass* of class 1 as $m_1 = \sum_i f_u(i)$ and the one of class 0 to be $m_0 = \sum_i (1 - f_u(i))$. We follow the simple rule that a point i is classified as 1 if and only if the inequality below holds.

$$q \frac{f_u(i)}{m_1} > (1-q) \frac{1 - f_u(i)}{m_0}$$

Here, q is the estimated desired proportion of 1's. This process is known as *class mass normalization* (CMN). Note that if expected proportions q and $1-q$, and masses m_0 and m_1 are respectively equal, we get $\frac{1}{2}$ threshold.

B. External Classifier

Suppose we are given a classifier h that predicts labels $h_u(i)$ on unlabeled nodes i. The idea is to augment the graph (V, E) by adding a node i' with label $h_u(i)$ to each unlabeled node i on the original graph with an edge having transition probability η. Then, the sum of adjacent edge weights is decreased by $1-\eta$, so as to preserve total transition probability for each node. The hyperparameter η is estimated as follows.

$$\hat{\eta} = \text{argmax}_{\eta \in \{\eta_0, \dots, \eta_s\}} \{\text{accuracy}(f_u(\eta))\}$$

Then, we apply the energy minimization method to the augmented graph. Intuitively, what we do is create, for each unlabeled node, a kind of path between the latter and a new node encoding the opinion of the other classifier. What this does is to give a certain cost, regulated by η, that is paid when computing the energy loss function, and increases when we differ too much from the external classifier's prediction. All this boils down to the new prediction formula below, where $P = D^{-1}W$.

$$f_u = (I - (1-\eta)P_{uu})^{-1}((1-\eta)P_{ul}f_l + \eta h_u)$$

VII. Learning the Weights

The method's performance heavily relies on the weight matrix W, whose parameters σ_d have to be estimated from data. We don't apply the likelihood process simply because this would prevent us from using unlabeled data. Instead, we look at the Shannon entropy of f, defined as (with $f_i = f(i)$)

$$H(f) = \frac{1}{u} \sum_{i=l+1}^{l+u} \tilde{H}(f_i), \text{ where}$$

$$\tilde{H}(z) = -z \log(z) - (1-z) \log(1-z) \text{ for } z \in]0, 1[.$$

It's worthwhile mentioning that $f_u(i) \in]0, 1[$ is a property that follows from the maximum principle for harmonic functions, as explained in [2], and thus entropy is a well-defined function. The goal is to minimize this entropy, as it results in a confident prediction : the more the entropy is small, the more the obtained predictions are close to 0 or 1. Concretely, we use a gradient descent algorithm with iteration step computed as follows.

$$\sigma_d^{k+1} = \sigma_d^k - \gamma_k \frac{\partial H}{\partial \sigma_d}, \; d \in \{1, \dots, m\}$$

VIII. Performance Evaluation

In this section, we evaluate the accuracy of our model (energy minimization with CMN classification - we note that, as shown in Figure 11, the CMN method beats the threshold one) in confrontation with linear kernel support vector machine (SVM), logistic regression (logreg) and k nearest neighbors (kNN) methods. In particular, we are interested in how label propagation behaves in comparison to other classification mehtods when confronted with very few labeled data. We also show how the incorporating of those methods as external classifiers improves the accuracy. Results of this section are obtained with a MNIST data set of 100 images represented by 256 pixels each, that we generate and pre-process with functions offered by the `Keras` library. Class priors q are estimated with Laplace smoothing, and parameters η are chosen with the principle mentioned in Section VI. The k-nearest neighbors comparison is implemented with an optimization function that chooses, for each given labeled set size l, the integer $k \in \{2, ..., l\}$ that maximizes the accuracy via kNN, *i.e.* we make it so that kNN gives its best shot.

What is interesting is what happens in the small labeled set size setting, which is shown in Figures 4, 5 and 6. In all three graphs, the adversarial method seems to suffer the lack of labeled data, whereas label propagation handles it well. For example, the kNN method has a 0.55% accuracy when confronted with 2% and 3% labeled set proportions. Label propagation has a lower accuracy bound around 0.75%. We also note that the combining of label propagation with any of the three other methods leads to overall improved accuracies in almost all labeled set sizes : the green curve clearly dominates the other curves, sometimes improving the best accuracy of the two separate methods by more than 5% (e.g. in Figure 5, with $l = 5$). It is also worthwhile mentioning that every now and then, a pathological behaviour takes place when combining label propagation with one of the external classifiers considered, as in Figure 5, with $l = 6$.

In large labeled set sizes (10% to 35%), the label propagation method performs pretty well, but not better than the other methods we consider. This just reinforces the idea that energy minimization is specifically designed to tackle the lack of labeled data, but is not aimed at making perfect predictions when facing large labeled set sizes. This is shown in Figures 10 and 9, where we plotted the accuracies in large labeled set sizes of label propagation against (and combined with) SVM and logistic regression. We note that, as in the case of low labeled set sizes, the combining of label propagation with another method improves the overall accuracy (and there is no pathological behaviour here).

IX. Other Experimental Results

In Figures 7 and 8, we underline the importance of a heterogeneous σ, *i.e.* a weight matrix that captures the essence of each direction, or feature, in the data set. What really matters, to distinguish the two Euclidean clusters in this figure, is the y-dimension, whilst the x-dimension doesn't matter much. Indeed, assigning an infinite value to σ_x results in a much better prediction, since it boils down to not taking the feature x into account.

Figures 12 and 13 compare the performance of the methods of part VI.B when η is optimized and $\eta = 0.1$ (which is used in the paper [1]). The optimized η is the one that maximizes, in some $\{\eta_0, \ldots, \eta_s\} \subset [0, 1]$, the accuracy of the classifier, which is trained on the labeled data ("training set") and is evaluated on the unlabeled data ("test set"). In Figure 12, the external classifier used is the SVM; we observe that with η optimized the classifier is more efficient than with $\eta = 0.1$. Similar conclusions can be done in Figure 13, with Logreg.

X. Conclusion

Label propagation via energy minimization has proven to be able to compete with other machine learning giants in the classical context of digits classification, and more importantly, has distinguished itself in the specific case where it is presented with a very small proportion of labeled data. Experimental results confirm the idea that this method is able to extract meaningful information about the structure of unlabeled data. One has to be careful however when it comes to the implementation of the main algorithms : it is necessary to make some modifications to the theory presented, like smoothing the matrix $D^{-1}W$, writing a Laplace smoothing for the class priors q, carefully choosing η when incorporating another classifier, and optimizing gradient descent and gradient computation so as to limit running time and cost of the method. Although considering an external classifier seems to improve overall accuracy, and can avoid cases of poor predicting on the part of label propagation, some pathological behaviour occasionally happens in the low labeled set size setting.

APPENDIX
(PROOFS AND GRAPHS)

Proposition A.1: The solution to the energy minimization problem introduced in Section III satisfies the harmonic condition that $\Delta f = D - W = 0$.

Proof. The proof follows from the following sequence of equivalences, and the fact that $\nabla E(f) = 0$ if and only if $\frac{\partial E(f)}{\partial f_k} = 0$ for all $k \in \{1, ..., n\}$.

$$\frac{\partial E(f)}{\partial f_k} = 0$$
$$\iff \sum_j w_{kj}(f_k - f_j) - \sum_i w_{ik}(f_i - f_k) = 0$$
$$\iff 2\sum_j w_{kj}(f_k - f_j) = 0$$
$$\iff \sum_j w_{kj} f_k = \sum_j w_{kj} f_j$$
$$\iff f_k = \frac{1}{d_k}\sum_j w_{kj} f_j$$ This means that the value of f at each unlabeled data point is the average of f at neighboring points, which is the harmonic condition
$$\iff Df = Wf \iff \Delta f = 0$$

Proposition A.2: The closed-form expression of the solution in the proposition above is expressed as $f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l$.

Proof. The proof immediately follows from the sequence of equivalences shown below.

$$f_j = \frac{1}{d_j}\sum_{k=1}^{n} w_{jk} f_k,\ j \in [n]$$
$$\iff d_j f_j = \sum_{k=1}^{n} w_{jk} f_k,\ j \in [n]$$
$$\iff d_j f_j - \sum_{k=l+1}^{l+u} w_{jk} f_k = \sum_{k=1}^{l} w_{jk} f_k,\ j \in [n]$$
$$\iff (D_{uu} - W_{uu}) f_u = W_{ul} f_l$$
$$\iff f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l$$

A quick note on the behaviour of kNN.
Its performance in very low dimension is to be expected, since when k is exactly the cardinality of the "training set", the corresponding classifier is just a constant. Moreover, we did not compute the results of kNN dimension in large labeled set size, since the kNN method generally suffers from the so-called *curse of dimensionality*, which would be our case as we work with images having 256 features, *i.e.* observations living in $\mathbb{R}^{256}$. (Note that our labeled set size corresponds to the cardinality of the training set when comparing label propagation via energy minimization with another method.)

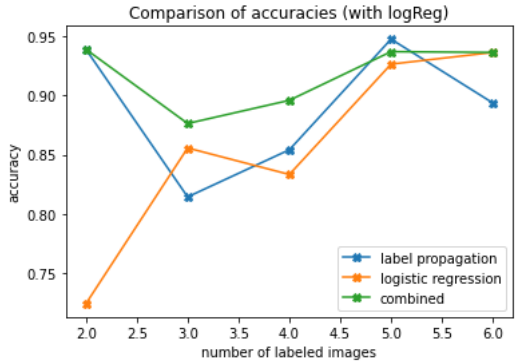
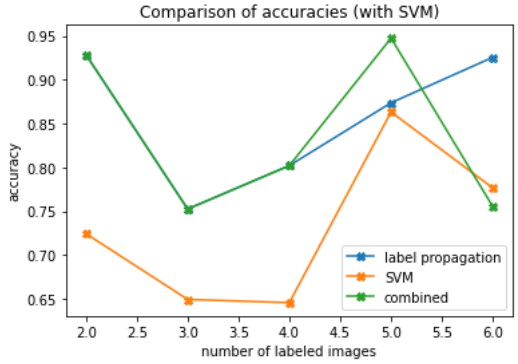


Figure 4: Accuracies in low labeled set sizes. LogReg against energy minimization. Optimized η_l, $\sigma_d = 380 \forall d$, $l \in \{2, 3, 4, 5, 6\}$, $n = 100$, $m = 256$.



.

Figure 5: Accuracies in low labeled set sizes. SVM competing against energy minimization. Optimized η_l, $\sigma_d = 380 \forall d$, $l \in \{2, 3, 4, 5, 6\}$, $n = 100$, $m = 256$.

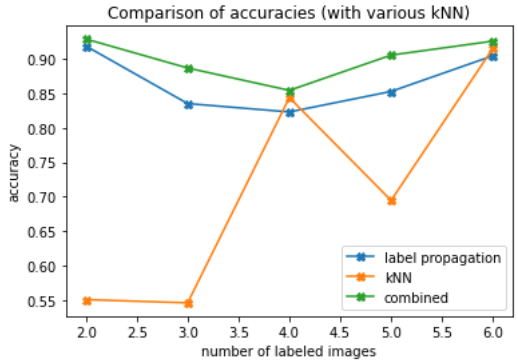


Figure 6: Accuracies in low labeled set sizes. kNN competing against energy minimization. Optimized η_l, $\sigma_d = 380 \forall d$, $l \in \{2, 3, 4, 5, 6\}$, $n = 100$, $m = 256$.

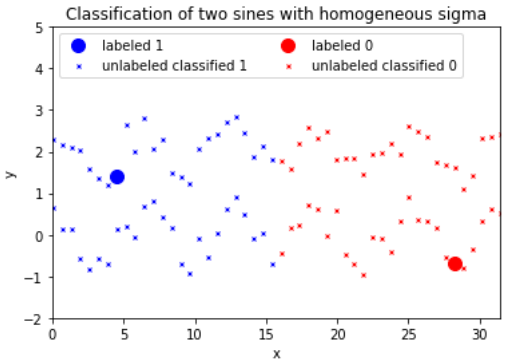


Figure 7: Harmonic solution on synthetic data set, with a homogeneous $\sigma : (\sigma_x, \sigma_y) = (100, 100)$.

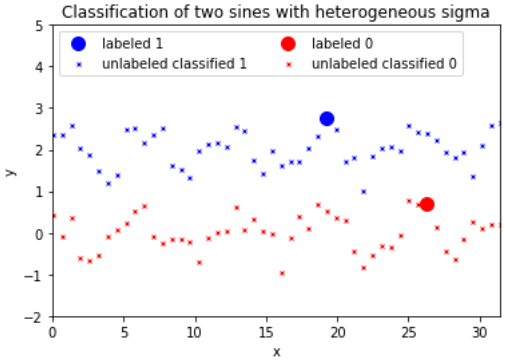


Figure 8: Harmonic solution on synthetic data set, with a heterogeneous $\sigma : (\sigma_x, \sigma_y) = (\infty, 100)$.

References

[1] Xiaojin Zhu, Zoubin Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[2] Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984.

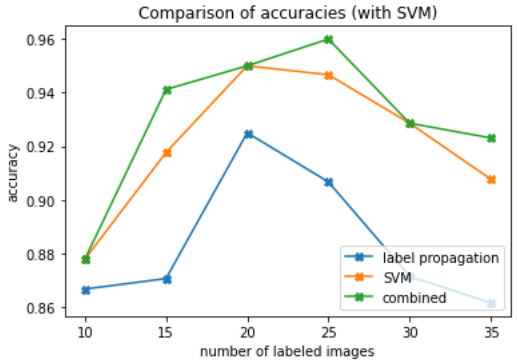


Figure 9: Accuracies in *large* labeled set sizes. SVM against energy minimization method. Optimized η_l, $\sigma_d = 380 \forall d$, $l \in \{10, 15, 20, 25, 30, 35\}$, $n = 100$, $m = 256$.

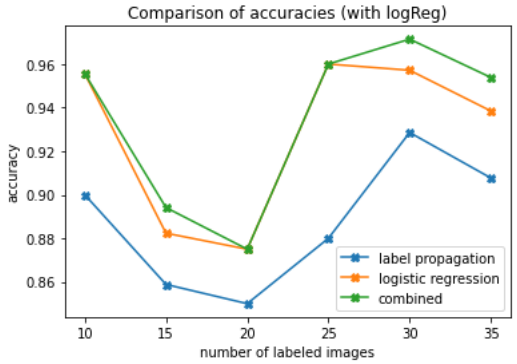


Figure 10: Accuracies in *large* labeled set sizes. LogReg against energy minimization method. Optimized η_l, $\sigma_d = 380 \forall d$, $l \in \{10, 15, 20, 25, 30, 35\}$, $n = 100$, $m = 256$.

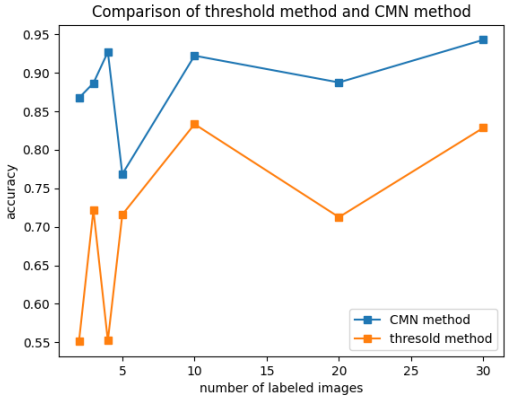


Figure 11: Accuracies of threshold method against CMN method with $\eta = 0.1$, $\sigma_d = 380 \forall d$, $l \in \{2, 3, 4, 5, 10, 20, 30\}$, $n = 100$, $m = 256$.

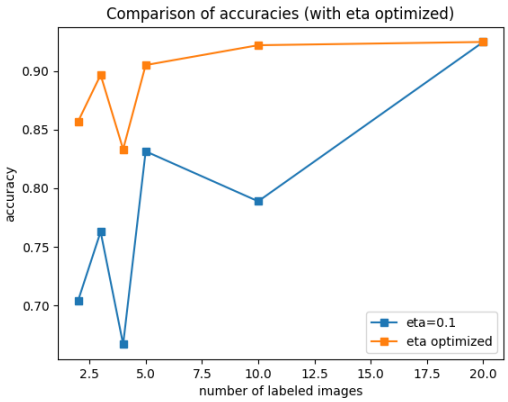


Figure 12: Accuracies of SVM as external classifier with optimized η_l against SVM as external classifier with $\eta = 0.1$, $\sigma_d = 380 \forall d$, $l \in \{2, 3, 4, 5, 10, 20\}$, $n = 100$, $m = 256$.

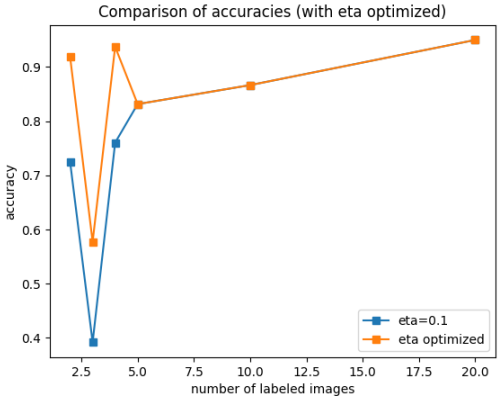


Figure 13: Accuracies of Logreg as external classifier with optimized η_l against Logreg as external classifier with $\eta = 0.1$, $\sigma_d = 380 \forall d$, $l \in \{2, 3, 4, 5, 10, 20\}$, $n = 100$, $m = 256$.