DE GRUYTER OPEN Acta Univ. Sapientiae, Informatica 7, 1 (2015) 18–30

DOI: 10.1515/ausi-2015-0010

A note on label propagation for semi-supervised learning

Zalán BODÓ
Babeş–Bolyai University
email: zbodo@cs.ubbcluj.ro

Lehel CSATÓ
Babeş–Bolyai University
email: lehel.csato@cs.ubbcluj.ro

Abstract. Semi-supervised learning has become an important and thoroughly studied subdomain of machine learning in the past few years, because gathering large unlabeled data is almost costless, and the costly human labeling process can be minimized by semi-supervision. Label propagation is a transductive semi-supervised learning method that operates on the—most of the time undirected—data graph. It was introduced in [8] and since many variants were proposed. However, the base algorithm has two variants: the first variant presented in [8] and its slightly modified version used afterwards, e.g. in [7]. This paper presents and compares the two algorithms—both theoretically and experimentally—and also tries to make a recommendation which variant to use.

1 Introduction

Label propagation is a *transductive graph-based* method for *semi-supervised* classification. It is transductive because the algorithm can predict the labels of the points included in the unlabeled learning dataset, it does not output an inductive classifier applicable for a new point. However, it is true that using a simple *trick* one can obtain a formula for calculating the label of an unknown point without re-learning [1]. We call it graph-based, because it is interpreted

Computing Classification System 1998: I.1.2, I.2.6
Mathematics Subject Classification 2010: 68W40, 68T10
Key words and phrases: label propagation, semi-supervised learning

as building a graph connecting the data points and assigning weights to these edges according to some similarity measure, and then *propagating the labels from the labeled points towards the unlabeled ones*. It is semi-supervised, since some labeled points are needed—usually a small number of such points, and can have a much larger set of unlabeled ones—which direct the algorithm towards a stable labeling configuration.

The aim of this short paper is to analyze two variants of the label propagation algorithm proposed for semi-supervised learning: the first variant that appeared in [8] and its *sightly* modified version used afterwards. An interesting fact is that there is an unmentioned minor modification in the second variant that alters the problem and produces slightly different outputs. Most of the researchers use the second variant of the basic label propagation algorithm as appeared in [7], without any reference to the discrepancy.

We study the differences between the two methods by analyzing the labels output by the algorithms, as well as the underlying optimization problems, and show this on some benchmark datasets. Other variants of label propagation and related methods that will not be discussed here can be found in [1, ch.11].

Section 2 presents the generic algorithm, while Subsections 2.1 and 2.2 present the above-mentioned variants of it. Section 3 analyzes and compares the different outputs of the two variants, while in Section 4 the optimization problems corresponding to the variants of label propagation are examined. In Section 5 the algorithms are compared experimentally on various datasets and the results of the comparison are discussed.

2 Label propagation

The iterative algorithm of label propagation is shown in Alg. 1. The matrix $\mathbf{Y}$ is an $N \times k$ matrix, where $N = \ell + u$ (ℓ denotes the number of labeled and u the number of unlabeled points) and k represent the size of the dataset and the number of classes, respectively. Later we will split the dataset into labeled and unlabeled parts, putting the labeled examples at the beginning of the dataset and use the notation $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_L \\ \mathbf{Y}_U \end{bmatrix}$, where $\mathbf{Y}_L$ denotes the known and $\mathbf{Y}_U$ the unknown labels. This is a matrix with rows containing the probabilities that a point belongs to a given class.

The matrix $\mathbf{T}$ is an $N \times N$ *transition* matrix realizing the propagation of the labels. The construction of this matrix will be detailed in the following subsections. Step 3 is optional in the algorithm—only the first version requires

Algorithm 1 Label propagation

```
1: repeat
2:     $\mathbf{Y} = \mathbf{T}\mathbf{Y}$
3:     (Row-normalize $\mathbf{Y}$.)
4:     Clamp the labeled data.
5: until convergence
```

it—this is the reason why this operation is put in paranthesis.

Label propagation is not necessarily an iterative method: the output labels of the unlabeled points can be expressed analytically.[1] First we partition the multiplication operation in step 2 of the algorithm:

$$\begin{bmatrix} \mathbf{Y}_L \\ \mathbf{Y}_U \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{LL} & \mathbf{T}_{LU} \\ \mathbf{T}_{UL} & \mathbf{T}_{UU} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_L \\ \mathbf{Y}_U \end{bmatrix}, \tag{1}$$

from which we can express the recursive formula for the unknown labels, that is

$$\mathbf{Y}_U = \mathbf{T}_{UL}\mathbf{Y}_L + \mathbf{T}_{UU}\mathbf{Y}_U.$$

The labels output by label propagation then can be expressed as

$$\mathbf{Y}_U = (\mathbf{I} - \mathbf{T}_{UU})^{-1}\,\mathbf{T}_{UL}\mathbf{Y}_L.$$

Obviously, in order to be able to solve the problem, $\mathbf{I}-\mathbf{T}_{UU}$ must be invertible, but we assume this is the case.[2]

2.1 The first variant

The subtle—but unmentioned in the literature and undiscussed—difference between the two variants lies in the construction of the transition matrix $\mathbf{T}$. This matrix is based on the graph built to represent data similarities. First we define the matrix $\mathbf{W}$ containing the similarities. In [8] this is constructed using the Gaussian similarity,

$$W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right), \quad i, j = 1, 2, \dots, N, \tag{2}$$

[1] As will be shown in Section 4, the label propagation algorithms presented in this paper have corresponding optimization problems, that can be solved in many different ways.

[2] A detailed analysis of the convergence of label propagation can be found in [8].

but other similarities can be used as well, provided that $\mathbf{I} - \mathbf{T}_{UU}$ remains invertible. The similarity matrix can also be sparse—the description of some useful matrix construction techniques can be found in [5].

We also introduce here the diagonal degree matrix defined as

$$\mathbf{D} = \operatorname{diag}(\mathbf{W1}).$$

Based on the similarities we can define the transition probability matrix,

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W},$$

in which P_{ij} contains the probability—based on data similarities—that from point i we transition to point j.

The first variant propagates the labels from the labeled points towards the unlabeled points, thus $\mathbf{T} := \mathbf{P}'$, that is the label is determined by

$$Y_{ij} = P_{1i}Y_{1j} + P_{2i}Y_{2j} + \cdots + P_{Ni}Y_{Nj}.$$

Similarly to the partitioning of $\mathbf{Y}$ into labeled and unlabeled parts, we can also split $\mathbf{W}$, $\mathbf{D}$ and $\mathbf{P}$, thus obtaining

$$\begin{aligned} \mathbf{Y}_U &= \left(\mathbf{I} - \mathbf{P}'_{UU}\right)\mathbf{P}'_{LU}\mathbf{Y}_L \\ &= \mathbf{D}_U\left(\mathbf{D}_U - \mathbf{W}_{UU}\right)^{-1}\mathbf{W}_{UL}\mathbf{D}_L^{-1}\mathbf{Y}_L. \end{aligned} \tag{3}$$

In this case re-normalization of the rows of $\mathbf{Y}$ is needed in the iterative algorithm, because

$$\begin{aligned} \sum_{j=1}^{k} Y_{ij} &= P_{1i}\sum_{j=1}^{k} Y_{1j} + \cdots + P_{Ni}\sum_{j=1}^{k} Y_{Nj} \\ &= P_{1i} + P_{2i} + \cdots + P_{Ni} \end{aligned}$$

does not sum to one. However, steps 2 and 3 of Alg. 1 can be combined into $\mathbf{Y} = \overline{\mathbf{T}}\mathbf{Y}$, where $\overline{\mathbf{T}}$ is the row-normalized matrix of $\mathbf{T}$, $\overline{T}_{ij} = T_{ij}/\sum_{k=1}^{N} T_{ik}, \forall i, j$. Thus, it is sufficient to perform row-normalization once, right before starting to propagate the labels.

2.2 The second variant

In the second variant the labels are propagated "backwards", that is $\mathbf{T} := \mathbf{P}$ is used for the transition matrix, resulting in the following label determination:

$$Y_{ij} = P_{i1}Y_{1j} + P_{i2}Y_{2j} + \cdots + P_{iN}Y_{Nj}.$$

In this case we can say that the label of a point is defined as the convex combination of its forward neighbors' labels. The analytic formula for calculating the labels becomes

$$\begin{aligned} \mathbf{Y}_U &= (\mathbf{I} - \mathbf{P}_{UU})^{-1}\,\mathbf{P}_{UL}\mathbf{Y}_L \\ &= (\mathbf{D}_U - \mathbf{W}_{UU})^{-1}\,\mathbf{W}_{UL}\mathbf{Y}_L. \end{aligned} \tag{4}$$

This version of label propagation appeared in [7] and in papers published afterwards, without mentioning the minor modification to the original algorithm. In this variant re-normalization is not needed, since the rows of $\mathbf{Y}$ sum to one:

$$\begin{aligned} \sum_{j=1}^{k} Y_{ij} &= P_{i1}\sum_{j=1}^{k} Y_{1j} + \cdots + P_{iN}\sum_{j=1}^{k} Y_{Nj} \\ &= P_{i1} + P_{i2} + \cdots + P_{iN} = 1. \end{aligned}$$

All the formulae used in label propagation can be rewritten using the *graph Laplacian* [2, 5], a central concept of graph-based learning methods, defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

The Laplacian possesses some interesting and advantageous properties [5], and will be used in Section 4 mostly to simplify the expressions.

3 Analysis of the outputs

Let us denote the $u \times \ell$ matrix $(\mathbf{D}_U - \mathbf{W}_{UU})^{-1}\,\mathbf{W}_{UL}$ appearing in both analytical formulae by $\mathbf{A}$, which is a matrix with stochastic vectors in its rows. The matrix $\mathbf{A}$ equals $(\mathbf{I} - \mathbf{P}_{UU})^{-1}\,\mathbf{P}_{UL}$ and we will use this to prove our previous statement (i.e. stochastic rows property). Using this notation we can write the recursive formula of the two methods as

$$\mathbf{Y}_U = \mathbf{D}_U\mathbf{A}\mathbf{D}_L^{-1}\mathbf{Y}_L \tag{5}$$

$$\mathbf{Y}_U = \mathbf{A}\mathbf{Y}_L. \tag{6}$$

We can prove that the rows of $\mathbf{A}$ are stochastic in two steps: (i) $\sum_{j=1}^{\ell} A_{ij} = 1, i = 1, 2, \ldots, u$, (ii) $A_{ij} \geq 0, i = 1, 2, \ldots, u, j = 1, 2, \ldots, \ell$. For the first property we use eq. (6) and check the sum of the i-th row of $\mathbf{Y}_U$, for which it is easy to see that

$$\begin{aligned} \sum_{j=1}^{\ell} (\mathbf{Y}_U)_{ij} &= A_{i1}\sum_{j=1}^{k} Y_{1j} + \cdots + A_{i\ell}\sum_{j=1}^{k} Y_{\ell j} \\ &= A_{i1} + A_{i2} + \cdots + A_{i\ell}, \end{aligned}$$

and since this is the second variant, we know that the rows of $\mathbf{Y}$ sum to one, therefore the rows of $\mathbf{A}$ sum to one as well.

For the second property we consider the definition of $\mathbf{A}$ as $(\mathbf{I}-\mathbf{P}_{UU})^{-1}\,\mathbf{P}_{UL}$ and use the Neumann series to rewrite it as

$$\mathbf{A} = \sum_{i=0}^{\infty} \mathbf{P}_{UU}^{i} \mathbf{P}_{UL}. \tag{7}$$

Since both $\mathbf{P}_{UU}$ and $\mathbf{P}_{UL}$ contain only nonnegative values, $\mathbf{A}$ also contains only nonnegatives.

From (7) the value A_{ij} can be interpreted as the probability that the first labeled node of a random walk starting at unlabeled node i is j.

In the first variant we can leave out the multiplication with $\mathbf{D}_U$, observing that it does not influence the result. Therefore we are left with the following two *very similar* formulae:

$$\mathbf{Y}_U = \mathbf{A}\mathbf{D}_L^{-1}\mathbf{Y}_L \tag{8}$$

$$\mathbf{Y}_U = \mathbf{A}\mathbf{Y}_L. \tag{9}$$

4 Analysis of the optimization problems

The outputs of the presented methods can be viewed as solutions of specific (semi-supervised) optimization problems. In this section we present and analyze the corresponding problems. We start with the second variant, because the optimization problem of the first label propagation variant can be viewed as a special case of the second variant's minimization problem.

Statement 1 *The output labels (4) in the second variant of label propagation are also a solution of the following optimization problem:*

$$\min_{\mathbf{y}_i, i=\ell+1,\dots,N} \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2, \tag{10}$$

where $\mathbf{y}_i$ denotes the i-th row of $\mathbf{Y}$, that is the probabilistic vector assigned to the i-th data point.

Proof. The minimizable expression can be written as $\mathrm{tr}(\mathbf{Y}\mathbf{Y}'\mathbf{L}) = \mathrm{tr}(\mathbf{Y}'\mathbf{L}\mathbf{Y})$, because

$$\begin{aligned}
\frac{1}{2}\sum_{i,j=1}^{N} W_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 &= \frac{1}{2}\sum_i \mathbf{y}_i'\mathbf{y}_i \sum_j W_{ij} + \frac{1}{2}\sum_j \mathbf{y}_j'\mathbf{y}_j \sum_i W_{ij} - \sum_{i,j} W_{ij}\mathbf{y}_i'\mathbf{y}_j \\
&= \sum_{i,j} \mathbf{y}_i'\mathbf{y}_i D_{ii} - \sum_{i,j} W_{ij}\mathbf{y}_i'\mathbf{y}_j = \mathrm{tr}(\mathbf{Y}'\mathbf{D}\mathbf{Y}) - \mathrm{tr}(\mathbf{Y}'\mathbf{W}\mathbf{Y}) \\
&= \mathrm{tr}(\mathbf{Y}'\mathbf{L}\mathbf{Y}).
\end{aligned}$$

Hence we can rewrite the optimization problem (10) in the more compact form of

$$\min_{\mathbf{Y}_u} \mathrm{tr}(\mathbf{Y}'\mathbf{L}\mathbf{Y}). \tag{11}$$

The $\mathbf{Y}_u$ that minimizes the above expression can be found by taking the derivative of the trace with respect to $\mathbf{Y}_u$ and setting it equal to zero. First we partition $\mathbf{L}$ and $\mathbf{Y}$ into labeled and unlabeled blocks, similarly to the matrices in (1), and expand our formula

$$\mathbf{Y}'\mathbf{L}\mathbf{Y} = \mathbf{Y}_L'\mathbf{L}_{LL}\mathbf{Y}_L + \mathbf{Y}_u'\mathbf{L}_{uL}\mathbf{Y}_L + \mathbf{Y}_L'\mathbf{L}_{Lu}\mathbf{Y}_u + \mathbf{Y}_u'\mathbf{L}_{uu}\mathbf{Y}_u.$$

Applying the trace operator we get

$$\mathrm{tr}(\mathbf{Y}'\mathbf{L}\mathbf{Y}) = \mathrm{tr}(\mathbf{Y}_L'\mathbf{L}_{LL}\mathbf{Y}_L + 2\mathbf{Y}_u'\mathbf{L}_{uL}\mathbf{Y}_L + \mathbf{Y}_u'\mathbf{L}_{uu}\mathbf{Y}_u).$$

Then we take the derivative of the trace with respect to $\mathbf{Y}_u$ and set it equal to zero, thus obtaining

$$\mathbf{Y}_u = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{uL}\mathbf{Y}_L = (\mathbf{D}_u - \mathbf{W}_{uu})^{-1}\mathbf{W}_{uL}\mathbf{Y}_L. \tag{12}$$

as in (4).[3] □

Statement 2 *The output labels (3) of the first variant of label propagation minimize the following expression:*

$$\min_{\mathbf{y}_i, i=\ell+1,\dots,N} \frac{1}{2}\sum_{i,j=1}^{N} W_{ij}\left\|\frac{\mathbf{y}_i}{D_{ii}} - \frac{\mathbf{y}_j}{D_{jj}}\right\|^2, \tag{13}$$

where D_{ii} *is the* i*-th diagonal element of* $\mathbf{D}$*.*

[3] In the derivation we used the following properties of the trace [4]: $\mathrm{tr}(\mathbf{A}+\mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$, $\mathrm{tr}(\mathbf{A}') = \mathrm{tr}(\mathbf{A})$, $\frac{\partial\,\mathrm{tr}(\mathbf{X}'\mathbf{A})}{\partial \mathbf{X}} = \mathbf{A}$, $\frac{\partial\,\mathrm{tr}(\mathbf{X}'\mathbf{A}\mathbf{X})}{\partial \mathbf{X}} = (\mathbf{A}+\mathbf{A}')\mathbf{X}$.

Proof. By using the substitution $\mathbf{z}_i := \mathbf{y}_i / D_{ii}$ (or $\mathbf{Z} := \mathbf{D}^{-1}\mathbf{Y}$) we arrive to the following optimization problem

$$\min_{\mathbf{y}_i, i=\ell+1,\dots,N} \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|^2 .$$

Similarly to our previous proof, we can write this as $\mathrm{tr}(\mathbf{Z}'\mathbf{L}\mathbf{Z})$, which equals $\mathrm{tr}(\mathbf{Y}'\mathbf{D}^{-1}\mathbf{L}\mathbf{D}^{-1}\mathbf{Y})$, where for simplicity we use the substitution $\mathbf{G} := \mathbf{D}^{-1}\mathbf{L}\mathbf{D}^{-1}$. Thus our optimization problem becomes

$$\min_{\mathbf{Y}_U} \mathrm{tr}(\mathbf{Y}'\mathbf{G}\mathbf{Y}).$$

Apart from the middle square matrix, this problem is identical to (11), therefore we can apply the same derivation. Using the results from (12) and substituting $\mathbf{L}$ back we obtain

$$\mathbf{Y}_U = -\mathbf{G}_{UU}^{-1}\mathbf{G}_{UL}\mathbf{Y}_L = \mathbf{D}_U(\mathbf{D}_U - \mathbf{W}_{UU})^{-1}\mathbf{W}_{UL}\mathbf{D}\mathrm{L}^{-1}\mathbf{Y}_L,$$

as in (3). □

The optimization problem in (10) can be explained as follows: we want to determine the stochastic vectors $\mathbf{y}_i$, belonging to the unlabeled points, so that to minimize the distance between these class *membership* vectors depending on, i.e. weighted by the similarities of the neighboring points. The first variant is a *normalized* version of the second: here, instead of $\mathbf{L}$ we use $\mathbf{D}^{-1}\mathbf{L}\mathbf{D}^{-1}$, a normalized graph Laplacian.[4] By dividing $\mathbf{y}_i$ by the degree of the point a greater weight is assigned to points having *fewer* or *distant* neighbors. This, as will be shown in the experiments, can yield a more balanced solution.

5 Experimental results and discussion

In the experiments we used 7 benchmark datasets from [1].[5] The main properties of these sets are summarized in Table 1. Every set has 2 × 12 splits: 12 random splits with 10 labeled points and 12 splits with 100 labeled points. In our experiments we used only the first split of the datasets with 10 labeled data.

[4] This Laplacian is the normalized version of the symmetric normalized Laplacian from [5].

[5] The datasets can be downloaded from `http://olivier.chapelle.cc/ssl-book/benchmarks.html`.

Dataset	Classes	Dimension	Points	Note
g241c	2	241	1500	artificial
g241n	2	241	1500	artificial
Digit1	2	241	1500	artificial
USPS	2	241	1500	imbalanced
COIL$_2$	2	241	1500	
BCI	2	117	400	
Text	2	11 960	1500	sparse discrete

Table 1: Properties of the datasets used in the experiments.

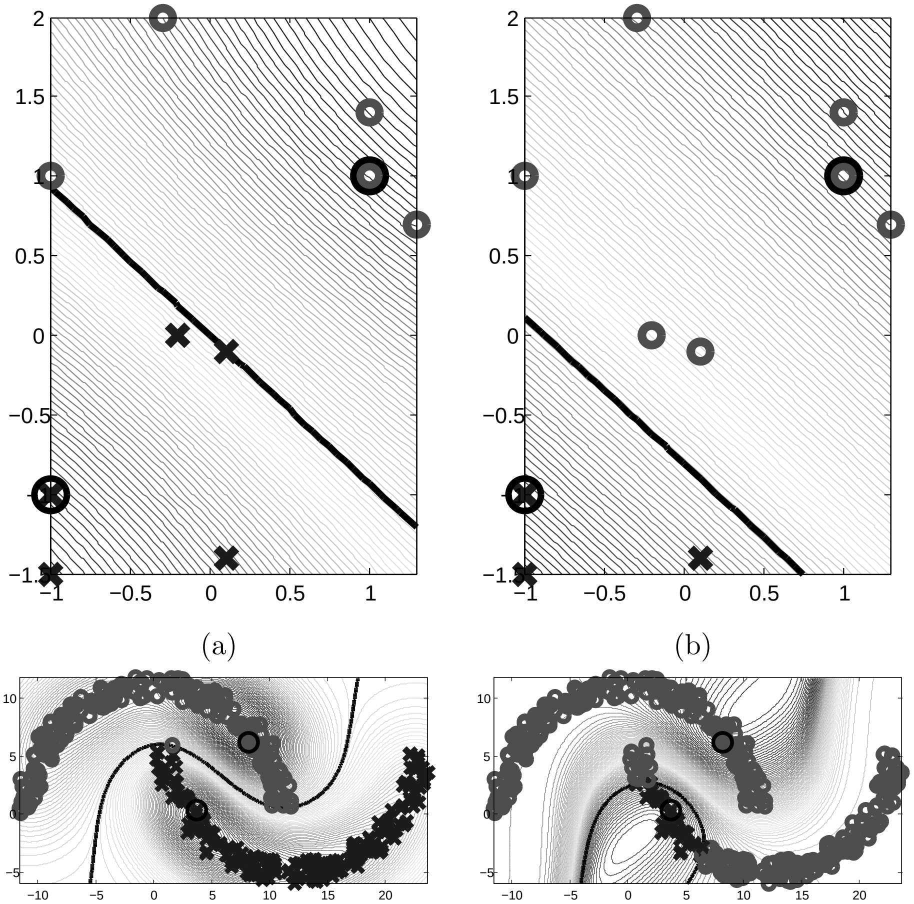
Dataset	σ	E	ΔE	$\Delta \mathbf{Y}_u$	Iterations	
g241c	5.8845	0.5013	0	5.1548	1219	1066
g241n	5.8914	0.5020	0	1.1718	1191	1133
Digit1	0.3941	0.2409	0	0.0334	7235	7270
USPS*	0.9	0.1906	0	0.7705	1	1
COIL$_2$*	400	0.4993	0	$3.12 \cdot 10^{-13}$	932	839
BCI	1.7296	0.5333	0	0.0378	3060	3083
Text*	1.3	0.4987	0	0.0045	780	781

Table 2: Experimental results obtained using Gaussian similarity with the indicated parameter: E – error, ΔE – error difference, $\Delta \mathbf{Y}_u$ – norm of the difference vector of the outputs.

Besides these we experimented with other two sets for visually demonstrating the difference between the analyzed methods. The first of these is the *2moons* dataset containing 2 labeled and 383 unlabeled points, while the second *simple* dataset contains 2 labeled and 8 unlabeled points.[6]

The results obtained are shown in Table 2. In all the experiments we used the Gaussian similarity (2) where the parameter was set using the procedure described in [8]. The minimum spanning tree of the data was constructed using Kruskal's algorithm [3]. The process of building the tree, i.e. connecting the points proceeds until the components being connected contain opposite labels; the length of this peculiar edge is denoted by d_0. Then—following the three-sigma or $68-95-99.7$ rule of the normal distribution [6]—the σ parameter of the Gaussian similarity is set to $d_0/3$. In this way it is expected that the "local propagation is mostly within classes" [8]. In four cases this method provided

[6] The datasets can be downloaded from http://www.cs.ubbcluj.ro/~zbodo/datasets.html.



(a)

(b)

(c)

(d)

Figure 1: The output of (the first and second variant of) label propagation for the (a), (b) *simple* and (c), (d) *2moons* datasets.

acceptable values, but for the remaining sets (marked with a star in Table 2) resulted in ill-conditioned (nearly singular) ($\mathbf{D}_{\mathsf{U}} - \mathbf{W}_{\mathsf{UU}}$) matrices. For these datasets the indicated parameters were set by investigating the histogram of the kernel values.

We list four values: (i) the error, (ii) the error difference $\Delta\mathsf{E}$, that is $\Delta\mathsf{E} = |\mathsf{E}_1 - \mathsf{E}_2|$, where E_i is the error obtained by the i-th method, (iii) the Frobenius norm of the difference vector of the outputs, $\mathbf{\Delta Y}_{\mathsf{U}} = \left\|\mathbf{Y}^1_{\mathsf{U}} - \mathbf{Y}^2_{\mathsf{U}}\right\|_{\mathsf{F}}$ and (iv) the number of iterations. The first three of these are calculated using the analytical formulae (3) and (4). In the last two columns of the table we show the number of iterations required for the iterative algorithms to converge; convergence was reached when the Frobenius norm of the difference matrix obtained from two consecutive steps did not exceed 10^{-3}. The initial matrix of $\mathbf{Y}_U$ was set to $\begin{bmatrix}\mathbf{1} & \mathbf{0}\end{bmatrix}$.[7]

In the contour plots of Figure 1 the outputs of label propagation are shown for two datasets: (a) and (b) show the results of the first and second variant of label propagation for the *simple* dataset, while (c) and (d) present the assigned labels for the *2moons* dataset. In both cases the Gaussian similarity was used with parameter $\sigma = 3$. The encircled points denote the labeled data and the thicker black curves show the decision boundaries of the classifier.[8] These were determined using the following methodology. Considering the optimization problems (10) and (13) one can determine the label of a newly arrived point by taking the derivative of the *new* objective functions with respect to y (y denoting the new, unknown label of the new point $\mathbf{x}$):

$$C_1 + \sum_{i=1}^{N} w(\mathbf{x}, \mathbf{x}_i) \left(\frac{y}{d} - \frac{y_i}{D_{ii}} \right)^2$$

$$C_2 + \sum_{i=1}^{N} w(\mathbf{x}, \mathbf{x}_i)(y - y_i)^2,$$

where C_1 and C_2 denote the unchanged parts of the objective functions and $w(\mathbf{x}, \mathbf{x}_i)$ is the similarity of points $\mathbf{x}$ and $\mathbf{x}_i$. Setting the derivatives equal to zero and expressing the label we obtain

$$\begin{aligned} y &= \sum_{i=1}^{N} \frac{w(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^{N} W_{ij}} y_i \\ y &= \sum_{i=1}^{N} \frac{w(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^{N} w(\mathbf{x}, \mathbf{x}_j)} y_i \end{aligned}$$

for the two variants. For the labels of the unlabeled points (y_i) in the dataset the labels given by the algorithms were used in the above formulae (not the *true* labels).

[7] We also experimented with *class mass normalization* [9] that uses the prior class distribution to influence the predictions, but no significant differences were observed in the results, therefore these results are not divulged here.

[8] We used here the binary version of label propagation, where $\mathbf{Y}$ is an $N \times 1$ vector, $\mathbf{Y}_L \in \{-1, +1\}^\ell$; $y_i \geq 0$ denotes a positive class assignment and $y_i < 0$ a negative label.

Examining the $\Delta \mathsf{E}$, $\Delta \mathbf{Y}_{\mathsf{U}}$ values and the iteration counts in Table 2 we cannot notice considerable differences. The error values show the inadequacy of the learning algorithm or its parameters.[9] We obtained seemingly acceptable error rates for two datasets, but the only set where label propagation performed well was Digit1, taking into account the imbalanced class distribution in the USPS dataset. In order to somehow experimentally compare the two variants we decided to use some toy datasets where the results can be easily visualized. Comparing the results in Figure 1 one can see that the normalization included in the first variant resulted in more balanced (and correct) solutions. We saw that (8) and (9) only differ by the inverse of the diagonal degree matrix $\mathbf{D}_{\mathsf{L}}$, a neglectable computational load from a complexity point of view. The iterative algorithms differ in no steps, since in the first variant it is sufficient to perform row-normalization only once, as discussed in Section 2.1. Normalizing the labels by the points' degree, however, can result in a more natural, more balanced labeling. Therefore, concluding the theoretical analysis and the experiments performed, we recommend to prefer the first variant of label propagation over the second variant, where possible.

Acknowledgement

The authors acknowledge the support of the Romanian Ministry of Education and Research via grant PN-II-RU-TE-2011-3-0278.

References

[1] O. Chapelle, B. Schölkopf, A. Zien, *Semi-Supervised Learning*, The MIT Press, Cambridge, 2006. ⇒18, 19, 25

[2] F. Chung, *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*, AMS, Philadelphia, 1997. ⇒22

[3] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, **7,** 1 (1956) 48–50. ⇒26

[4] H. Lütkepohl, *Handbook of Matrices*, John Wiley & Sons, Chichester, 1996. ⇒ 24

[5] U. von Luxburg, A tutorial on spectral clustering, *Stat. Comput.*, **17**, 4 (2007) 395–416. ⇒21, 22, 25

[6] M. W. Trosset, *An Introduction to Statistical Inference and Its Applications with R*, Chapman & Hall/CRC Texts in Statistical Science, CRC Press, Boca Raton, 2009. ⇒26

[9] By parameters we refer both to the similarity function and its parameters.

[7] X. Zhu, Semi-supervised learning *with graphs*, PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 2005. ⇒18, 19, 22

[8] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002. ⇒18, 19, 20, 26

[9] X. Zhu, Z. Ghahramani, J. D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, *Proc. 20th ICML*, 2003. pp. 912–919. ⇒28

Received: August 11, 2014 • *Revised: January 23, 2015*