Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

LASS: A Simple Assignment Model with Laplacian Smoothing

Miguel Á. Carreira-Perpiñán and **Weiran Wang**
Electrical Engineering and Computer Science, School of Engineering,
University of California, Merced

Abstract

We consider the problem of learning soft assignments of N items to K categories given two sources of information: an item-category similarity matrix, which encourages items to be assigned to categories they are similar to (and to not be assigned to categories they are dissimilar to), and an item-item similarity matrix, which encourages similar items to have similar assignments. We propose a simple quadratic programming model that captures this intuition. We give necessary conditions for its solution to be unique, define an out-of-sample mapping, and derive a simple, effective training algorithm based on the alternating direction method of multipliers. The model predicts reasonable assignments from even a few similarity values, and can be seen as a generalization of semisupervised learning. It is particularly useful when items naturally belong to multiple categories, as for example when annotating documents with keywords or pictures with tags, with partially tagged items, or when the categories have complex interrelations (e.g. hierarchical) that are unknown.

1 Introduction

A major success in machine learning in recent years has been the development of semisupervised learning (SSL) (Chapelle, Schölkopf, and Zien 2006), where we are given labels for only a few of the training points. Many SSL approaches rely on a neighborhood graph constructed on the training data (labeled and unlabeled), typically weighted with similarity values. The Laplacian of this graph is used to construct a quadratic nonnegative function that measures the agreement of possible labelings with the graph structure, and minimizing it given the existing labels has the effect of propagating them over the graph. Laplacian-based formulations are conceptually simple, computationally efficient (since the Laplacian is usually sparse), have a solid foundation in graph theory and linear algebra (Chung 1997; Doyle and Snell 1984), and most importantly work very well in practice. The graph Laplacian has been widely exploited in machine learning, computer vision and graphics, and other areas: as mentioned, in semisupervised learning, manifold regularization and graph priors (Zhu, Ghahramani, and Lafferty 2003; Belkin, Niyogi, and Sindhwani 2006; Zhou et al. 2004) for regression, classification and applications such as supervised image segmentation (Grady 2006), where one solves a Laplacian-based linear system; in spectral clustering (Shi and Malik 2000), possibly with constraints (Lu and Carreira-Perpiñán 2008), and spectral dimensionality reduction (Belkin and Niyogi 2003), where one uses eigenvectors of the Laplacian; or in clustering, manifold denoising and surface smoothing (Carreira-Perpiñán 2006; Wang and Carreira-Perpiñán 2010; Taubin 1995), where one iterates products of the data with the Laplacian.

We concern ourselves with assignment problems in a semisupervised learning setting, where we have N items and K categories and we want to find soft assignments of items to categories given some information. This information often takes the form of partial tags or annotations, e.g. for pictures in websites such as Flickr, blog entries, etc. Let us consider a specific example where the items are documents (e.g. papers submitted to this conference) and the categories are keywords. Any given paper will likely be associated to a larger or smaller extent with many keywords, but most authors will tag their papers with only a few of them, usually the most distinctive (although, as we know, there may be other reasons). Thus, few papers will be tagged as "computer science" or "machine learning" because those keywords are perceived as redundant given, say, "semisupervised learning". However, considered in a larger context (e.g. to include biology papers), such keywords would be valuable. Besides, categories may have various correlations that are unknown to us but that affect the assignments. For example, a hierarchical structure implies that "machine learning" belongs to "computer science" (although it does to "applied maths" to some extent as well). In general, we consider categories as sets having various intersection, inclusion and exclusion relations. Finally, it is sometimes practical to tag an item as *not associated* with a certain category, e.g. "this paper is not about regression" or "this patient does not have fever", particularly if this helps to make it distinctive. In summary, in this type of applications, *it is impractical for an item to be fully labeled over all categories, but it is natural for it to be associated or disassociated with a few categories*. This can be coded with item-category similarity values that are positive or negative, respectively, with the magnitude indicating the degree of association, and zero meaning indifference

or ignorance. We call this source of partial supervisory information, which is specific for each item irrespectively of other items, the *wisdom of the expert*.

We also consider another practical source of information. Usually it is easy to construct a similarity of a given item to other items, at least its nearest neighbors. For example, with documents or images, this could be based on a bag-of-words representation. We would expect similar items to have similar assignment vectors, and this can be captured with an item-item similarity matrix and its graph Laplacian. We call this source of information, which is about an item in the context of other items, the *wisdom of the crowd*.

In this paper, we propose a simple model that combines both types of information as a quadratic program. We give some properties of the solution, define an out-of-sample mapping, derive a training algorithm, and illustrate the model with document and image data. Proofs and derivations are in (Carreira-Perpiñán and Wang 2014).

2 The Laplacian Assignment (LASS) Model

We consider the following assignment problem. We have N items and K categories, and we want to determine soft assignments z_{nk} of each item n to each category k, where $z_{nk} \in [0,1]$ and $\sum_{k=1}^{K} z_{nk} = 1$ for each $n = 1,\dots,N$. We are given two similarity matrices, suitably defined, and typically sparse: an item-item similarity matrix $\mathbf{W}$, which is an $N \times N$ matrix of affinities $w_{nm} \geq 0$ between each pair of items n and m; and an item-category similarity matrix $\mathbf{G}$, which is an $N \times K$ matrix of affinities $g_{nk} \in \mathbb{R}$ between each pair of item n and category k (negative affinities, i.e., dissimilarities, are allowed in $\mathbf{G}$). We want to assign items to categories optimally as follows:

$$\min_{\mathbf{Z}} \quad \lambda \operatorname{tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right) - \operatorname{tr}\left(\mathbf{G}^T\mathbf{Z}\right) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{Z}\mathbf{1}_K = \mathbf{1}_N,\ \mathbf{Z} \geq \mathbf{0} \tag{1b}$$

where $\lambda > 0$, $\mathbf{1}_K$ is a vector of K ones, and $\mathbf{L}$ is the $N \times N$ graph Laplacian matrix, obtained as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)$ is the degree matrix of the weighted graph defined by $\mathbf{W}$. The problem is a quadratic program (QP) over an $N \times K$ matrix $\mathbf{Z} = (\mathbf{z}_1,\dots,\mathbf{z}_K)$, i.e., NK variables[1] altogether, where $\mathbf{z}_k$, the kth column of $\mathbf{Z}$, contains the assignments of each item to category k. We will call problems of the type (1) *Laplacian assignment problems (LASS)*. Minimizing objective (1a) encourages items to be assigned to categories with which they have high similarity (the linear term in $\mathbf{G}$), while encouraging similar items to have similar assignments (the Laplacian term in $\mathbf{L}$), since $\operatorname{tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right) = \frac{1}{2}\sum_{n,m=1}^{N} w_{nm}\left\|\mathbf{z}_n - \mathbf{z}_m\right\|^2$ where $\mathbf{z}_n$ is the nth row of $\mathbf{Z}$, i.e., the assignments for item n. Although we could absorb λ inside $\mathbf{G}$, we will find it more convenient to fix the scale of each similarity g_{nk} to, say, the interval $[-1,1]$ (where ± 1 mean maximum (dis)similarity and 0 ignorance), and then let λ control the strength of the Laplacian term.

[1] We will use a boldface vector $\mathbf{z}_n$ or to mean the nth row of $\mathbf{Z}$ (as a column vector), and a boldface vector $\mathbf{z}_k$ to mean the kth column of $\mathbf{Z}$ (likewise for $\mathbf{g}_n$ or $\mathbf{g}_k$). The context and the index (n or k) will determine which is the case. This mild notation abuse will simplify the explanations.

Particular cases We can determine the solution(s) for the extreme values of λ. (1) If $\lambda = 0$, then the LASS problem is a linear program and separates over each item $n = 1,\dots,N$. The solution is $z_{nk} = \delta(k - k_{\max}(n))$ where $k_{\max}(n) = \arg\max\{g_{nk},\ k = 1,\dots,K\}$, i.e., each item is assigned to its most similar category. This tells us what the linear term can do by itself. (2) If $\lambda = \infty$ or $\mathbf{G} = \mathbf{0}$, then the LASS problem is a QP with an infinite number of solutions of the form $\mathbf{z}_n = \mathbf{z}_m$ for each $n, m = 1,\dots,N$, i.e., all items have the same assignments. This tells us what the Laplacian term can do by itself. With intermediate $\lambda > 0$, more interesting solutions appear (particularly when the similarity matrices are sparse), where the item-category similarities are propagated to all points through the item-item similarities.

We can also show: (1) An item for which no similarity to any category is given (i.e., no expert information) receives as assignment the average of its neighbors. This corresponds to the SSL prediction. (2) A category for which no item has a positive similarity receives no assignments.

Existence and unicity of the solution The LASS problem is a convex QP, so general results of convex optimization tell us that all minima are global minima. However, since the Hessian of the objective function is positive semidefinite, there can be multiple minima. Theorem 2.1 characterizes the solutions, and its corollary gives a sufficient condition for the minimum to be unique.

Theorem 2.1. *Assume the graph Laplacian* $\mathbf{L}$ *corresponds to a connected graph and let* $\mathbf{Z}^* \in \mathbb{R}^{N\times K}$ *be a solution (minimizer) of the LASS problem* (1)*. Then, any other solution has the form* $\mathbf{Z}^* + \mathbf{1}_N\mathbf{p}^T$ *where* $\mathbf{p} \in \mathbb{R}^K$ *satisfies the conditions:*

$$\mathbf{p}^T\mathbf{1}_K = 0, \quad \mathbf{p}^T(\mathbf{G}^T\mathbf{1}_N) = 0, \quad \mathbf{Z}^* + \mathbf{1}_N\mathbf{p}^T \geq \mathbf{0}. \tag{2}$$

In particular, for each $k = 1,\dots,K$ *for which* $\exists n \in \{1,\dots,N\}$: $z^*_{nk} = 0$*, then* $p_k \geq 0$*.*

Corollary 2.2. *Under the assumptions of th. 2.1, if* $\max_k\left(\min_n\left(z^*_{nk}\right)\right) = 0$ *then the solution* $\mathbf{Z}^*$ *is unique.*

Note the following. If the graph Laplacian $\mathbf{L}$ corresponds to a graph with multiple connected components, then the LASS problem separates into a problem for each component, and theorem 2 holds in each component. Computationally, it is also more efficient to solve each problem separately. The set (2) of solutions to a LASS problem is a convex polytope. The condition of corollary 2.2 means that each category has at least one item with a zero assignment to it. In practice, we can expect this condition to hold, and therefore the solution to be unique, if the categories are sufficiently distinctive and λ is small enough. Experimentally, we have not found nonunique solutions. Practically, one can always make the solution unique by replacing $\mathbf{L}$ with $\mathbf{L} + \epsilon\mathbf{I}_N$ where $\epsilon > 0$ is a small value, since this makes the objective strongly convex. (This has the equivalent meaning of adding a penalty $\epsilon\|\mathbf{Z}\|_F^2$ to it, which has the effect of biasing the assignment vector $\mathbf{z}_n$ of each item towards the simplex barycenter, i.e., uniform assignments.) However, as noted, nonunique solutions appear to be very rare with practical data, so this does not seem necessary.

3 Simple, Efficient Algorithm to Solve the QP

It is possible to solve problem (1) in different ways, but one must be careful in designing an effective algorithm because the number of variables and the number of constraints grows proportionally to the number of data points, and can then be very large. We describe here one algorithm that is very simple, has guaranteed convergence without line searches, and takes advantage of the structure of the problem and the sparsity of $\mathbf{L}$. It is based on the alternating direction method of multipliers (ADMM) (Boyd et al. 2011). The basic idea is to introduce new variables $\mathbf{Y}$ that replace the inequalities with an indicator function, construct the augmented Lagrangian (using a penalty parameter $\rho > 0$) and cyclically update the primal variables $\mathbf{Z}$, the auxiliary variables $\mathbf{Y}$, and the Lagrange multipliers for the constraints ($\boldsymbol{\nu}$ and $\mathbf{U}$). The success of this crucially relies on being able to compute efficiently the update of $\mathbf{Z}$, which involves a large linear system of NK equations. We capitalize on the sparsity structure of the problem by applying a direct linear solver using the Schur's complement and caching the Cholesky factorization of $\mathbf{L}$. Solving our system can be shown to be equivalent to solving K systems of N equations where the coefficient matrix is the same for each system and besides is constant and sparse, equal to $2\lambda\mathbf{L} + \rho\mathbf{I}$. In turn, these linear systems may be solved efficiently in one of the two following ways: (1) preferably, by caching the Cholesky factorization of this matrix (using a good permutation to reduce fill-in), if it does not add so much fill that it can be stored; or (2) by using an iterative linear solver such as conjugate gradients, initialized with a warm start, preconditioned, and exiting it before convergence, so as to carry out faster, inexact $\mathbf{Z}$-updates. In either case, the resulting step over $\mathbf{Z}$ becomes linear in NK.

The final algorithm is as follows. The input are the affinity matrices $\mathbf{G}_{N\times K}$ and $\mathbf{W}_{N\times N}$, from which we construct the graph Laplacian $\mathbf{L}_{N\times N}$. We then choose $\rho > 0$ and set

$$\mathbf{h} = -\tfrac{1}{K}\mathbf{G}\mathbf{1}_K + \tfrac{\rho}{K}\mathbf{1}_N \qquad \mathbf{R}\mathbf{R}^T = 2\lambda\mathbf{L} + \rho\mathbf{I}.$$

The Cholesky factor $\mathbf{R}$ is used to solve linear system (3b). We then iterate, in order, eqs. (3a)–(3d) until convergence:

$$\boldsymbol{\nu} \leftarrow \tfrac{\rho}{K}(\mathbf{Y} - \mathbf{U})\mathbf{1}_K - \mathbf{h} \tag{3a}$$

$$\mathbf{Z} \leftarrow (2\lambda\mathbf{L} + \rho\mathbf{I})^{-1}(\rho(\mathbf{Y} - \mathbf{U}) + \mathbf{G} - \boldsymbol{\nu}\mathbf{1}_K^T) \tag{3b}$$

$$\mathbf{Y} \leftarrow (\mathbf{Z} + \mathbf{U})_+ \tag{3c}$$

$$\mathbf{U} \leftarrow \mathbf{U} + \mathbf{Z} - \mathbf{Y} \tag{3d}$$

where $\mathbf{Z}_{N\times K}$ are the primal variables, $\mathbf{Y}_{N\times K}$ the auxiliary variables, $\mathbf{U}_{N\times K}$ the Lagrange multiplier estimates for $\mathbf{Y}$, and $\boldsymbol{\nu}_{N\times 1}$ the Lagrange multipliers for the equality constraints. The solution for the linear system in the $\mathbf{Z}$-update may be obtained by using two triangular backsolves if using the Cholesky factor of $2\lambda\mathbf{L} + \rho\mathbf{I}$, or using an iterative method such as conjugate gradients if the Cholesky factor is not available. Convergence of this ADMM iteration to the global minimum of problem (1) in value and to a feasible point is guaranteed for any $\rho > 0$ (Boyd et al. 2011).

Theorem 3.1. *At each iterate in updates* (3)*,* $\mathbf{Z}\mathbf{1}_K = \mathbf{1}_N$ *and* $\mathbf{U} \leq \mathbf{0}$*. Upon convergence of algorithm* (3)*,* $\mathbf{Z}$ *is a solution with Lagrange multipliers* $\boldsymbol{\pi} = -\boldsymbol{\nu}$ *and* $\mathbf{M} = -\rho\mathbf{U}$*.*

In practice, the algorithm is stopped before convergence, and $\mathbf{Z}$, $\boldsymbol{\pi} = -\boldsymbol{\nu}$ and $\mathbf{M} = -\rho\mathbf{U}$ are estimates for a solution and its Lagrange multipliers, respectively. The estimate $\mathbf{Z}$ may not be feasible, in particular the values z_{nk} need not be in $[0, 1]$, since this is only guaranteed upon convergence. If needed, a feasible point may be obtained by projecting each row of $\mathbf{Z}$ onto the simplex (see section 4). We initialize $\mathbf{Y} = \mathbf{U} = \mathbf{0}$. We stop when $\left\|\mathbf{Z}^{(\tau+1)} - \mathbf{Z}^{(\tau)}\right\|_1$ falls below a set tolerance (e.g. 10^{-5}).

Each step in (3) is $\mathcal{O}(NK)$ except for the linear system solution in (3b). If $\mathbf{L}$ is sparse, using the Cholesky factor makes this step $\mathcal{O}(NK)$ as well, and adds a one-time setup cost of computing the Cholesky factor (which is also linear in N with sufficiently sparse matrices). Thus, each iteration of the algorithm is cheap. In practice, for good values of ρ, the algorithm quickly approaches the solution in the first iterations and then converges slowly, as is known with ADMM algorithms in general. However, since each iteration is so cheap, we can run a large number of them if high accuracy is needed. As a sample runtime, for a problem with $N = 10\,000$ items and $K = 10$ categories (i.e., $\mathbf{Z}$ has 10^5 parameters) and using a 100-nearest-neighbor graph, the Cholesky factorization takes 0.5 s and each iteration takes 0.15 s in a PC. We discuss how to set the penalty parameter ρ in (Carreira-Perpiñán and Wang 2014).

4 Out-of-sample Mapping

Having trained the system, that is, having found the optimal assignments $\mathbf{Z}$ for the training set items, we are given a new, test item $\mathbf{x}$ (for example, a new point $\mathbf{x} \in \mathbb{R}^D$), along with its item-item and item-category similarities $\mathbf{w} = (w_n)$, $n = 1, \dots, N$ and $\mathbf{g} = (g_k)$, $k = 1, \dots, K$, respectively, and we wish to find its assignment $\mathbf{z}(\mathbf{x})$ to each category. Following (Carreira-Perpiñán and Lu 2007), this can be achieved by solving a problem of the form (1) with a dataset consisting of the original training set augmented with $\mathbf{x}$, but keeping $\mathbf{Z}$ fixed to the values obtained during training. Hence, the only free parameter is the assignment vector $\mathbf{z}$ for the new point $\mathbf{x}$. After dropping constant terms, the optimization problem (1) reduces to the following quadratic program over K variables:

$$\min_{\mathbf{z}} \|\mathbf{z} - (\bar{\mathbf{z}} + \gamma\mathbf{g})\|^2 \quad \text{s.t.} \quad \mathbf{z}^T\mathbf{1}_K = 1,\ \mathbf{z} \geq \mathbf{0} \tag{4}$$

$$\gamma = \tfrac{1}{2\lambda(\mathbf{1}_N^T\mathbf{w})} \qquad \bar{\mathbf{z}} = \tfrac{\mathbf{Z}^T\mathbf{w}}{\mathbf{1}_N^T\mathbf{w}} = \textstyle\sum_{n=1}^N \frac{w_n}{\sum_{n'=1}^N w_{n'}}\mathbf{z}_n$$

where $\bar{\mathbf{z}}$ is a weighted average of the training points' assignments, and so $\bar{\mathbf{z}} + \gamma\mathbf{g}$ is itself an average between this and the item-category affinities. Thus, the solution is the Euclidean projection $\Pi(\bar{\mathbf{z}} + \gamma\mathbf{g})$ of the K-dimensional vector $\bar{\mathbf{z}} + \gamma\mathbf{g}$ onto the probability simplex. This can be efficiently computed, in a finite number of steps, with a simple $\mathcal{O}(K \log K)$ algorithm (Duchi et al. 2008; Wang and Carreira-Perpiñán 2013). Computationally, assuming $\mathbf{w}$ is sparse, the most expensive step is finding the neighbors to construct $\mathbf{w}$. With large N, one should use some form of hashing (Shakhnarovich, Indyk, and Darrell 2006) to retrieve approximate neighbors quickly.

As a function of λ, the out-of-sample mapping takes the following extreme values. If $\lambda = 0$ or $\mathbf{w} = \mathbf{0}$, $z_k =$

$\delta(k - k_{\max})$ where $k_{\max} = \arg\max\{g_k,\ k = 1, \ldots, K\}$, i.e., the item is assigned to its most similar similar category (or any mixture thereof in case of ties). If $\lambda = \infty$ or $\mathbf{g} = \mathbf{0}$, $\mathbf{z} = \bar{\mathbf{z}}$, independently of $\mathbf{g}$. This is the SSL out-of-sample mapping. In between these, the out-of-sample mapping as a function of λ is a piecewise linear path in the simplex, which represents the tradeoff between the crowd ($\mathbf{w}$) and expert ($\mathbf{g}$) wisdoms. This path is quite different from the simple average of $\bar{\mathbf{z}}$ and $\mathbf{g}$ (which need not even be feasible), and may produce exact 0s or 1s for some entries.

The out-of-sample mapping offers an extra degree of flexibility to the user, who has the prerogative to set λ to favor more or less the expert vs the crowd opinion, or to explore the entire $\lambda \in [0, \infty)$ continuum. The user can also explore what-if scenarios by changing $\mathbf{g}$, given the vector $\mathbf{w}$ (e.g. how would the assignment vector look like if we think that test item $\mathbf{x}$ belongs to category k but not to category k'?).

Note that the out-of-sample mapping is nonlinear and nonparametric, and it maps an input $\mathbf{x}$ (given its affinity information) onto a valid assignment vector in the probability simplex. Hence, LASS can also be considered as *learning nonparametric conditional distributions over the categories, given partial supervision*.

5 Related Work

In semisupervised learning (SSL) with a Laplacian penalty (Zhu, Ghahramani, and Lafferty 2003), the basic idea is that we are given an affinity matrix $\mathbf{W}$ and corresponding graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ on N items, and the labels for a subset of the items. Then, the labels for the remaining, unlabeled items are such that they minimize the Laplacian penalty, or equivalently they are the smoothest function on the graph that satisfies the given labels ("harmonic" function). Call $\mathbf{Z}_u$ of $N_u \times K$ and $\mathbf{Z}_l$ of $N_l \times K$ the matrices of labels for the unlabeled and labeled items, respectively, where $N = N_l + N_u$, and $\mathbf{Z}^T = (\mathbf{Z}_u^T\ \mathbf{Z}_l^T)$. To obtain $\mathbf{Z}_u$ we minimize $\mathrm{tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)$ over $\mathbf{Z}_u$, with fixed $\mathbf{Z}_l$:

$$\begin{aligned}\min_{\mathbf{Z}_u} \mathrm{tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right) &= \min_{\mathbf{Z}_u} \mathrm{tr}\left(\begin{pmatrix}\mathbf{Z}_u\\ \mathbf{Z}_l\end{pmatrix}^T \begin{pmatrix}\mathbf{L}_u & \mathbf{L}_{ul}\\ \mathbf{L}_{ul}^T & \mathbf{L}_l\end{pmatrix}\begin{pmatrix}\mathbf{Z}_u\\ \mathbf{Z}_l\end{pmatrix}\right)\\ &= \min_{\mathbf{Z}_u} \mathrm{tr}\left(\mathbf{Z}_u^T\mathbf{L}_u\mathbf{Z}_u + 2\mathbf{Z}_l^T\mathbf{L}_{ul}^T\mathbf{Z}_u\right) + \text{constant}\\ &\Rightarrow \mathbf{Z}_u = -\mathbf{L}_u^{-1}\mathbf{L}_{ul}\mathbf{Z}_l = \mathbf{L}_u^{-1}\mathbf{W}_{ul}\mathbf{Z}_l. \quad (5)\end{aligned}$$

Thus, computationally the solution involves a sparse linear system of $N_u \times N_u$. An out-of-sample mapping for a new test item $\mathbf{x}$ with affinity vector $\mathbf{w}$ wrt the the training set can be derived by SSL again, taking $\mathbf{Z}_l$ of $N \times K$ as all the trained labels (given and predicted) and $\mathbf{Z}_u = \mathbf{z}^T$ as the free label. This gives a closed-form expression $\mathbf{z}(\mathbf{x}) = \sum_{n=1}^{N} \frac{w_n}{\sum_{n'=1}^{N} w_{n'}} \mathbf{z}_n$ which is the average of the labels of $\mathbf{x}$'s neighbors, making clear the smoothing behavior of the Laplacian. SSL with a Laplacian penalty is very effective in problems where there are very few labeled items, i.e., $N_u \gg N_l$, but the graph structure is highly predictive of each item's labels. Essentially, the given labels are propagated throughout the graph.

In the special case where the given labels are valid assignments (nonnegative with unit sum), the labels predicted by SSL will also be valid assignments, and we need not subject the problem explicitly to simplex constraints, which simplifies it computationally. This occurs in the standard semisupervised classification setting where *each item belongs to only one category* and we use the $\mathbf{z}_n$ vectors to implement a 1-of-K coding (e.g. as used for supervised clustering in (Grady 2006)). However, *in general SSL does not produce valid assignments*, e.g. if the given labels are not valid assignments, or in other widely used variations of SSL, such as using class mass normalization (Zhu, Ghahramani, and Lafferty 2003), or using the normalized graph Laplacian instead of the unnormalized one, or using label penalties (Zhou et al. 2004). In the latter case (also similar to the "dongle" variation of SSL; (Zhu, Ghahramani, and Lafferty 2003)), one minimizes the Laplacian penalty plus a term equal to the squared distance of the labeled points (considered free parameters as well) to the labels $\mathbf{Z}_l$ provided. Thus, this penalizes the labeled points from deviating from their intended labels, rather than forcing them to equal them. This was extended by (Subramanya and Bilmes 2011) (replacing squared losses with Kullback-Leibler divergences and adding an additional entropy term) to learning probability distributions, i.e., where the labels $\mathbf{Z}_l$ are entire distributions over the K classes, with each item-class probability specified exactly. All these approaches rely on the following: *they use provided, specific label values $\mathbf{Z}_l$ as targets to be (ideally) met by the parameters*.

Relation with LASS LASS and SSL are similar in that (1) $\mathbf{L}$ plays the same role, i.e., to propagate label information in a smooth way according to the item-item graph; and (2) both rely on some given data to learn $\mathbf{Z}$: the similarity matrix $\mathbf{G}$ in LASS and the given labels $\mathbf{Z}_l$ in SSL. LASS and SSL differ as follows. (1) The use of the simplex constraints, necessary to ensure valid assignments, which also means all the assignment values in LASS are interdependent, unlike in the classical SSL, where the prediction for each category can be solved independently. (2) A fundamental difference is in the supervision provided. If in LASS we were given actual labels $\mathbf{Z}_l$ for some of the items, we would simply use them just as in SSL, and the LASS problem with $\mathbf{Z}_l$ having given assignments would be:

$$\begin{aligned}\min_{\mathbf{Z}_u} \quad & \lambda\,\mathrm{tr}\left(\mathbf{Z}_u^T\mathbf{L}_u\mathbf{Z}_u + 2\mathbf{Z}_l^T\mathbf{L}_{ul}^T\mathbf{Z}_u\right) - \mathrm{tr}\left(\mathbf{G}_u^T\mathbf{Z}_u\right)\\ \text{s.t.} \quad & \mathbf{Z}_u\mathbf{1}_K = \mathbf{1}_{N_u},\ \mathbf{Z}_u \geq \mathbf{0}.\end{aligned}$$

However, the $\mathbf{G}$ term provides soft, partial "labels", and this information differs from (hard) labels $\mathbf{Z}_l$. Indeed, when the label to be learned for each item is an assignment vector, the concept of "labeling" breaks down, for two reasons. First, if the number of categories K is not very small and an item n has nonzero assignments to multiple categories, in practice it is hard for a user to have to give a value (or tag) for every single relevant category. Giving partial information is much easier, by simply setting $g_{nk} = 1$ for the most relevant categories, possibly setting $g_{nk} = -1$ for a few categories, and setting $g_{nk} = 0$ for the rest (we stick to ± 1 and 0 similarities for simplicity). Second, because the assignment values are constrained to be in the simplex, *we cannot give actual values for individual entries* (unless we give the

entire assignment vector). For example, setting an entry to 1 implicitly forces the other entries to 0. In summary, *the semantics of the item-category similarities in LASS is that, where nonzero, they encourage the corresponding assignment towards relatively high or low values (for positive and negative similarities, respectively), and where zero, they reflect ignorance and are non-committing*, something which is close to a user's intuition, but generally difficult to achieve by setting assignment values directly.

Not being able to commit to specific assignment values, especially where $g_{nk} = 0$, also implies that it is not possible to transform meaningfully a given item-category sparse similarity matrix $\mathbf{G}$ into an assignment vector. Given a sparse matrix $\mathbf{G}$, if we insist in setting full assignments for each item n having a nonzero vector $\mathbf{g}_n$ (so we can use these with SSL), perhaps the best one can do is to follow this labeling procedure: for each k, set $z_{nk} = 1$, ϵ or 0 if $g_{nk} = 1$, 0 or -1, respectively, and normalize $\mathbf{z}_n$ (where $1 \gg \epsilon \geq 0$ is a smoothing user parameter). This forces a zero assignment for each negative-similarity category, and distributes the unit assignment mass over the remaining categories. Obviously, this likely forces many z_{nk} to wrong values and, as we show in the experiments, works poorly.

The difference between SSL and LASS is clearly seen in the out-of-sample mapping. In SSL, the information provided for a test item is just the vector $\mathbf{w}$ of similarities to other items, and the SSL out-of-sample mapping coincides with $\bar{\mathbf{z}}$ in the LASS out-of-sample mapping, i.e., the average of its neighbors' assignments. With LASS, in addition to $\mathbf{w}$ we also give the vector $\mathbf{g}$ of similarities to categories. If $\mathbf{g} = \mathbf{0}$, the predictions of LASS and SSL coincide. Otherwise, LASS trades off both $\mathbf{w}$ and $\mathbf{g}$. This is particularly important when $\mathbf{w}$ is not very informative, e.g. if it has many nonzero entries of similar magnitude, or all entries are very small (an outlying or "new" item).

6 Experiments

We test LASS in: (1) a single-label task (digit recognition), where using assignment vectors is not strictly necessary, but LASS still achieves the best classification performance. (2) Document categorization and image tagging, where learning assignment vectors and using partial labels is necessary.

Digit recognition We test LASS in a classification task where each data point has only one valid label. We randomly sample 10 000 MNIST digit images (http://yann.lecun.com/exdb/mnist) and compute the 10-nearest-neighbor graph with similarities based on the Euclidean distance between images. We then randomly select N_l images from each of the 0–9 categories and give them the correct label. We compare with a nearest neighbor classifier (NN), one-vs-all kernel support vector machine (KSVM) using RBF kernel of width $\sigma = 5$ and hinge loss penalty parameter C selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3\}$ and SSL (Zhu, Ghahramani, and Lafferty 2003; Grady 2006) using 1-out-of-10 coding for the labeled points. We also include two variants of SSL: SSL1 (Zhu, Ghahramani, and Lafferty 2003) normalizes the assignments from SSL so that the prior distribution of the different classes is respected. SSL2 uses the normalized graph Laplacian instead of the unnormalized one in SSL (Zhou et al. 2004). SSL1 and SSL2 improve over SSL but neither of them produce valid assignments (they do not lie on the probability simplex). For LASS and SSL/1/2 we assign each point to the category with the highest prediction value. We let all algorithms use their respective optimal parameters (e.g. λ in LASS is determined by a grid search).

Fig. 1(left) shows the classification error over 20 different labeled/unlabeled partitions of the dataset as a function of N_l (errorbars not shown to avoid clutter). The accuracy of all algorithms improves as the number of labeled points increase, particularly for NN and KSVM, which are template matchers. But when only few points are labeled, the methods that make use of Laplacian smoothing significantly outperform them. LASS (runtime: 40 s) consistently achieves the best accuracy while producing valid assignments.

Document categorization We want to predict assignments of documents to topics, where each document may belong to multiple topics, in the 20-newsgroups dataset of the UCI KDD Archive ($N = 11\,269$ documents). We manually add 7 new topics (comp.sys, rec.sport, computer, recreation, politics, science and religion) based on the hierarchical structure and perceived similarity of groups (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware). This yields $K = 27$ topics and each document can belong to 1 to 3 topics. To construct feature vectors, we remove words that occur in 5 or fewer documents, and then extract the TFIDF (term frequency $\times$ inverse document frequency) feature of the documents. We generate the $\mathbf{G}$ similarity matrix by randomly selecting N_l documents from each of the 27 topics, and giving each document one $+1$ label (the topic it is selected from) and five -1 labels (topics it does not belong to). For SSL, we turn this into assignment labels as described in section 5, and select the smoothing parameter ϵ optimally from $\{0,\ 0.005,\ 0.01,\ 0.02,\ 0.05,\ 0.1,\ 0.2\}$.

To evaluate the performance, we select for each test document the T topics to which it has highest predicted assignments (where $T \in \{1, 2, 3\}$ is the actual number of topics this document belongs to), consider them as predicted label set, and compare them with ground truth labels. We consider it as an error if the predicted label set and the ground truth set differ. NN classification does not apply here because no document is fully labeled. We can apply one-vs-all linear SVM (hinge loss penalty parameter C selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3\}$) because we do have

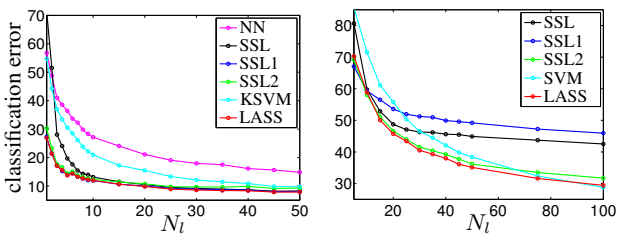

Figure 1: Classification error (%) vs number of labeled points for each class on MNIST (left) and 20-newsgroups (right) datasets.

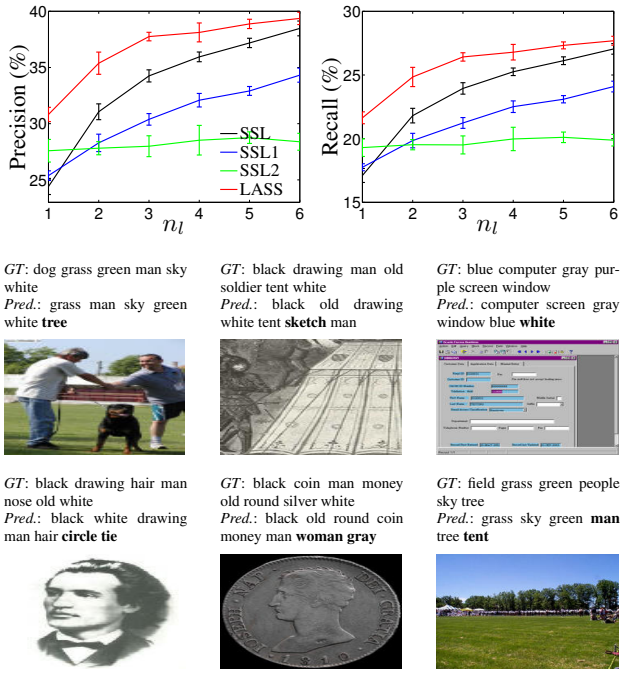

Figure 2: Results on ESP game dataset. *Top*: precision and recall vs number of positive affinities n_l, at a fixed annotation length 5, averaged over 20 runs. *Bottom*: sample predictions of LASS on test images when $n_l = 5$. We show the highest predictions (*Pred.*, in assignment order) up to the number of tags in the ground truth (*GT*, unordered), with **mismatches** in boldface.

training points for each topic. Fig. 1(right) shows the mean classification error over 20 random labeled/unlabeled partitions of the dataset as a function of N_l. The accuracy of all algorithms again improves when N_l increases. LASS outperforms all other algorithms at nearly all N_l values, and, unlike them, always produces valid assignments.

Image tagging This task fully benefits from the ability of LASS to handle partial labels, and predict full assignment vectors. In the problem of image tagging, each image can typically be tagged with multiple categories of a large number of possible categories. However, a user will usually tag only a few of the relevant categories for it (e.g. out of laziness) and will miss tagging other categories that would be relevant too. The task given a test image is to predict an assignment vector, i.e., to fill in "soft tags", for all categories. When there are many possible categories, trying to fill in the missing assignments for the partially labeled samples (so we can use SSL to propagate them to the unlabeled samples) is pointless. In contrast, LASS does not require these missing assignments, by conveniently providing zero affinities.

We demonstrate LASS on a subset of the ESP game (von Ahn and Dabbish 2004) images used by (Guillaumin et al. 2009). We select the images in the training set that are tagged with at least 6 categories (words), resulting in 6 100 images with a total of 267 non-empty categories, with 7.2 categories per image on average. We use the same image feature sets as (Guillaumin et al. 2009) to compute distances between images and build a 10-nearest neighbor graph. We give partial information for 4 600 images and provide item-category affinities for each image in the following way: we give positive affinity $(+1)$ for a random subset of size n_l from the categories it is tagged with, and give negative affinities (-1) randomly for 5 out of the 20 most frequent categories it is not tagged with. Providing negative affinities in this way stops the algorithm from concentrating most of the probability mass on the most frequent categories. The other 1 500 images are completely unlabeled and used for testing. SSL fills in missing assignments of the partially labeled samples as described in the document categorization task. Parameters are selected based on grid search for each algorithm.

We evaluate the performance of different algorithms using the precision, recall and F–1 score (averaged over sample images) on the test samples while fixing the annotation length at 5, i.e., each image is tagged with the 5 categories of highest assignment. (Although LASS admits tags as similarity values at test time, we do not use them here.) We vary the number of positive tags n_l from 1 to 6. Fig. 2 shows the results for SSL and LASS over 20 runs (each with a different random selection of test set and partial affinities). We could not run one-versus-all SVMs because there are no negative samples for most categories. In SSL2, the highest prediction values are nearly always the most frequent categories. We see that LASS greatly improves over SSL, especially when smaller numbers of positive affinities are given.

7 Conclusion

We have proposed a simple quadratic programming model for learning assignments of items to categories that combines two complementary and possibly conflicting sources of information: the crowd wisdom and the expert wisdom. It is particularly attractive when fully labeling an item is impractical, or when categories have a complex structure and items can genuinely belong to multiple categories to different extents. It provides a different way to incorporate supervision to that of traditional semisupervised learning, which is ill-suited for this setting because the similarity information cannot be faithfully transformed into assignment labels.

We expect LASS to apply to problems beyond semisupervised learning, such as clustering, and in social network applications, with image, sound or text data that is partially tagged. It can also be extended to handle tensor data or have additional terms in its objective, for example to represent relations between categories with a category-category similarity matrix, or even to use negative similarities in the item-item graph. A further application of LASS is to learn probability distributions that are conditional on partial supervisory information, since, in effect, the out-of-sample mapping is a nonparametric mapping from the affinity information to a distribution over categories. Another research direction is to accelerate the convergence of the training algorithm, particularly with large datasets with many categories, where we may expect each row of $\mathbf{Z}$ to be sparse.

References

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6):1373–1396.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Machine Learning Research* 7:2399–2434.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Carreira-Perpiñán, M. Á., and Lu, Z. 2007. The Laplacian Eigenmaps Latent Variable Model. In Meilă, M., and Shen, X., eds., *Proc. of the 11th Int. Workshop on Artificial Intelligence and Statistics (AISTATS 2007)*, 59–66.

Carreira-Perpiñán, M. Á., and Wang, W. 2014. LASS: A simple assignment model with Laplacian smoothing. Unpublished manuscript, arXiv.

Carreira-Perpiñán, M. Á. 2006. Fast nonparametric clustering with Gaussian blurring mean-shift. In Cohen, W. W., and Moore, A., eds., *Proc. of the 23rd Int. Conf. Machine Learning (ICML'06)*, 153–160.

Chapelle, O.; Schölkopf, B.; and Zien, A., eds. 2006. *Semi-Supervised Learning*. Adaptive Computation and Machine Learning Series. MIT Press.

Chung, F. R. K. 1997. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. Providence, RI: American Mathematical Society.

Doyle, P. G., and Snell, J. L. 1984. *Random Walks and Electric Networks*, volume 22 of *Carus Mathematical Monographs*. Mathematical Association of America.

Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; and Chandra, T. 2008. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In McCallum, A., and Roweis, S., eds., *Proc. of the 25th Int. Conf. Machine Learning (ICML'08)*, 272–279.

Grady, L. 2006. Random walks for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(11):1768–1783.

Guillaumin, M.; Mensink, T.; Verbeek, J.; and Schmid, C. 2009. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proc. 12th Int. Conf. Computer Vision (ICCV'09)*, 309–316.

Lu, Z., and Carreira-Perpiñán, M. Á. 2008. Constrained spectral clustering through affinity propagation. In *Proc. of the 2008 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'08)*.

Shakhnarovich, G.; Indyk, P.; and Darrell, T., eds. 2006. *Nearest-Neighbor Methods in Learning and Vision*. Neural Information Processing Series. Cambridge, MA: MIT Press.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22(8):888–905.

Subramanya, A., and Bilmes, J. 2011. Semi-supervised learning with measure propagation. *J. Machine Learning Research* 12:3311–3370.

Taubin, G. 1995. A signal processing approach to fair surface design. In Mair, S. G., and Cook, R., eds., *Proc. of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1995)*, 351–358.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proc. ACM Int. Conf. Human Factors in Computing Systems (CHI 2004)*, 319–326.

Wang, W., and Carreira-Perpiñán, M. Á. 2010. Manifold blurring mean shift algorithms for manifold denoising. In *Proc. of the 2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'10)*, 1759–1766.

Wang, W., and Carreira-Perpiñán, M. Á. 2013. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. Unpublished manuscript, arXiv:1309.1541.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In Thrun, S.; Saul, L. K.; and Schölkopf, B., eds., *Advances in Neural Information Processing Systems (NIPS)*, volume 16, 321–328. MIT Press, Cambridge, MA.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In Fawcett, T., and Mishra, N., eds., *Proc. of the 20th Int. Conf. Machine Learning (ICML'03)*, 912–919.