Multi-Label Learning on Tensor Product Graph

Jonathan Q. Jiang
City University of Hong Kong, Kowloon, Hong Kong
qiajiang@cityu.edu.hk

Abstract

A large family of graph-based semi-supervised algorithms have been developed intuitively and pragmatically for the multi-label learning problem. These methods, however, only implicitly exploited the label correlation, as either part of graph weight or an additional constraint, to improve overall classification performance. Despite their seemingly quite different formulations, we show that all existing approaches can be uniformly referred to as a Label Propagation (LP) or Random Walk with Restart (RWR) on a Cartesian Product Graph (CPG). Inspired by this discovery, we introduce a new framework for multi-label classification task, employing the Tensor Product Graph (TPG)—the tensor product of the data graph with the class (label) graph—in which not only the intra-class but also the inter-class associations are explicitly represented as weighted edges among graph vertices. In stead of computing directly on TPG, we derive an iterative algorithm, which is guaranteed to converge and with the same computational complexity and the same amount of storage as the standard label propagation on the original data graph. Applications to four benchmark multi-label data sets illustrate that our method outperforms several state-of-the-art approaches.

Introduction

Multi-label learning problem are ubiquitous in real-world applications, ranging from image classification (Boutell et al. 2004), to text categorization (Schapire and Singer 2000; McCallum 1999), and to functional genomics (Elissee and Weston 2002). In all these cases, each instance is associated with a set of labels (classes) which are interdependent rather than mutually exclusive. For example, “face” and “body” are often simultaneously assigned to a certain video clip, and, contrariwise, “bonfire” and “waterscape” are generally never co-appear on an image at the same time. Consequently, the pivotal challenge for developing multi-label learning algorithms lies in how to utilize class (label) correlation to facilitate class-membership inference and thus to improve classification accuracy.

Research of multi-label learning was initially motivated by the difficulty of concept ambiguity encountered in text categorization, where each document is possibly involved in several predefined topics (McCallum 1999; Schapire and Singer 2000). Soon after, various well-established tricks originally developed in single-label learning have been employed to promote multi-label classification studies, such as graph-based algorithms (Chen et al. 2008; Zha et al. 2008; Wang, Huang, and Ding 2009), discriminative embedding (Ji et al. 2008; Zhang and Zhou 2008), nonnegative matrix factorization (Liu, Jin, and Yang 2006), maximum entropy model (Zhu et al. 2005), and many others.

Among these schemes, graph-based methods have attracted the most attention due mainly to their intrinsic superiority of data representation and visualization (Zhou et al. 2004; Lafon and Lee 2006). Despite the achieved progress, there are still many open questions. In particular, all existing approaches implicitly incorporate label correlation into standard label propagation algorithms, as either part of the graph weights (Kang, Jin, and Sukthankar 2006; Chen et al. 2008) or an additional constraint (Wang, Huang, and Ding 2009; Zha et al. 2008). How could we exploit label correlation explicitly in graph model? Is there a common principle behind the existing algorithms designed intuitively and pragmatically with different purposes? Furthermore, it is worthwhile to note that the available information are not fully leveraged in standard label propagation process since label correlation induce another graph, i.e., class graph (Zhou et al. 2007).

In this work, we attempt to explicitly utilize the total information leveraged by the instances and their associated labels. The contribution of this paper is three-fold:

- We show that after certain appropriate reformulation, the existing graph-based semi-supervised multi-label learning approaches can be uniformly referred to as a Label Propagation (LP) process or Random Walk with Restart (RWR) on a Cartesian Product Graph (CPG).
- A novel algorithm called MLTPG (Multi-label Learning on Tensor Product Graph) is proposed. Unlike many existing approaches, both the intra-class and inter-class associations are explicitly represented as the weighted links in a Tensor Product Graph (TPG) of the data graph with the class graph. Since TPG takes into account the higher-order relationships, it comes at no surprise that the classification performance is significantly improved.
- We never compute the label spreading process directly on the TPG. Instead, we derive an iterative algorithm with the same amount of storage as the standard propagation on the data graph, which is guaranteed to coverage.

Preliminaries

Basic Graph Theory An *undirected graph* $G(V, E)$ consists of a finite *vertex set* V, an *edge set* E and a *weighted adjacency matrix* $W \in \Re^{|V|\times|V|}$ which associates a positive scalar $w(x, y)$ with each edge $(x, y) \in E$. Given a specific vertex x, the degree function $d : V \rightarrow \Re^{+}$ is defined by $d_x = \sum_{y:(x,y)\in E} w(x, y)$. Let $D = \mathrm{diag}(d_x)_{x\in V}$ be the diagonal *degree matrix*, the weighted adjacency matrix W can be symmetrically or asymmetrically normalized as $W_s = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ and $W_r = D^{-1}W$, which corresponds to the standard LP process (Zhou et al. 2004) or RWR (Chung 1997), respectively. In the existing literature, there are three related matrices called the *graph Laplacian*, and there does not appear to be a consensus on the nomenclature (von Luxburg 2007). The *unnormalized graph Laplacian* L is defined as $L = D - W$. There are also two normalized graph Laplacian (Chung 1997), respectively given by

$$L_s = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - W_s \quad L_r = D^{-1}L = I - W_r$$

Product Graph Given two graphs $G'(V', E')$ and $G''(V'', E'')$, a *graph product* is a certain kind of binary operation that produces a graph $G(V, E)$ with $|V'||V''|$ vertices, each representing a pair of vertices from G' and G'', respectively. An edge exists in E iff the corresponding vertices satisfy conditions of a certain type[1]. Specifically, in the *Cartesian product graph* $G^{\square} = G'\square G''$, $x^{\square} = (x', x'')$ adjacent with $y^{\square} = (y', y'')$ whenever $x' = y'$ and $(x'', y'') \in E''$ or $x'' = y''$ and $(x', y') \in E'$; an edge exists in the *tensor product graph* $G^{\times} = G' \times G''$ iff the corresponding vertices are adjacent in both G' and G'', respectively. Thus, $E^{\times} = \{((x', y'), (x'', y'')) : (x', y') \in E' \wedge (x'', y'') \in E''\}$. We refer the reader to (Harary 1994; Knuth 2008) for further details on product graphs and their properties and only review a necessary lemma which serves as important foundation for our following analysis

Lemma 1. *If W' and W'' are the weighted adjacency matrices of graphs G' and G'', respectively, the weighted adjacency matrix of the Cartesian product graph $G^{\square}$ is $W^{\square} = W' \oplus W''$, and the weighted adjacency matrix of the tensor product graph $G^{\times}$ is $W^{\times} = W' \otimes W''$.*

where$\oplus$ and $\otimes$ denote the Kronecker sum and Kronecker product, respectively. They are linked by the well-known property: let $A \in \Re^{n\times n}$, $B \in \Re^{m\times m}$, and I_k denote the $k \times k$ identity matrix, then $A \oplus B = A \otimes I_m + I_n \otimes B$.

Notations A^T, $\mathrm{Tr}(A)$, and $\rho(A)$ denote the transpose, the trace, and the spectral radius of matrix A. $\vec{A}$ represents the vectorization operator $\mathrm{vec}(A)$ that stacks the columns of matrix A one after the next into a column vector. $\|\cdot\|$ and $\|\cdot\|_F$ denotes the ℓ_2-norm and F-norm, respectively. I always denotes the identity matrix, its dimensions being apparent from the context. The matrices $\mathcal{L}$ and $\mathcal{W}$ generally represent the appropriate form of the graph Laplacian and weighted adjacency matrix.

Graph-based Semi-supervised Learning: A Unified View

Problem Formulation Let $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ be partially labeled data set with K classes, where each instance $\boldsymbol{x}_i \in \Re^d$ is associated with a subset of class labels represented by a binary vector $\boldsymbol{y}_i \in \{0, 1\}^K$ such that $\boldsymbol{y}_i(k) = 1$ if $\boldsymbol{x}_i$ belongs to the k-th class, and 0 otherwise. Let $G = (V, E)$ be an undirected graph over the data set, where each vertex corresponds to an instance, and each edge has a non-negative weight $w(i, j)$ typically reflecting the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. For convenience, we write $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]^T$ and $Y = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n]^T$. Suppose the first l instances are already annotated, our goal is to predict labels $\{\boldsymbol{y}_i\}_{i=l+1}^n$ for unlabeled samples $\{\boldsymbol{x}_i\}_{i=l+1}^n$. In most cases, the learning system will generate a real-value matrix $F = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n]^T = [\boldsymbol{f}^1, \ldots, \boldsymbol{f}^K] = (f_{ik}) \in \Re^{n\times K}$ where f_{ik} represents the likelihood of $\boldsymbol{x}_i$ belonging to the k-th class. The learning problem on a partially labeled graph G generally can be thought of seeking for a classification function $f : V \rightarrow \Re^K$, which is sufficient smooth on closely related vertices, while simultaneously changes the initial label assignment as little as possible. This view can be formulated as the following optimization problem (Belkin, Matveeva, and Niyogi 2004; Zhou and Schölkopf 2004)

$$\arg\min_{f\in\mathcal{H}(V)} \left\{\mathcal{S}_G(f) + \mu\|f(\boldsymbol{x}) - \boldsymbol{y}\|_F^2\right\} \tag{1}$$

Single-label learning If $\sum_{k=1}^K Y_{ik} = 1$, i.e., each instance belongs to exactly one class, the data set is a single-label data set. Traditional single-label graph-based semi-supervised learning approaches (Zhou et al. 2004; Wang and Zhang 2006; Zhu, Ghahramani, and Lafferty 2003) only take into account the data graph G beased on certain pairwise similarity (or equivalently distance) metrics. In this case, the smoothness term is usually formulated as

$$\mathcal{S}_G = \sum_{k=1}^{K} \mathcal{S}^k(\boldsymbol{f}^k) = \sum_{k=1}^{K} \boldsymbol{f}^{k^T}\mathcal{L}\boldsymbol{f}^k = \mathrm{Tr}\left(F^T(I - \mathcal{W})F\right) \tag{2}$$

Substituting Eq. (3) into regularization framework and differentiating it with respect to F, we get the iterative solution (Zhou et al. 2004)

$$F^{(t+1)} = \alpha\mathcal{W}F^{(t)} + (1 - \alpha)Y \tag{3}$$

where $\alpha = \frac{1}{1+\mu}$. Introducing an isomorphism $\Re^{n\times K} = \Re^n \otimes \Re^K \cong \Re^{nK}$ between the spaces of matrix and vector in coordinates to convert the matrix F into a column vector $\vec{F} = [\boldsymbol{f}^1, \ldots, \boldsymbol{f}^K]^T$ and vectorizing both sides

$$\vec{F}^{(t+1)} = \alpha\left(\mathcal{W} \otimes I\right)\vec{F}^{(t)} + (1 - \alpha)\vec{Y} \tag{4}$$

The matrix $\mathcal{W} \otimes I$ is the adjacency matrix of a special product graph which can be understood as K independent copies of the data graph G. Therefore, the single-label methods

[1] There are $3 \times 3 - 1 = 8$ cases to be decided (three possibilities for each, with the case where both are equal eliminated) and thus there are $2^8 = 256$ different types of graph products that can be defined.

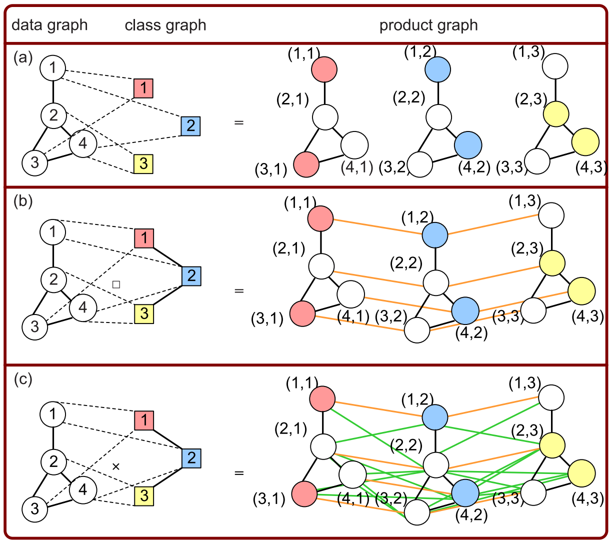


Figure 1: The key principles of different graph-based semi-supervised learning algorithms. The symbol (i, k) denotes that the instance $\boldsymbol{x}_i$ belongs to the k-th class.

perform the binary classification on K individual copies of the data graph, each copy for one specific class (see Fig. 1(a)). The existing approaches rely on an almost identical formulation but only differ in details. In particular, the Local and Global Consistency (LGC) method (Zhou et al. 2004) utilized the symmetrically normalized adjacency matrix $\mathcal{W} = W_s$ of the data graph G, which can be understood as a label propagation process; the Gaussian Random Field and harmonic function (GRF) method (Zhu, Ghahramani, and Lafferty 2003) used the adjacency matrix corresponding to unlabeled data, which is also a label propagation process under the condition $\alpha \to 0(\mu \to \infty)$; the Linear Neighborhood Propagation (LNP) method (Wang and Zhang 2006) constructed the adjacency matrix through local linear coefficients and exploited the asymmetrically normalized adjacency matrix $\mathcal{W} = W_r$ that corresponds to a personalized random walk.

Multi-label learning If $\sum_{k=1}^{K} Y_{ik} \geq 1$, i.e., each instance is possibly associated with more than one class label, the data set is a multi-label data set. In this scenario, the label correlation among different classes induces another graph, the class graph $G'(V', E')$, where $V' = \{1, \ldots, K\}$ and $E' \subseteq V' \times V'$. Zha et al. introduced two multi-label learning algorithms, called Multi-label Gaussian harmonic function (MLGF) method, and Multi-label local global consistency (MLCF) method, by incorporating the correlations among multiple labels into the single-label approaches GRF and LCF respectively. Let $\mathcal{W}'$ be the appropriate adjacency matrix of class graph G', the solution of MLCF is[2]

$$(1 - \mathcal{W})F + \mu(F - Y) - \beta F \mathcal{W}' = 0 \tag{5}$$

Let $\alpha = \frac{1}{1+\mu}$ and we have an iterative equation

$$F^{(t+1)} = \alpha(\mathcal{W}F^{(t)} + \beta F^{(t)}\mathcal{W}') + (1 - \alpha)Y \tag{6}$$

Vectorizing both sides, it yields

$$\vec{F}^{(t+1)} = \alpha(I \otimes \mathcal{W} + \beta \mathcal{W}' \otimes I)\vec{F}^{(t)} + (1 - \alpha)\vec{Y}$$

$$\vec{F}^{(t+1)} = \alpha(\beta \mathcal{W}' \oplus \mathcal{W})\vec{F}^{(t)} + (1 - \alpha)\vec{Y} \tag{7}$$

where $\mathcal{W} = W_s$ and $\mathcal{W}' = W'$ in MLCF. The relationship between MLCF and MLGF is similar to the relationship between LGC and GRF. Thus, MLGF is the same as MLCF except for $\mathcal{W}$ being the adjacency matrix of unlabeled data and $\alpha \to 0$.

In addition, Chen et al. proposed a new multi-label learning approach by solving a Sylvester equation (SMSE). The solution is (Chen et al. 2008)

$$(1 - \mathcal{W})F + \mu(F - Y) + \nu F(1 - \mathcal{W}') = 0 \tag{8}$$

Let $\alpha = \frac{\mu+\nu}{1+\mu+\nu}$ and $\beta = \frac{\nu}{1+\nu}$ and we have

$$F^{(t+1)} = \alpha((1 - \beta)\mathcal{W}F^{(t)} + \beta F^{(t)}\mathcal{W}') + (1 - \alpha)Y \tag{9}$$

Likewise, it equals to

$$\vec{F}^{(t+1)} = \alpha(\beta \mathcal{W}' \oplus (1 - \beta)\mathcal{W})\vec{F}^{(t)} + (1 - \alpha)\vec{Y} \tag{10}$$

where $\mathcal{W} = W_s$ and $\mathcal{W}' = W_s'$ in SMSE.

According to Lemma 1, we can conclude that the existing multi-label methods also exploit an almost identical framework, i.e., label propagation on a CPG which forms a crosstalk of data graph and class graph. More specifically, the CPG is horizontally K copies of data graph G, each for one class; vertically, it is n copies of class graph G', each for one instance (see Fig.1(b)).

The MLTGP Algorithm

Motivations Let us first consider a motivating example in Fig.1 where are 4 instances and 3 classes marked by red, blue and yellow respectively. The initial label assignments are represented as the dot-dashed lines. Here, we attempt to predict whether the instance $\boldsymbol{x}_2$ should be associated with the 1st class. In standard single-label propoagation, $\boldsymbol{x}_2$ can barely receive the intra-class label information from its neighbors $\boldsymbol{x}_1$, $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ through the black edges within the first copy of data graph. In existing multi-label learning framework, the knowledge that instances $\boldsymbol{x}_1$ and $\boldsymbol{x}_4$ annotated with the 2nd class can also provide certain evidence for the existence of an association between instance $\boldsymbol{x}_2$ with the 1st class since there is a strong correlation between the 1st class and the 2nd class. *Besides, the instance $\boldsymbol{x}_3$ possibly gives an additional clue to this prediction as long as it can receive adequate information during the propagation process to confirm that it is likely to belong to the 2nd class.* This motivates us to develop a learning algorithm to fully leverage all the provided label information.

[2] Our formulation seems different from the original one but they are essentially identical.

Formulation Inspired by the above-mentioned example, we design the smoothness term as

$$\mathcal{S}(\boldsymbol{f}) = \mathcal{S}_G(\boldsymbol{f}) + \nu\mathcal{S}_{G'}(\boldsymbol{f}) + \omega\mathcal{S}_{G^\times}(\boldsymbol{f}) \qquad (11)$$

The first two terms require the classification function should be sufficient smooth on data graph and class graph. The third term enforces that the classification function should vary slowly on the closely related regions on the TPG. That is, each association between instance and class is likely to be paried with some known association(s) with their instances and classes closely related in the data graph and class graph, respectively. Using the symmetrically normalized graph Laplacian[3], we specify the three terms as

$$\mathcal{S}_G = \mathrm{Tr}(F^T(I - W_s)F) \qquad (12)$$

$$\mathcal{S}_{G'} = \sum_{i=1}^{n} \boldsymbol{f}_i L'_s \boldsymbol{f}_i^T = \mathrm{Tr}(F(I - W'_s)F^T) \qquad (13)$$

$$\mathcal{S}_{G^\times} = \vec{F}^T L_s^\times \vec{F} = \vec{F}^T(I - W'_s \otimes W_s)\vec{F} \qquad (14)$$

Substituting Eq. (11)-(14) into Eq. (1) and differentiating it with respect to F, we have

$$\begin{aligned}&(I - I \otimes W_s)\vec{F} + \nu(I - W'_s \otimes I)\vec{F} \\ &+\omega(I - W'_s \otimes W_s)\vec{F} + \mu(\vec{F} - \vec{Y}) = 0 \qquad (15)\end{aligned}$$

Let $\alpha = \frac{1+\nu+\omega}{1+\mu+\nu+\omega}$, $\beta = \frac{1}{1+\nu+\omega}$ and $\gamma = \frac{\nu}{1+\nu+\omega}$, then we can get an iterative equation after the simple reformulations as in (Zhou et al. 2004)

$$\vec{F}^{(t+1)} = \alpha\widetilde{W}\vec{F}^{(t)} + (1-\alpha)\vec{Y} \qquad (16)$$

where $\widetilde{W} = \beta I \otimes W_s + \gamma W'_s \otimes I + (1-\beta-\gamma)W'_s \otimes W_s$ is actually the normalized adjacency matrix of the TPG. This iterative computing is possibly impractical, especially for large graphs, by the reason of high computational complexity and storage requirement. To be exact, the order of the product graph is $O(nK)$, whereas the order of original data graph is merely $O(n)$. Fortunately, the following lemma holds (Magnus and Neudecker 1999)

Lemma 2. *Given three matrices $A \in \Re^{k\times l}$, $B \in \Re^{l\times m}$, and $C \in \Re^{m\times n}$, then*

$$\overrightarrow{ABC} = \left(C^T \otimes A\right)\vec{B} \quad \overrightarrow{AB} = (I \otimes A)\vec{B}$$

So, Eq. (17) can be reformulated as

$$\begin{aligned}F^{(t+1)} \quad &= \quad \alpha(\beta W_s F^{(t)} + \gamma W'_s F^{(t)} \\ &\qquad +(1-\beta-\gamma)W_s F^{(t)} W'_s) + (1-\alpha)Y(17)\end{aligned}$$

Obviously, this iterative algorithm is with the same computational complexity and the same amount of storage as the standard single-label learning on the data graph. The pseudo-code of the whole algorithm is summarized in Algorithm 1. Now, we shall discuss the convergence as well as its rate which are given in the following lemma and theorem.

Lemma 3. *Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $A \in \Re^{n\times n}$, and $\mu_1, \ldots, \mu_m$ be those of $B \in \Re^{m\times m}$, then the eigenvalues of $A \otimes B$ are $\lambda_i\mu_j$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.*

Lemma 4. *Let $A_1 = I \otimes W_s$, $A_2 = W'_s \otimes I$, and $A_3 = W'_s \otimes W_s$, then the set $\{A_1, A_2, A_3\}$ is commute.*

Proof. On one hand, $A_1A_2 = (I \otimes W_s)(W'_s \otimes I) = (I \times W'_s) \otimes (W_s \times I) = W'_s \otimes W_s$, and on the other hand, $A_2A_1 = (W'_s \otimes I)(I \otimes W_s) = (W'_s \times I) \otimes (I \times W_s) = W'_s \otimes W_s$. Thus, we have $A_1A_2 = A_2A_1$. Similarly, we can verify $A_iA_j = A_jA_i$ for other pairs i and j which holds the lemma. □

Theorem 1. *Algorithm 1 converges to the solution $\vec{F}^* = \left(I - \alpha\widetilde{W}\right)^{-1}\vec{Y}$ with the asymptotic rate*

$$R \geq -\ln\alpha - \ln\left(\beta\rho(W_s) + \gamma\rho(W'_s) + (1-\beta-\gamma)\rho(W_s)\rho(W'_s)\right)$$

Proof. Algorithm 1 equivalently performs the iterative Eq. (16). Suppose the sequence $\{\vec{F}^{(t)}\}$ converges to $\vec{F}^*$, then the error is $\epsilon^{(t)} = \vec{F}^{(t)} - \vec{F}^*$ and we have $\epsilon^{(t+1)} = \alpha\widetilde{W}\epsilon^{(t)} = \cdots = \left(\alpha\widetilde{W}\right)^t \epsilon^{(0)}$, $t = 0, 1, \ldots$ According to Lemma 3, $\rho(A_1) \leq 1$, $\rho(A_2) \leq 1$ and $\rho(A_3) \leq 1$ holds since $\rho(W_s) \leq 1$ and $\rho(W'_s) \leq 1$. In addition, Lemma 4 indicates that $\{A_i\}_{i=1}^3$ are simultaneously triangularizable (Horn and Johnson 1990). In other words, one can order the eigenvalues and choose the eigenbasis such that $\rho(A_i + A_j) \leq \rho(A_i) + \rho(A_j)$ for all pair i and j. Thus,

$$\begin{aligned}\rho(\alpha\widetilde{W}) \quad &= \quad \alpha\rho(\beta A_1 + \gamma A_2 + (1-\beta-\gamma)A_3) \\ &\leq \quad \alpha\left(\beta\rho(A_1) + \gamma\rho(A_2) + (1-\beta-\gamma)\rho(A_3)\right) \\ &\leq \quad \alpha(\beta + \gamma + (1-\beta-\gamma)) \leq \alpha < 1\end{aligned}$$

which guarantees the convergence of the algorithm. Furthermore, the asymptotic rate of convergence is

$$\begin{aligned}R &= -\ln\rho(\alpha\widetilde{W}) \\ &\geq -\ln\left(\alpha\left(\beta\rho(A_1) + \gamma\rho(A_2) + (1-\beta-\gamma)\rho(A_3)\right)\right) \\ &\geq -\ln\alpha - \ln\left(\beta\rho(W_s) + \gamma\rho(W'_s) + (1-\beta-\gamma)\rho(W_s)\rho(W'_s)\right)\end{aligned}$$

which completes the proof. □

Experiments

Data Sets and Experiment Settings

We use the following four benchmark multi-label data sets:
MSRC[4] data set contains 591 images annotated by 22 classes[5]. In our experiment, we use only image level annotation built upon the pixel level. We employ the same method as Boutell et al. to represent an image as a $8 \times 8 \times 3 \times 2 = 384$-dimensional vector.
Yahoo data set is described in (Ueda and Saito 2003), which is a multi-topic web page data sets complied from 11 top-level topics in the "yahoo.com" domain. Each web page is represented as a 37187-dimensional feature vector. We use the "scinece" topic because it has maximum number of labels, which contains 6345 web pages with 22 labels.

[3] The other two graph Laplacians can be used interchangeably in the formula and similar analysis can be made.

[4] http://research.microsoft.com/en-us/projects/objectclassrecognition/default.htm

[5] As suggested by the authors, the class "horse " are excluded since it has extremely few labeled instances.

Table 1: Classification performance of the six compared methods on the four multi-label data sets by 10-fold cross validation.

Data	Metrics		MLGF	MLCF	SMSE	MCGF	MLLS	MLTPG
MSRC	Macro	Precision	0.118	0.122	0.125	0.144	0.215	**0.241**
	average	F1 score	0.207	0.229	0.227	0.205	0.336	**0.368**
	Micro	Precision	0.195	0.207	0.214	0.200	0.319	**0.320**
	average	F1 score	0.117	0.128	0.125	0.127	0.205	**0.250**
Yahoo (Science)	Macro	Precision	0.238	0.247	0.229	0.250	0.333	**0.336**
	average	F1 score	0.329	0.353	0.272	0.357	0.404	**0.417**
	Micro	Precision	0.252	0.248	0.259	0.267	0.340	**0.346**
	average	F1 score	0.361	0.367	0.315	0.372	0.429	**0.441**
Music emotion	Macro	Precision	0.314	0.335	0.339	0.346	0.427	**0.499**
	average	F1 score	0.474	0.496	0.478	0.520	0.554	**0.591**
	Micro	Precision	0.327	0.342	0.349	0.366	0.429	**0.496**
	average	F1 score	0.479	0.502	0.489	0.527	0.568	**0.595**
Yeast	Macro	Precision	0.294	0.304	0.298	0.305	0.416	**0.416**
	average	F1 score	0.420	0.421	0.423	0.467	0.468	**0.476**
	Micro	Precision	0.296	0.304	0.315	0.376	0.417	**0.419**
	average	F1 score	0.457	0.466	0.460	0.480	0.505	**0.524**

Algorithm 1: Multi-label Learning on Tensor Product Graph (MLTPG)

```
Input:
Adjacency matrix of data graph $W$
Adjacency matrix of class graph $W'$
Three parameters $\alpha$, $\beta$ and $\gamma$
Number of maximal iteration $\max_{\text{iter}}$
The error parameter $\epsilon$
Output: Function prediction $\widetilde{Y}$
Calculate $W_s$ and $W'_s$;
$F^{(0)} = Y$;
for $t = 1; t \leq \max_{\text{iter}}$ do
    repeat
        $F^{(t+1)} = \alpha(\beta W_s F^{(t)} + \gamma W'_s F^{(t)} + (1 - \beta - \gamma) W_s F^{(t)} W'_s) + (1 - \alpha) Y$;
    until $\|F^{(t+1)} - F^{(t)}\|_F \leq \epsilon$;
end
Predict labels $\widetilde{Y}$ for unlabeled instances using $F$ by adaptive decision boundary method (Wang, Huang, and Ding 2009);
```

Music emotion data set (Trohidis et al. 2008) comprises 593 songs with 6 emotions. The dimensionality of the data points is 72.

Yeast data set (Elissee and Weston 2002) is formed by micro-array expression data and phylogenetic profiles with 2417 genes. Each gene is expressed as a 107-dimensional vector, which is associated with with 14 functions (labels).

Implementation Details

The proposed MLTPG approach requires both the adjacency matrices of data graph and class graph as input. We compute instance similarity using the Gaussian kernel function as $w(i, j) = \exp(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2)$ if $i \neq j$ and $w(i, j) = 0$ otherwise, where we empirically set $\sigma = \sum_{i \neq j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|/[(n(n-1)]$. In addition, we use co-occurrence based label similarity defined as (Wang, Huang, and Ding 2009)

$$w'(k, l) = \cos(\boldsymbol{y}^k, \boldsymbol{y}^l) = \frac{\langle \boldsymbol{y}^k, \boldsymbol{y}^l \rangle}{\|\boldsymbol{y}^k\| \|\boldsymbol{y}^l\|} \tag{18}$$

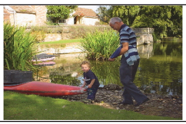
where $\boldsymbol{y}^k$ is the k-th column of matrix Y, thus $\langle \boldsymbol{y}^k, \boldsymbol{y}^l \rangle$ counts the common samples annotated with both the k-th and l-th classes. The parameter α indicates how much the relative amount of information from the TPG and the initial label information. Following (Zhou and Schölkopf 2004), we fix it to be 0.99 in all our experiments. Furthermore, two other parameters β and γ are respectively selected via the cross validation process. The finial predicted labels are derived from the score matrix F through the adaptive decision boundary method (Wang, Huang, and Ding 2009).

Classification Performance

We use standard 10-fold cross validation to evaluate the learning performance of our algorithm, and compare the experimental results with the following state-of-the-art graph-based multi-label learning approaches: (1) MLGF method, (2) MLCF method, (3) SMSE method, (4) Multi-label Correlated Green's Function (MCGF) (Wang, Huang, and Ding 2009) method. In addition, we also choose one dimensionality reduction based approaches: (5) Multi-label Least Square (MLLS) (Ji et al. 2008) method, which outperforms other methods belonging to the same type in previous studies. For the first three methods, we follow the detailed algorithms as described in the original work. For MCGF and MLLS, we use the codes posted by the authors. For the class graph, it is recomputed at each folder as it insists only on the training portion of the data.

In statistical learning, *precision* and F1 *score* are conventional metrics used to evaluate the classification performance. For each class, precision and F1 score are computed following the standard definition for a binary classification

Table 2: Prediction results of five images in MSRC data set by six compared approaches. Our method can predict all the labels for the images, while other methods can only recall part of the labels. The labels predicted by different algorithm but not in ground truth are in bold font, which, however, can be clearly seen in the images.

Image ID	6_2_s	7_30_s	12_34_s	19_25_s	3_1_s
					
MLGF	body, building,face,grass	car, grass, tree, sky	bird, tree	body, face	building, grass
MLCF	body, building, face, grass	body, car, grass, tree, sky	bird, **grass**, tree	body, **building**, face, grass	building, grass
SMSE	body, building, face, grass	body, car, grass, tree, sky	bird, **grass**	body, face, grass, **tree**	building, grass, road, sky
MCGF	body, building, grass face, road	body, car, grass, tree	bird, **grass** tree	body, **building**, face,	building, grass road
MLLS	body, building, grass face, road	body, car, grass tree, **road**, sky	bird, **grass**, tree	body, grass, face, **tree**	building,grass, tree, road, sky
MLTPG	body, building, grass face, tree, road,**sky**	body, car, grass tree, **road**, sky	bird, **grass**, tree, **road**	body, **building**, face grass, **sky**, **tree**	building,grass, tree, road, sky

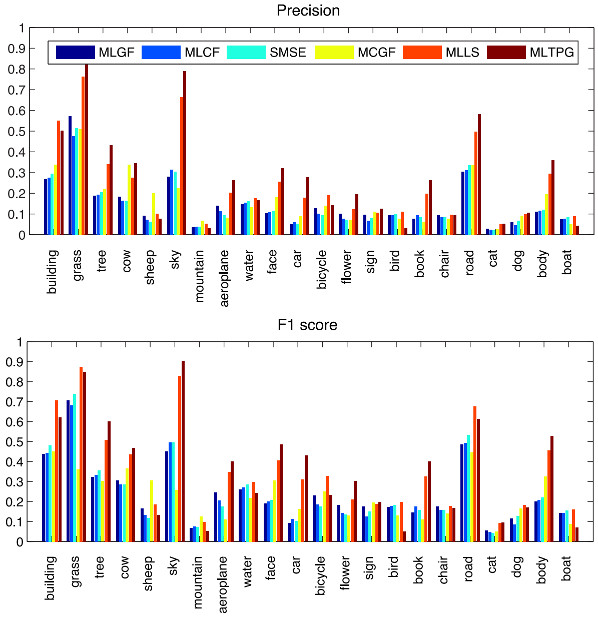


Figure 2: The class-wise precision and F1 score of six approaches on the MRSC data set.

problem. To address multi-label classification, macro average and micro average of precision and F1 score are used to assess the overall performance across multiple labels (Lewis et al. 2004).

The results in Table 1 show that our proposed algorithm MLTPG consistently, sometimes significantly, outperforms other five approaches. On average, our approach achieves more than 40%, 28%, 10% and 40% improvement compared to the best performance of the five methods on the four multi-label data sets, respectively. This indicates the effectiveness of our approach in multi-label learning problems, and provides a concrete evidence of the usefulness of the higher-order information.

We further check the precision and F1 score for each class of the four multi-label data sets. As expected, MLTPG is superior to other approaches over the vast majority of the classes, while degrades slightly on a few ones in each data set. Take the class-wise classification performance on MSRC data set for example. As illustrated in Fig.2, for precision, MLTPG outperforms other methods over 14 of all the 22 classes, especially the "background"categories that are closely correlated with many other "objective"classes, such as "grass", "sky", "water"and "road". For the "objective"classes, the proposed algorithm also works fairly well. Compared to the best performance of the other five approaches, some of the improvements are significant, such as the 60%, 56% and 34% improvement on "flower", "car"and "book", respectively. By contrast, it collapses when a few other "objective classes"(e.g., "sheep", "bird", etc.) are considered since both these classes have weak co-occurrence correlations with other categories. Similar phenomenon can be observed for F1 score. Table 2, where all the labels are correctly recovered by our algorithm while other approaches can only predict parts of the labels. More interestingly, the proposed method can associate some images with labels that are not in ground truth, but can be clearly seen in these images. One typical instance is the image 19_25_s.bmp annotated with only three labels "body", "face", and "grass"in ground truth. This image, nevertheless, contains another objectives, e.g., "building", "tree", etc. Although previous methods can identify some missing labels, only the proposed algorithm can simultaneously recover three labels "building", "sky"and "tree". This once again validates the benefits of taking higher-order relationships into account.

Conclusions

We show that the existing graph-based semi-supervised multi-label learning approaches can be uniformly formulated as a label propagation process or random walk with restart on a Cartesian Product Graph (CPG). Motivated by this discovery, we propose a new framework by employing the TPG of data graph and class graph for multi-label learning. Different from many existing approaches, both the intra-class and inter-class associations are explicitly represented as its weighted links and the proposed approach can be understood as a label propagation to spread the known labels on TPG. To avoid the high computational complexity and storage requirement, we never compute it directly on TPG and instead derive an iterative algorithm with the same computational complexity and the same amount of storage as the standard propagation on the data graph, which is guaranteed to converge. Applications to four benchmark multi-label data sets validation the effectiveness and efficiency of our algorithm as well as the usefulness of the higher-order association information.

References

Belkin, M.; Matveeva, I.; and Niyogi, P. 2004. Regularization and semi-supervised learning on large graphs. In *International Conference on Learning Theory*, 624–638.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition* 37(9):1757–1771.

Chen, G.; Song, Y.; Wang, F.; and Zhang, C. 2008. Semi-supervised multi-label learning by solving a sylvester equation. In *SIAM International Conference on Data Ming*, 410–419.

Chung, F. 1997. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, 2nd edition.

Elissee, A., and Weston, J. 2002. A kernel method for multi-labelled classifiation. In *Advances in Neural Information Processing Systems*, 681–687.

Harary, F. 1994. *Graph Theory*. Reading, MA: Addison-Wesley.

Horn, R. A., and Johnson, C. R. 1990. *Matrix Analysis*. Cambridge University Press, 2nd edition.

Ji, S.; Tang, L.; Yu, S.; and Ye, J. 2008. Extracting shared subspace for multi-label classification. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 381–389.

Kang, F.; Jin, R.; and Sukthankar, R. 2006. Correlated label propagation with application to multi-label learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1719–1726.

Knuth, D. E. 2008. *Introduction to Combinatorial Algorithms and Boolean Functions*. Reading, Massachusetts: Addison-Wesley.

Lafon, S., and Lee, A. B. 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Patt. Anal. Mach. Intell.* 28(9):1393–1403.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5:361–397.

Liu, Y.; Jin, R.; and Yang, L. 2006. Semi-supervised multi-label learning by constrained nonnegative matrix factorization. In *AAAI conference on Artificial Intelligence*, 421–426.

Magnus, J. R., and Neudecker, H. 1999. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, 2nd edition.

McCallum, A. K. 1999. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*.

Schapire, R. E., and Singer, Y. 2000. Boostexter: a boosting-based system for text categorization. *Machine Learning* 39(2/3):135–168.

Trohidis, K.; Tsoumakas, G.; Kalliris, G.; and Vlahavas, I. 2008. Multilabel classification of music into emotions. In *Proc. 2008 International Conference on Music Information Retrieval*, 325–330.

Ueda, N., and Saito, K. 2003. Parametric mixture models for multilabeled text. In *Advances in Neural Information Processing Systems*, volume 15, 721–728.

von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17(4):395–416.

Wang, F., and Zhang, C. 2006. Label propagation through linear neighborhood. In *International Conference on Machine Learning*, 985–992.

Wang, H.; Huang, H.; and Ding, C. 2009. Image annotation using multi-label correlated greens function. In *IEEE International Conference on Computer Vision*, 1–8.

Zha, Z.-J.; Mei, T.; Wang, J.; Wang, Z.; and Hua, X.-S. 2008. Graph-based semi-supervised learning with multiple labels. In *IEEE International Conference on Multimedia & Expo*, 1321–1324.

Zhang, Y., and Zhou, Z. 2008. Multi-label dimensionality reduction via dependence maximization. In *AAAI conference on Artificial Intelligence*, 1503–1505.

Zhou, D., and Schölkopf, B. 2004. A regularization framework for learning from graph data. In *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, 321–328.

Zhou, D.; Orshanskiy, S. A.; Zha, H.; and Giles, C. L. 2007. Co-ranking authors and documents in a heterogeneous network. In *IEEE International Conference on Data Mining*, 739–744.

Zhu, S.; Ji, X.; Xu, W.; and Gong, Y. 2005. Multi-labelled classification using maximum entropy method. In *ACM SIGIR Conference*, 274–281.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference Machine Learning*, 912–919.