Graph-based Label Propagation in Digital Media: A Review

OLGA ZOIDI, EFTYCHIA FOTIADOU, NIKOS NIKOLAIDIS and IOANNIS PITAS, Aristotle University of Thessaloniki

The expansion of the internet over the last decade and the proliferation of online social communities, such as Facebook, Google+ and Twitter and multimedia sharing sites such as, YouTube, Flickr and Picasa has led to a vast increase of available information to the user. In the case of multimedia data, such as images and videos, fast querying and processing of the available information requires the annotation of the multimedia data with semantic descriptors, i.e., labels. However, only a small proportion of the available data are labeled. The rest should undergo an annotation-labeling process. The necessity for the creation of automatic annotation algorithms gave birth to label propagation and semi-supervised learning. In this study, basic concepts in graph-based label propagation methods are discussed. More precisely, methods for proper graph construction based on the structure of the available data and label inference methods for spreading label information from a few labeled data to a larger set of unlabeled data are reviewed. Furthermore, applications of label propagation algorithms in digital media, as well as evaluation metrics for measuring their performance are presented.

Categories and Subject Descriptors: A.1 [**Introductory and survey**]; G.2.2 [**Discrete mathematics**]: Graph Theory; I.5.4 [**Pattern Recognition**]: Applications

General Terms: Algorithms, Documentation, Theory

Additional Key Words and Phrases: Semi-supervised learning, label propagation

ACM Reference Format:
Olga Zoidi, Eftychia Fotiadou, Nikos Nikolaidis and Ioannis Pitas. 2013. Graph-based Label Propagation in Digital Media: A Review *ACM Comput. Surv.* V, N, Article A (January YYYY), 35 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The introduction of the world wide web in the 1990s and the creation of portable digital devices in the 2000s allowed massive audiovisual user-fed content creation. More recently, social networking and social media, which connect users to each other and typically involve user-fed content, have been a big success. The current period is characterized by the direct interconnection of individuals and information sharing through

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 287674 (3DTVS). This publication reflects only the author's views. The European Union is not liable for any use that may be made of the information contained therein.
Author's addresses: O. Zoidi, E. Fotiadou, N. Nikolaidis and I. Pitas, Department of Informatics, Aristotle University of Thessaloniki, Box 451, Thessaloniki 54124, GREECE

DOI: http://dx.doi.org/10.1145/0000000.0000000

the Internet. Such information is typically accessible online. A great percentage of the available information concerns multimedia content, i.e., text, audio, images, videos and animations. One of the biggest multimedia sharing websites is YouTube, whose servers host video clips totaling hundreds of petabytes. The most common way for describing and searching the multimedia content is by applying semantic labels (tags), which capture its basic characteristics. This procedure is called annotation.

Annotations can serve the semantic multimedia data search. In websites such as YouTube, annotation is performed manually by the individual users: every time the user uploads a video to the YouTube database, he/she also associates it with tags, which can be used in a subsequent keyword-based search. However, manual annotation is not feasible, when the user is confronted with a large amount of unlabeled media content. This situation arises also in the case of television content annotation in television broadcaster archives. In this case, archivists perform a coarse annotation of the entire video, which is often insufficient for journalists to directly access video shots/frames of interest. Such problems can be overcome with semi-automatic annotation techniques, based on label propagation.

Label propagation is a semi-automatic annotation process for labeling a large amount of unlabeled data, when manual annotation is prohibitive. The objective of label propagation algorithms is to spread the labels from a small set of labeled data to a larger set of unlabeled data. Let us define the set of labeled data $\mathcal{X}_L = \{\mathbf{x_i}\}_{i=1}^{n_l}$, which are assigned labels from the set $\mathcal{L} = \{l_j\}_{j=1}^{L}$ and a set of unlabeled data $\mathcal{X}_U = \{\mathbf{x_i}\}_{i=1}^{n_u}$. Without loss of generality, we define the set of labeled and unlabeled data as $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_l}, \mathbf{x}_{n_l+1}, \ldots, \mathbf{x}_N\}$, $N = n_l + n_u$. The vector $\mathbf{y} = [y_1, \ldots, y_{n_l}, 0, \ldots, 0]^T = [\mathbf{y}_L \mathbf{y}_U]^T \in \mathcal{L}^N$ contains the labels of the labeled data in the first n_l positions and takes the value 0 in the last n_u positions. In matrix notation, the label matrix $\mathbf{Y} \in \Re^{N \times L}$ is defined as the matrix with entries $Y_{il} = 1$ if the i-th sample has the l-th label and $Y_{il} = 0$ otherwise. The objective of label propagation methods is to spread the labels in $\mathcal{L}$ from the set of labeled data $\mathcal{X}_L$ to the set of unlabeled data $\mathcal{X}_U$. Label propagation is essentially a diffusion process that exploits media content item (e.g. video segment) similarities. Similar diffusion approaches have been used in social networks for recommendation/preferences/concept propagation. Label propagation takes into consideration the following two requirements: 1) the labels of the initial labeled media items should remain unchanged and 2) media data that are "close" or "similar" to each other or lie in a compact domain of the feature space should be assigned the same label. The term "closeness" refers to the distance or similarity of the data projections to the utilized feature space. The most common way of describing the label propagation process is through a graph, where the data projections to the feature space represent the graph nodes and their pairwise distances (or equivalently, similarities), represent the weights of the graph edges. Then, label inference is performed along the graph paths that connect labeled nodes to unlabeled ones. The form of the graph, namely the weights on the graph edges, affects the performance of the label inference methods. Therefore, care should be taken, in order to construct a graph that captures efficiently the multimedia data structure. Moreover, label propagation performance is highly affected from the selection of the initial set of labeled data. Thus, in the beginning of the semi-automatic annotation procedure the small set of data that is manually assigned labels should be selected so that it maximizes the propagation accuracy.

Label propagation algorithms consist a subclass of the more general semi-supervised classifiers class. Semi-supervised learning refers to the exploitation of a great number of unlabeled data, in combination with a much smaller number of labeled data in the

construction of classifiers. Semi-supervised classifiers offer a better understanding of the data class distributions than supervised classifiers, which use only labeled data for training. Semi-supervised learning methods often lead to better classification accuracy than supervised ones. However, they cannot be used as a panacea. Limitations to the effectiveness of semi-supervised learning techniques are discussed in [Shai et al. 2008],[Singh et al. 2008]. Semi-supervised classifiers are divided into two categories, with respect to their application domain [Zhu 2008]: transductive classifiers, which learn a local representation of the data space and, therefore, can be employed only on the available labeled and unlabeled data and inductive classifiers, which learn a global representation of the data space and, therefore, can be also employed on ”unknown” data that do not belong to the originally available labeled and unlabeled datasets. Label propagation algorithms are transductive classifiers, i.e., they operate on a specific graph, whose nodes are the available labeled and unlabeled data. However, some recent works extend label propagation in the inductive setting, e.g., by assigning to incoming data the label of the closest labeled sample.

Label propagation algorithms encounter several challenges that affect their performance. An important one is the insufficiency of available labeled data, especially when label propagation is applied on real world datasets [Yan et al. 2003][Yan et al. 2006] . Another one is the proper choice of a distance function, which affects highly the structure of the constructed data graph [Maier et al. 2008]. Another challenge is the curse of dimensionality [Beyer et al. 1999], [Evangelista et al. 2006]. There are many situations in which data need to be represented by high-dimensionality low-level features. In such situations, the number of available labeled data that are required for good propagation performance increases prohibitively. Moreover, there are several methods that consider statistical models for the labeled data. If the number of labeled data is limited, the estimated statistical models are inaccurate for the entire dataset. The majority of datasets in computer vision consist of a few hundreds of labeled images/videos. The largest annotated multimedia data base that exists is ImageNet [Russakovsky et al. 2013], that consists of 1.2 million images for training, 50K images for validation and 100K images for testing object detection algorithms. Moreover, the largest dataset employed in semi-supervised learning is the extended MIT face database, where approximately 900K images where used for training and 24K for testing [Tsang and Kwok 2006]. Finally, label propagation methods in video data should take into account the temporal data consistency, ensuring that the semantic concept variation is small in local video temporal neighborhoods, i.e., within than across shots.

Label propagation has a wide range of applications. As mentioned above, it can be employed for multimedia content annotation [Phi et al. 2011]. Moreover, it can be exploited in the inference of semantic concepts from community-contributed images [Tang et al. 2009]. In medical imaging, it has been employed for anatomical brain MRI segmentation [Heckemann et al. 2006]. In biology, it finds application in disease gene discovery [Hwang and Kuang 2010], protein classification [Weston et al. 2005] and interaction studies [Letovsky and Kasif 2003]. In social networks, label propagation is employed for community detection [Gregory 2010]. In language analysis, label propagation has been used for document re-ranking [Yang et al. 2006], word sense disambiguation [Niu et al. 2005], noun phrase anaphoricity learning [Zhou and Kong 2009], word polarity induction [Rao and Ravichandran 2009] and classification [Speriosu et al. 2011]. In multimedia sharing sites, such as Youtube or Flickr, label propagation can be adopted in order to provide or to improve semantic annotations for the multimedia objects, or to recommend multimedia objects and groups to users. The feature descriptors used to represent the aforementioned objects depend on the ap-

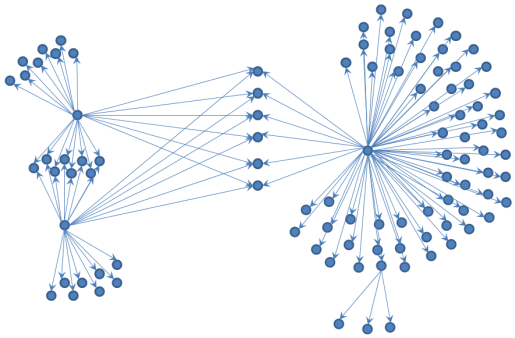


Fig. 1. Graph of a recommendation Network.

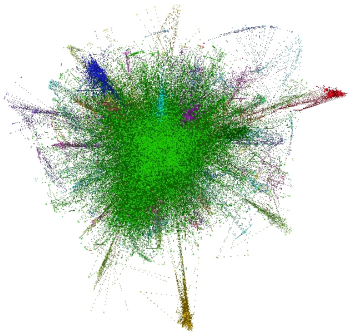


Fig. 2. 3D video content similarity graph.

plication and the multimedia type. Content-based descriptors can be visual or audio features, extracted from image/video or audio data respectively. For example, image annotation in image-sharing sites such as Flickr utilizes visual features in order to represent the semantic concepts appearing in the images [Houle et al. 2013], [Tang et al. 2011]. Furthermore, other descriptor types can be utilized, which capture the user activity within the community. As an example, video recommendation in Youtube can be based on users co-view information [Baluja et al. 2008]. In the current survey, we focus on the study of graph-based label propagation algorithms and their application in digital media. Our purpose is to summarize the basic notions and principles, as well as the work done in this area so far.

2. GRAPH CONSTRUCTION

2.1. Basic graph theory concepts

Graphs provide a natural representation of entities and relationships between them. They are used in various research fields to mathematically represent a wide range of networks describing data relationships [Wasserman and Faust 1994]. Given a set of N entities and their pairwise relationships, a graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ is constructed, where the set of nodes $\mathcal{V}$ represents the entities and the set of edges $\mathcal{E}$ represents their relationships. Figure 1 depicts the graph of a recommendation network. Such a network represents how a recommendation, which may refer to a product or an idea, is propagated (through arrows) among the individuals (nodes) in the network. Figure 2 shows a graph, whose nodes are 3D videos in YouTube. Its edges represent content similarity. In label propagation on such data, we want to propagate content labels from labeled 3D video nodes to unlabeled ones along these edges. Such labels can be, e.g., 'shallow depth', 'uncomfortable for viewing', 'good 3D quality', etc.

There are two types of graphs, based on the type of the pairwise relationships: *binary* and *weighted* ones. In binary graphs, the adjacency matrix $\mathbf{A} \in \Re^{N \times N}$ is defined as:

$$A_{ij} = \begin{cases} 1, & \text{if nodes i and j are connected to each other} \\ 0, & \text{if nodes i and j are not connected to each other.} \end{cases} \quad (1)$$

In weighted graphs, the (i, j)- entry of the weight matrix $\mathbf{W} \in \Re^{N \times N}$ is the similarity W_{ij} between the i-th and j-th node. When the relationships depend on the node relation direction, then the graph is called *directed* and the edges are represented by arrows. Otherwise, the graph is *undirected*. In undirected graphs, the adjacency and

weight matrices are symmetric, i.e. $\mathbf{A} = \mathbf{A}^T$, $\mathbf{W} = \mathbf{W}^T$. If all graph nodes have the same number of neighbors, i.e., they have the same number of incident edges, then the graph is a *regular* one. In the special case where every graph node is connected to all other graph nodes, the graph is a *complete* one. The *degree* is the number of the incident edges at a node. It is associated to the "importance" of a node in the graph. The higher the degree is, the more a node is important in a graph, as many edges converge to it. Nodes without connections are called *isolates*, and their degree equals 0. In a directed graph, the in-degree represents the number of edges ending at a node, while the out-degree represents the number of edges starting from a node. A sequence of consecutive edges in a graph defines a *path*. For a path to exist between two nodes, it must be possible to travel from one node to the other one, through an uninterrupted sequence of edges. In binary graphs, the *path length* is the number of the traversed edges. In weighted graphs, the length of the path is the sum of the inverse edge weights that belong to the path [Newman 2001]. A more general definition of the path length is given in [Opsahl et al. 2010], as being the sum of the inverse graph weights raised to a power of a, where a is a regulation term that determines the significance of the graph weights, versus the number of the edges in the path.

Given a graph with weight matrix $\mathbf{W}$, the degree matrix $\mathbf{D} \in \Re^{N \times N}$ is defined as the diagonal matrix, whose i-th entry is given by $D_{ii} = \sum_j W_{ij}$. Moreover, the combinatorial graph Laplacian matrix $\mathbf{L} \in \Re^{N \times N}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The normalized Laplacian $\tilde{\mathbf{L}} \in \Re^{N \times N}$ is defined as $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Matrices $\mathbf{D}$, $\mathbf{L}$ and $\tilde{\mathbf{L}}$ are widely used in label inference methods, as will be shown in Section 3.

2.2. Graph construction methods

The first step in label propagation deals with the construction of a suitable graph for proper multimedia data representation. Essentially, label propagation is a classification task. Therefore, a proper graph should capture the data characteristics that are most discriminative for the label propagation task. This means that the constructed graph should maximize the label propagation accuracy, i.e., the label spread should be performed among data that belong to the same class. Graph construction methods can be divided into three categories: neighborhood methods, local reconstruction methods and metric learning methods. Neighborhood methods result in sparse graphs, however they are sensitive to the selection of the node neighborhood which greatly influences the propagation accuracy. Local reconstruction methods construct more robust sparse graphs, however they are not suitable for large datasets, since they have high computational complexity. On the other hand, metric learning methods have the advantage of incorporating prior knowledge for the data. The selection of the graph construction method depends on the label propagation application. A study on the significance of graph construction can be found in [Maier et al. 2008].

2.2.1. Neighborhood methods. In neighborhood methods, each graph node described by a feature vector is connected only to its closest neighbors. There are two types of neighborhood graphs: k-nearest neighbor graphs (k-NNG) and e-neighborhood graphs [Talukdar 2009]. In k-NNG, each node is connected to its k nearest nodes, where node proximity is measured by a distance function, such as the Euclidean distance on the feature space. For a graph with N nodes, if we set $k = N - 1$, then the graph is complete, i.e., each node is connected with every other graph node. By setting a small value for k, the resulting graph is sparse. If a node i belongs to the k nearest neighbors of node j, this does not mean that node j belongs to the k-nearest neighbors of node

i. Therefore, k-NNG results in irregular graphs, where the nodes have different degrees. In e-neighborhood graphs, each node i is connected to all nodes, which lie within a ball around node i with radius e. The disadvantages of e-neighborhood graphs are that they are sensitive to the value of e and that they commonly create graph structures with disconnected components. Traditionally, both neighborhood creation methods are not scalable, i.e., the growth of nodes increases the computational complexity of the graph construction super-linearly. This drawback is alleviated in [Satuluri and Parthasarathy 2009] and [Wang et al. 2012], by introducing scalable methods for searching for the k-nearest neighbors in very large graphs.

k-NNG and e-neighborhood graph construction methods result in irregular graphs. The construction of regular graphs is guaranteed in the b-matching method proposed by Jebara et al. in [Jebara et al. 2009]. The b-matching method consists of two steps: a) graph sparsification, which deals with the selection of the graph edges that will be present in the final graph and b) edge re-weighting. b-matching operates on the distance matrix $\mathbf{D} \in \Re^{N \times N}$, which is derived from the weight matrix $\mathbf{W}$ by $D_{ij} = \sqrt{W_{ii} + W_{jj} - 2W_{ij}}$. In the sparsification step, the algorithm searches for the binary matrix $\mathbf{P} \in \{0,1\}^{N \times N}$ that minimizes the objective function:

$$\min_{\mathbf{P}} \sum_{ij} P_{ij} D_{ij}, \tag{2}$$

subject to the constraints (in short s.t.) $\sum_j P_{ij} = b, P_{ii} = 0, P_{ij} = P_{ji}, \forall i, j = 1, \ldots, N$. Edge re-weighting can be performed in three ways: a) by using binary weights ($\mathbf{W} = \mathbf{P}$), b) by applying Gaussian kernel weighting $W_{ij} = P_{ij} \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right)$, where $d(\mathbf{x}_i, \mathbf{x}_j)$ is some distance function between the node vectors $\mathbf{x}_i$, $\mathbf{x}_j$ describing the node multimedia data and σ is the kernel bandwidth or, c) motivated by the Locally Linear Embeding (LLE) algorithm [Roweis and Saul 2000], by finding the coefficients that minimize the reconstruction error:

$$\min_{\mathbf{W}} \sum_i \|\mathbf{x}_i - \sum_j P_{ij} W_{ij} \mathbf{x}_j\|^2 \ \text{ s.t. } \sum_j W_{ij} = 1, W_{ij} \geq 0. \tag{3}$$

2.2.2. Local reconstruction. In most label propagation methods, data are organized in complete graphs, where similarities between nodes are calculated through a Gaussian function, such as the Radial Basis Function (RBF):

$$W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right), \tag{4}$$

where σ is the variance of the Gaussian function. The greatest disadvantage of this graph construction is that the resulting weight matrix $\mathbf{W}$ and, subsequently, the label propagation results depend highly on the value of σ. This problem is addressed in local graph reconstruction methods. Such methods aim at the reconstruction of each graph node from other nodes in the graph, usually as their linear combination. In Linear Neighborhood Propagation (LNP) [Wang and Zhang 2006], the k nearest neighbors of each graph node are selected, based on the RBF kernel similarity. The objective is the minimization of the reconstruction error of each node from its k nearest neighbors, subject to the constraints that the coefficients W_{ij} of each linear combination are positive and sum to one:

$$\min_{W_{ij}} \left\| \mathbf{x}_i - \sum_{j:\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} W_{ij} \mathbf{x}_j \right\|^2 \quad \text{s.t. } \sum_j W_{ij} = 1, \ \ W_{ij} \geq 0, \tag{5}$$

where $\mathcal{N}(x_i)$ is the neighborhood of node i. A sparse graph is then reconstructed, where each node is connected only to its k nearest neighbors, with weights equal to the respective coefficients of the linear combination. An extension of LNP to the non-linear space through kernels was introduced in [Tang et al. 2008]. In the resulting Kernel Linear Neighborhood Propagation (KLNP) method, the feature data are mapped to a higher dimensional space by a kernel mapping $\phi : \mathbf{X} \rightarrow \mathbf{\Phi}$. The new objective function then takes the form:

$$\min_{W_{ij}} \left\| \phi(\mathbf{x}_i) - \sum_{j:\phi(\mathbf{x}_j)\in\mathcal{N}(\phi(\mathbf{x}_i))} W_{ij}\phi(\mathbf{x}_j) \right\|^2 \quad \text{s.t. } \sum_j W_{ij} = 1, \quad W_{ij} \geq 0, \tag{6}$$

Another variant of LNP is the Correlative Linear Neighborhood Propagation (CLNP) introduced in [Tang et al. 2009]. In CLNP, the objective is to exploit prior information about the semantic correlations between the data labels in graph construction. In [Cheng et al. 2009], Cheng et al. proposed a method for graph reconstruction using a Sparsity Induced Similarity (SIS) measure. SIS is based on the intuitive notion that the sparse decomposition of a node indicates its true neighborhood structure. Therefore, for each graph node, the sparsest decomposition in other graph nodes is searched.

Similarly to [Wang and Zhang 2006], Daitch et al. proposed a hard and an a-soft graph reconstruction method in [Daitch et al. 2009], which aim at minimizing the objective function:

$$\min_{\mathbf{W}} \sum_i \|d_i\mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j\|^2, \tag{7}$$

where $d_i = \sum_j W_{ij}$ is the weighted degree of node i. One can notice that, in contrast to LNP (5), the nodes j are not required to belong to the neighborhood of node i. In hard graph construction, each node is constrained to have a weighted degree of at least 1 ($d_i \geq 1$). In a-soft graph construction, this constraint is relaxed to $\sum_i(\max(0, 1-d_i))^2 \leq aN$, where a is a hyper-parameter. The a-soft constraint implies that, in the case of outliers, the weighted degrees of the respective nodes are allowed to take a lower value.

Motivated by the study of Rao et al. [Rao 2002], Tang et al. proposed a kNN-Sparse graph construction method [Tang et al. 2011], which removes most of the semantically-unrelated edges between nodes by performing a sparse reconstruction of each node from its k nearest neighbors. The objective is the minimization of:

$$\min_{\mathbf{w}_i} \|\mathbf{w}_i\|_1, \quad \text{s.t. } \mathbf{x}_i = \mathbf{B}_i\mathbf{w}_i, \tag{8}$$

where $\mathbf{B}_i$ is the matrix of the samples that belong to the k nearest neighbors of $\mathbf{x}_i$ and $\mathbf{w}_i$ is the vector of the reconstruction coefficients. By selecting $k = N - 1$, then the one-vs-all sparse graph reconstruction method is obtained [Tang et al. 2009]. The graph weights are selected according to the following rule:

$$w_{ij} = \begin{cases} \mathbf{w}_i(p), & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \text{ and } j = i_p \\ 0, & \text{if } \mathbf{x}_j \notin \mathcal{N}(\mathbf{x}_i), \end{cases} \tag{9}$$

where i_p denotes the p-th neighbor of $\mathbf{x}_i$.

A novel method that extends the LNP algorithm in hypergraphs is introduced in [Wang et al. 2009]. In hypergraphs, hyperedges connect any number of nodes instead of exactly two, as common graph edges do, i.e., they represent multiple node relationships. Given the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$, the hypergraph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ is constructed, where $\mathcal{V}$ is the set of nodes of graph $\mathcal{G}$ and $\mathcal{E}'$ is the set of hyperedges. The hyperedges $e' \in \mathcal{E}'$ are subsets of $\mathcal{E}$. In [Wang et al. 2009], the hyperedge e'_i is defined as the set of all the adjacent edges to node i. If the graph $\mathcal{G}$ is cast into a first-order Intrinsic Gaussian Markov Random Field (IGMRF) framework [Rue and Held 2005],

then the hypergraph $\mathcal{G}'$ can be cast into a second-order IGMRF framework, where the increment for hyperedge e'_i is defined as:

$$d_i = y_i - \sum_{j \in \mathcal{N}_i} W_{ij} y_j, \quad \sum_{j \in \mathcal{N}_i} W_{ij} = 1, \tag{10}$$

where $\mathcal{N}_i$ is the set of neighboring nodes to node i and y_i the label of node i. The hyperedges weights W_{ij} are computed by solving the quadratic problem (5).

2.2.3. Metric Learning. In both supervised and semi-supervised learning algorithms, the classification performance depends highly on the metric used for computing the distances between multimedia data. In graph construction, data distance is exploited, in order to calculate a similarity measure (edge weight) between multimedia data (graph nodes). In the absence of prior information, most algorithms employ the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ for computing the similarities between two nodes. However, Euclidean distance contains no information for an underlying structure (e.g., cluster or manifold) that may exist in the training data. If labeled data are available, they can be exploited in the construction of a distance metric that sets small distance values between data with the same label and large values between data with different labels. Generally, metric learning algorithms estimate a Mahalanobis distance $d_A(\mathbf{x}_i, \mathbf{x}_j)$ between data [Xing et al. 2002]:

$$d_A(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)}, \tag{11}$$

by imposing constraints derived from the labeled data (supervised metric learning). In the above equation (11), $\mathbf{A} \in \Re^{N \times N}$ is a positive semi-definite matrix $\mathbf{A} \succeq 0$. Mahalanobis distance is a generalization of the Euclidean distance, which performs linear scaling and rotation in each dimension of the feature space. A proper choice of matrix $\mathbf{A}$ should increase the significance of informative dimensions and ignore the non-informative ones. By setting $\mathbf{A} = \mathbf{I}_N$, where $\mathbf{I}_N \in \Re^{N \times N}$ is the identity matrix, we obtain the Euclidean distance. The constraints can be divided into two categories. Those that are applied on the set of similar data pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ (e.g., the distance between similar data pairs should not be greater that a predetermined threshold) and those that are applied on the set of dissimilar data pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}$ (e.g., the distance between dissimilar data pairs should be over a threshold value). Finally, the graph weights W_{ij} are calculated according to a rule of the form:

$$W_{ij} \propto \exp\{-d_A(\mathbf{x}_i, \mathbf{x}_j)\}. \tag{12}$$

Metric learning algorithms try to minimize a cost function of $f(\mathbf{A})$ subject to the constraints $g(\mathbf{A})$:

$$\min_{\mathbf{A} \succeq 0} f(\mathbf{A}), \text{ s.t.} \quad g(\mathbf{A}) \tag{13}$$

In the method proposed by Xing et al. in [Xing et al. 2002], the objective is to minimize the squared sum of distances between similar data pairs, under the constraint that the sum of distances between dissimilar data pairs does not drop under a threshold. Equivalently, the objective is the maximization of the sum of distances between dissimilar data pairs, under the constraint that the squared sum of distances of similar data pairs does not exceed some threshold. In the Information-Theoretic Metric Learning (ITML) algorithm [Davis et al. 2007], prior knowledge about the inter-object distances and the matrix $\mathbf{A}$, denoted by $\mathbf{A}_0$, is considered. The objective function $f(\mathbf{A})$ targets the minimization of the Log-Determinant divergence of matrices $\mathbf{A}$, $\mathbf{A}_0$:

$$D_{ld}(\mathbf{A}, \mathbf{A}_0) = \mathbf{tr}(\mathbf{A}\mathbf{A}_0^{-1}) - \log\det(\mathbf{A}\mathbf{A}_0^{-1}) - N, \tag{14}$$

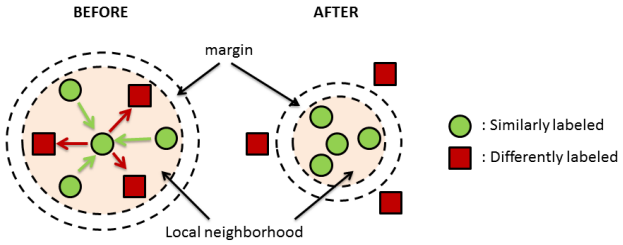


Fig. 3. Schematic illustration of LMNN.

subject to the constraints that the sum of distances between similar data pairs is under a threshold u and the sum of distances between dissimilar data pairs is over a threshold l.

In [Goldberger et al. 2004] and [Weinberger and Saul 2009], the objective is to transform the feature space, so that application of the metric distance satisfies the k-NN classification objective, i.e., the k nearest neighbors have the same label. In Neighborhood Component Analysis (NCA) [Goldberger et al. 2004], matrix $\mathbf{A}$ in (11) is written in the form $\mathbf{A} = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L}$ is the transformation matrix. Then, the objective is the maximization of the expected number of correctly classified nodes:

$$f(\mathbf{L}) = \sum_i \sum_{j \in \mathcal{C}_i} \frac{\exp\left(-\|\mathbf{L}\mathbf{x}_i - \mathbf{L}\mathbf{x}_j\|\|_2^2\right)}{\sum_{k \neq i} \exp\left(-\|\mathbf{L}\mathbf{x}_i - \mathbf{L}\mathbf{x}_k\|_2^2\right)}, \tag{15}$$

where $\mathcal{C}_i$ is the set of nodes with the same label as node $\mathbf{x}_i$. In Large Margin Nearest Neighbor (LMNN) algorithm [Weinberger and Saul 2009], for each labeled node, k nearest neighbors (target neighbors) are determined based on the Euclidean distance. The algorithm then searches for a transformation of the feature space that pulls the target nodes closer to the labeled node and pushes nearby nodes with different labels (impostors) away by a margin, which is determined from the distance of the impostor to the nearest target neighbor. A schematic illustration of LMNN algorithm is depicted in Figure 3. Similarly to [Goldberger et al. 2004] and [Weinberger and Saul 2009], a Support Vector Machine approach to metric learning called MLSVM, was proposed in [Nguyen and Guo 2008], that incorporates the kernel trick on the data $\mathcal{X}$.

In the method proposed in [Bilenko et al. 2004], a novel distance metric learning method is presented, that assumes a different metric for each cluster, thus allowing different shapes for different clusters. For each cluster c, a weight matrix $\mathbf{A}_c$ is searched that minimizes the total squared Euclidean distance between the data of the cluster and the cluster centroids μ_c and maximizes the complete data log-likelihood, under must-link and cannot-link constraints:

$$\arg\min_{\mathbf{A}_c} \sum_{\mathbf{x}_i \in \mathcal{X}} \left(\|\mathbf{x}_i - \mu_c\|^2_{\mathbf{A}_c} - \log(\det(\mathbf{A}_c))\right) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} w_{ij}\mathbf{1}[l_i \neq l_j] + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \bar{w}_{ij}\mathbf{1}[l_i = l_j], \tag{16}$$

where

$$\|\mathbf{x}_i - \mu_c\|_{\mathbf{A}_c} = \sqrt{(\mathbf{x}_i - \mu_c)^T\mathbf{A}_c(\mathbf{x}_i - \mu_c)} \tag{17}$$

and $\mathbf{1}$ is the indicator function $\mathbf{1}[true] = 1$ and $\mathbf{1}[false] = 0$. The above mentioned metric learning algorithms are supervised in nature, as they exploit a priori information provided by the labeled data. Dhillon et al. [Dhillon et al. 2010] introduced a semi-supervised metric learning algorithm, which exploits information obtained from both labeled and unlabeled data, called Inference Driven Metric Learning (IDML). The idea behind IDML is to combine supervised metric learning algorithms with transductive graph-based label propagation algorithms in a unified framework. Metric learning and

label propagation are performed sequentially in an iterative manner, by applying label propagation on the graph constructed by metric learning. At the end of each iteration, the set of labeled data is enriched with the data that were assigned a label with a high confidence. In the next iteration, a new graph is constructed based on the enriched labeled dataset. The procedure continues, until no more data can be added to the labeled dataset. In Laplacian regularized metric learning (LRML) [Hoi et al. 2008], knowledge of unlabeled data information is exploited through the graph Laplacian. The intuitive notion behind LRML is to find the metric that minimizes the distance between the sample $\mathbf{x}_i$ and its k nearest neighbors $\mathcal{N}(\mathbf{x}_i)$.

When no labeled data are available, unsupervised metric learning is related to manifold learning and, subsequently, dimensionality reduction [Yang 2006]. Dimensionality reduction methods search for the low-dimensional manifold that preserves the geometric relationships, i.e., the distance between the data. The most commonly used unsupervised dimensionality reduction methods are Principal Component Analysis (PCA) [Jolliffe 2002], Multidimensional Scaling (MDS) [Borg and Groenen 2005], Locally Linear Embedding (LLE) [Saul and Roweis 2003], the Laplacian Eigenmap [Belkin and Niyogi 2003] and ISOMAP [Tenenbaum et al. 2000]. A comprehensive study on the relationship between unsupervised distance metric learning and dimensionality reduction can be found in [Yang 2006].

3. LABEL INFERENCE

After graph construction, label propagation is performed on the data (graph nodes) according to some label inference method. Label inference refers to the way the labels $\mathcal{L}$ are spread from the set of labeled data $\mathcal{X}_L$ to the set of unlabeled data $\mathcal{X}_U$. Label inference methods can be divided into several categories, based on the rules that govern the label spread and the type and number of graphs they apply to.

3.1. Iterative algorithms

In iterative label propagation algorithms, label spread is performed gradually on the unlabeled data, according to some update rule, which converges to a stationary state as $t \to \infty$. The stationary state of each iterative algorithm can be computed beforehand. Therefore, in practice, these methods are performed in a single step. In one of the earlier methods on label propagation proposed by Zhu et al. [Zhu and Ghahramani 2002], the labels of the labeled and unlabeled nodes are updated according to:

$$\mathbf{F}^{(t+1)} = \mathbf{D}^{-1}\mathbf{W}\mathbf{F}^{(t)}, \tag{18}$$

where $\mathbf{F} \in \Re^{N \times L}$ is a classification function that assigns labels to the labeled and unlabeled data and $\mathbf{D} = \text{diag}\{D_{ii} = \sum_j W_{ij}\}$ is the degree matrix. The term $\mathbf{D}^{-1}\mathbf{W}$ corresponds to the probability of each node to take the label of its neighbors. The iterative algorithm (18) is also employed in [Phi et al. 2011] for label propagation in facial images (anchor persons), where additional information was exploited concerning the anchor person appearances in the videos. A similar iterative algorithm, inspired by the Jacobi iterative method for linear systems [Barrett et al. 1994] is introduced in [Bengio et al. 2006], where the update rules of the labeled and unlabeled nodes are given by:

$$y_i^{(t+1)} = \frac{\sum_j W_{ij} y_j^{(t)} + \frac{1}{\mu} y_i^{(t)}}{\sum_j W_{ij} + \frac{1}{\mu} + \epsilon}, \quad y_i^{(t+1)} = \frac{\sum_j W_{ij} y_j^{(t)}}{\sum_j W_{ij} + \epsilon}, \tag{19}$$

respectively, where μ is a parameter that regulates the weight of the labeled node.

Zhou et al. [Zhou et al. 2004] proposed an iterative process:

$$\mathbf{F}^{(t+1)} = \mu(\mathbf{I} - \tilde{\mathbf{L}})\mathbf{F}^{(t)} + (1-\mu)\mathbf{Y}, \text{ where } \mathbf{F}^0 = \mathbf{Y}, \tag{20}$$

of gradual spread of label information through the graph, where $\tilde{\mathbf{L}}$ is the normalized graph Laplacian, which will be discussed in subsection 3.4. In each iteration, the graph nodes receive information from their neighbors, while maintaining the information of their initial state (label). The function $\mathbf{F}$ assigns to each node, either labeled or unlabeled, L values, one for each candidate label. The new node label is the one that corresponds to the highest $\mathbf{F}$ value. This method was incorporated in the graph-based active learning method introduced in [Long et al. 2008]. Similarly to [Zhou et al. 2004], Wang et al. [Wang and Zhang 2006] employed the update rule:

$$\mathbf{F}^{(t+1)} = \alpha\mathbf{W}\mathbf{F}^{(t)} + (1-\alpha)\mathbf{Y}, \text{ where } \mathbf{F}^0 = \mathbf{Y}. \tag{21}$$

3.2. Random walks

In label propagation methods based on random walks, the classification decision is taken by comparing the expected random steps required to reach the unlabeled nodes, starting from the labeled nodes of different classes. In the method proposed by Szummer et al. [Szummer and Jaakkola 2002], the transition probabilities p_{ij} from node i to node j are given by:

$$p_{ij} = \frac{W_{ij}}{\sum_k W_{ik}}. \tag{22}$$

Then, node k takes the label y_{start}, if the probability that we arrive at node k after t steps starting from some node with label y_{start} is greater than 0.5. This probability is given by:

$$P^t(y_{start}|k) = \sum_{i=1}^{N} P(y = y_{start}|i)P_{0|t}(i|k), \tag{23}$$

where $P(y|i)$ is the probability that node i has the label y and $P_{0|t}(i|k)$ is the probability of reaching node k starting from node i in t steps. The probabilities $P(y|i)$ are estimated using an iterative EM algorithm or by maximizing a criterion that leads to a closed-form solution. In the two-class classification problem in [Zhou and Schölkopf 2004], the transition probability matrix $\mathbf{P}$ is defined as:

$$\mathbf{P} = (1-a)\mathbf{I} + a\mathbf{D}^{-1}\mathbf{W}, \tag{24}$$

where $a \in (0,1)$. The classification decision is taken by comparing the commute times $\bar{G}_{ij}$ to the labeled nodes of different classes with labels $\mathcal{L} = \{+1, -1\}$:

$$p_+(\mathbf{x}_i) = \sum_{j|y_j=1} \bar{G}_{ij},\ p_-(\mathbf{x}_i) = \sum_{j|y_j=-1} \bar{G}_{ij}, \tag{25}$$

where $\bar{G} = (\mathbf{I} - a\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})^{-1}$. Commute time $\bar{G}_{ij}$ is the expected number of steps required to reach node j with a random walk starting from node i, and get back to i. The Adsorption algorithm proposed in [Baluja et al. 2008] can be interpreted as a random walk process on a graph. Each node i is associated with a probability distribution function N_i defined over its neighbors, calculated as in (22).Then, the label distribution L_i of a node i can be considered as a convex combination of its neighbors' label distributions $L_i = \sum_j N_i(j)L_j$.

3.3. Graph regularization

Generally, label propagation on graphs defines a classification function $\mathbf{f} \in \Re^N$ on both labeled and unlabeled data that spreads the labels from the labeled to the unlabeled graph nodes. The classification function $\mathbf{f}$ should try to maintain the original labels on the labeled nodes as much as possible while applying the same label on unlabeled nodes that lie close to each other or belong to the same structure (e.g., cluster or manifold). The second assumption implies that $\mathbf{f}$ should be smooth over the entire graph. This results in a regularization framework of the form:

$$\min_{\mathbf{f}}\{\alpha\mathcal{C}(\mathbf{f}_L) + \beta\mathcal{S}(\mathbf{f})\}, \tag{26}$$

where $\mathcal{C}(\mathbf{f}_L)$ is a cost function on the labeled nodes that penalizes the divergence of the output labels from the initial labels and $\mathcal{S}(\mathbf{f})$ is a smoothness constraint on the whole graph. α and β are regularization parameters, which capture the trade-off between the two terms. Usually, the smoothness constraint is of the form:

$$\mathcal{S}(\mathbf{f}) = \mathbf{f}^T\mathbf{S}\mathbf{f}, \tag{27}$$

where $\mathbf{S}$ is a smoothing matrix. The algorithms of this category differ in the choice of the cost function and smoothness constraint, as well as in the incorporation of additional constraints. In one of the earlier works, Zhou et al. [Zhou et al. 2004] proposed a label propagation method which ensures local and global consistency. The algorithm minimizes the quadratic cost function on the labeled data:

$$\mathcal{C}(\mathbf{f}_L) = (\mathbf{f}_L - \mathbf{y}_L)^T(\mathbf{f}_L - \mathbf{y}_L), \tag{28}$$

under the smoothness constraint:

$$\mathcal{S}(\mathbf{f}) = \mathbf{f}^T\tilde{\mathbf{L}}\mathbf{f}, \tag{29}$$

where $\tilde{\mathbf{L}}$ is the normalized Laplacian. It is proven that the iterative process (20) converges to the global minima of the regularization framework defined by (26)-(29) [Zhou et al. 2004]. The equivalence between the regularization framework and the iterative process is applied in numerous other graph-based propagation algorithms [Wang and Zhang 2006].

In [Wang and Zhang 2006], [Tang et al. 2011], the cost function used is the one in (26) and the smoothness matrix is $\mathbf{S} = \mathbf{I} - \mathbf{W}$, where $\mathbf{W}$ is the weight matrix. In [Belkin et al. 2004], two regularization methods are introduced, namely Tikhonov regularization:

$$\min_{\mathbf{f}}\left\{\frac{1}{k}(\mathbf{f}_L - \mathbf{y}_L)^T(\mathbf{f}_L - \mathbf{y}_L) + \gamma\mathbf{f}^T\mathbf{S}\mathbf{f}\right\},\ \gamma \in \Re \tag{30}$$

and interpolated regularization:

$$\min_{\mathbf{f}}\left\{\mathbf{f}^T\mathbf{S}\mathbf{f}\right\}, \tag{31}$$

where k is the number of nodes, which belong to the regularization manifold and $\mathbf{S} = \mathbf{L}$ or $\mathbf{S} = \mathbf{L}^p$, $p \in \mathbf{N}$, under the stabilization constraint $\sum \mathbf{f}_i = 0$.

The measure propagation method in [Subramanya and Bilmes 2011] introduces an objective function which relies on the minimization of the Kullback-Leibler divergence between probability measures that encode label membership probabilities. Specifically, for each node i of the graph and for each node j of the labeled set of nodes, two probability measures are defined: $p_i(l), i = 1, .., N$ expresses the (predicted) probability that node i belongs to the label (class) l, $l \in \mathcal{L}$, while $r_j(l), j = 1, ..., n_l$ is the known probability distribution of the labeled nodes. The objective function takes the form:

$$\min_{\mathbf{P}} \sum_{i=1}^{n_l} D_{KL}(r_i||p_i) + \mu \sum_{i=1}^{N} \sum_{j\in\mathcal{N}(i)} w_{ij} D_{KL}(p_i||p_j) - \nu \sum_{i=1}^{N} H(p_i), \tag{32}$$

where D_{KL} denotes the Kullback-Leibler divergence between p_i and q_j, calculated as $D_{KL}(p||q) = \sum_y p(y) \log \frac{p(y)}{q(y)}$, while $H(p) = -\sum_y p(y) \log p(y)$ is the Shannon entropy of p. The two first terms correspond to the constraints expressed in (26) and the third term enforces probability distributions p_i to be close to the uniform distribution.

The method in [Zhu et al. 2003] formulates the regularization problem by defining a Gaussian Random Field on the graph and minimizing the quadratic energy function $\mathbf{f}^T\mathbf{L}\mathbf{f}$, while retaining the initial labels of the labeled nodes, by setting the parameter a in (26) to ∞. The minimum energy function satisfies the harmonic property, i.e., it is equivalent to the average energy of the neighboring nodes. Zhu et al. studied the relationship between Gaussian random fields and Gaussian processes in [Zhu et al. 2003], using a spectrum transformation on the graph Laplacian matrix, which will be described in subsection 3.4.

The graph mincuts method [Blum and Chawla 2001] targets the problem of binary label propagation with labels $\mathcal{L} = \{-1, 1\}$ as a clustering problem, which finds the minimum set of edges whose removal isolate the nodes with label 1 from those with label -1. The regularization framework of graph mincut takes the form:

$$\min_{\mathbf{f}} \{a(\mathbf{f}_L - \mathbf{y}_L)^T(\mathbf{f}_L - \mathbf{y}_L) + \frac{1}{2}\mathbf{f}^T\mathbf{L}\mathbf{f}\}, \tag{33}$$

where $a \to \infty$, under the constraint $\mathbf{f}_i \in \{0, 1\}$. In [Blum et al. 2004], the mincut algorithm is performed multiple times on the graph, by adding random noise on the edge weights. In each iteration, a label is assigned to the unlabeled nodes. Each unlabeled node then takes the label having the maximum assignment frequency. This randomized mincut algorithm provides a confidence measure for the assigned labels. In [Joachims 2003], spectral graph partitioning is performed through the constrained ratiocut algorithm that adds a quadratic penalty to the objective function of standard ratio cut [Hagen and Kahng 1992]:

$$\min_{\mathbf{f}} \{\mathbf{f}^T\mathbf{L}\mathbf{f} + c(\mathbf{f} - \mathbf{g})^T\mathbf{C}(\mathbf{f} - \mathbf{g})\} \quad \text{s.t. } \mathbf{f}^T\mathbf{1} = 0 \text{ and } \mathbf{f}^T\mathbf{f} = N, \tag{34}$$

where c is a regularization parameter and $\mathbf{C}$ is a diagonal matrix, whose i-th diagonal element contains a misclassification cost for node i.

The method proposed in [Talukdar and Crammer 2009] consists a modification of the Adsorption algorithm [Baluja et al. 2008]. Similarly to [Baluja et al. 2008], label propagation is regarded as a random walk, where each node i is associated three probabilities: injection (p_i^{inj}), referring to the case when the walk stops and the predefined label vector $\mathbf{y}$ is returned, continue (p_i^{cont}), expressing the probability that the walk continues to another neighboring node j according to the value of W_{ij}, and abandon or dummy probability (p_i^{abnd}), corresponding to the case of abandoning the walk and assigning a dummy label, expressing uncertainty of the node's label. The dummy probability reflects a node's degree: the higher the degree the higher the value of p^{abnd} is. The introduction of the dummy probability accounts for the fact that high degree nodes can be unreliable, in the sense that they may connect dissimilar nodes. In contrast to the Adsorption algorithm, learning in [Talukdar and Crammer 2009] is stated as a convex optimization problem. Apart from the requirements for the labels of initially labeled nodes to remain unchanged and the smoothness of the labeling function, an additional regularization term is introduced, that accounts for discounting of high degree nodes:

$$\min_{\mathbf{f}} \left\{ \mu_1 \left(\mathbf{y} - \mathbf{f}\right)^T \mathbf{S} \left(\mathbf{y} - \mathbf{f}\right) + \mu_2 \mathbf{f}^T\mathbf{L}\mathbf{f} + \mu_3 \left|\left|\mathbf{f} - \mathbf{r}\right|\right|^2 \right\}, \tag{35}$$

where $\mathbf{r}$ is a vector of length $m + 1$ (m denoting the number of labels), with m first entries equal to zero and the last equal to p_i^{abnd}.

In [Orbach and Crammer 2012], the notion of confidence of the label assignments is incorporated in the learning process, in order to take into consideration the level of agreement between neighboring nodes. Label propagation is treated as an optimization over two sets of parameters, defined for each of N nodes. The first set consists of the score vectors $\mathbf{f}_i, \in \mathbb{R}^m$, representing node's i degree of belonging to each of the m classes. Furthermore, each node is associated with a diagonal matrix $\mathbf{\Sigma}_i \in \mathbb{R}^{m \times m}$, the entries of which express the uncertainties of the corresponding scores in $\mathbf{f}_i$. The optimization problem is defined by the following objective function:

$$\min_{\mathbf{f},\mathbf{\Sigma}} \left\{ \frac{1}{4} \sum_{i,j=1}^{N} w_{ij} \left[(\mathbf{f}_i - \mathbf{f}_j)^T (\mathbf{\Sigma}_i^{-1} + \mathbf{\Sigma}_j^{-1})(\mathbf{f}_i - \mathbf{f}_j) \right] + \frac{1}{2} \sum_{i=1}^{n_l} \left[(\mathbf{f}_{Li} - \mathbf{y}_{Li})^T (\mathbf{\Sigma}_i^{-1} + \frac{1}{\gamma}\mathbf{I})(\mathbf{f}_{Li} - \mathbf{y}_{Li}) \right] + \alpha \sum_{i=1}^{N} Tr\mathbf{\Sigma}_i - \beta \sum_{i=1}^{N} \log\det \mathbf{\Sigma}_i \right\}, \quad (36)$$

where α, β are constant parameters. The first term expresses the requirement for neighboring nodes to have similar scores, the second accounts for the requirement that the scores $\mathbf{f}_L$ of the labeled nodes must be close to the input labels $\mathbf{y}_L$, while the third term forces the uncertainty matrix to be close to a predefined matrix.

3.4. Regularization with graph kernels

In manifold regularization methods with graph kernels, the smoothness constraint (27) is written in the form:

$$\mathcal{S}(\mathbf{f}) = \|\mathbf{f}\|_{\mathcal{H}} = \mathbf{f}^T \mathbf{K} \mathbf{f}, \quad (37)$$

where $\mathbf{K}$ is a kernel associated with the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. Graph kernels capture the local and global structure of the data space. A function $\mathbf{K}$ is considered to be a kernel function, if it is symmetric and positive semi-definite. A kernel matrix is equivalent to the Gram matrix, i.e., the matrix of all possible inner products of the data. Kondor et al. [Kondor and Lafferty 2002] exploited the property that the exponentiation operation on any symmetric matrix $\mathbf{H}$ results to a symmetric and positive semi-definite matrix $\mathbf{K}$:

$$\mathbf{K} = e^{\beta \mathbf{H}} = \lim_{n \to \infty} \left(1 + \frac{\beta \mathbf{H}}{n} \right)^n, \quad (38)$$

to define an exponential family of kernel functions, where $\mathbf{H}$ is called the generator and β is a bandwidth parameter. The exponential kernel $\mathbf{K}$ has the property that, if the generator $\mathbf{H}$ represents the local structure of the data space, then $\mathbf{K}$ represents the global structure of the data space. Equation (38) leads to the following differential equation:

$$\frac{d}{d\beta}\mathbf{K} = \mathbf{H}\mathbf{K}. \quad (39)$$

Equation (39) is the heat equation on graph $\mathbf{G}$ (see Section 4). The resulting kernels are called diffusion or heat kernels. By considering an un-weighted non-directional graph and by choosing:

$$\mathbf{H} = \begin{cases} 1, & \text{node } i \text{ connected to node } j \\ -d_i, & i{=}j \\ 0, & \text{otherwise,} \end{cases} \quad (40)$$

the solution of the heat equation (39) in the infinite discrete space with initial conditions $\mathbf{K}(x_i, x_j) = \delta(x_i - x_j)$, where δ is the Dirac function, is the Gaussian function:

$$\mathbf{K}(x_i, x_j) = \frac{1}{\sqrt{4\pi\beta}} e^{-\frac{\|x_i - x_j\|_2^2}{4\beta}}. \quad (41)$$

In (40) d_i denotes the degree of node i. By setting $\beta = \sigma^2/2$, (41) becomes equal to the Gaussian kernel function:

$$\mathbf{K}(x_i, x_j) = \frac{1}{\sqrt{4\pi\sigma^2}} e^{-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}}. \tag{42}$$

This means that the Gaussian kernel function is a special case of the more general diffusion kernels (41). The same analysis can be also applied in symmetric weighted graphs and/or multigraphs, by setting H_{ij} equal to the sum of weights connecting nodes i and j.

Another way for defining kernel families on graphs comes for the spectral analysis of the graph Laplacian matrix [Smola and Kondor 2003]. Let us define the eigensystem of $\mathbf{L}$ as $\{\lambda_i, \phi_i\}$, where $\mathbf{L} = \sum_i \lambda_i \phi_i \phi_i^T$. The normalized graph Laplacian $\tilde{\mathbf{L}}$ has some very interesting properties for the graph structure [Smola and Kondor 2003], [Zhu et al. 2005]. It is symmetric, positive semi-definite (even if $\mathbf{W}$ is not) and its eigenvalues take values in the range $[0, 2]$, with $\lambda_0 = 0$, where λ_0 is the smallest eigenvalue of $\tilde{\mathbf{L}}$. Moreover, the algebraic multiplicity of λ_0 indicates the number of disjoint graph components. In the case of a regular graph, the matrices $\mathbf{W}, \mathbf{L}, \tilde{\mathbf{L}}$ have the same eigenvectors. Another important property of $\tilde{\mathbf{L}}$ is that the eigenvectors ϕ that correspond to smaller eigenvalues λ of $\tilde{\mathbf{L}}$ are smoother than the eigenvectors that correspond to larger eigenvalues, while the eigenvector that corresponds to $\lambda_0 = 0$ is constant. This means that the function $\sum_{ij} W_{ij}(\phi(i) - \phi(j))^2$ takes smaller values, when ϕ corresponds to a smaller eigenvalue.

The spectral analysis of the normalized graph Laplacian matrix $\tilde{\mathbf{L}}$, namely, the eigenvectors of $\tilde{\mathbf{L}}$ contain information about the graph partitions and, therefore, are an important tool for graph clustering. A popular clustering method based on the spectral analysis of $\tilde{\mathbf{L}}$ is the normalized graph cut algorithm [Shi and Malik 2000], which performs recursive bi-partitions of the graph according to the entry of the eigenvector that corresponds to the second smallest eigenvalue of $\tilde{\mathbf{L}}$ that minimizes the normalized cut (Ncut) disassociation measure. Smola and Kondor [Smola and Kondor 2003] define a class of regularization functionals on graphs of the form:

$$\langle \mathbf{f}, \mathbf{f} \rangle_{\mathcal{H}} = \langle \mathbf{f}, r(\tilde{\mathbf{L}})\mathbf{f} \rangle, \tag{43}$$

where $r(\tilde{\mathbf{L}}) := \sum_i r(\lambda_i)\phi_i \phi_i^T$ and $\langle \cdot \rangle_{\mathcal{H}}$ denotes the inner vector product in the RKHS $\mathcal{H}$ with kernel:

$$\mathbf{K} = \sum_i r^{-1}(\lambda_i)\phi_i \phi_i^T. \tag{44}$$

All kernel functions are derived from (44) with a proper choice of spectrum transform of the Laplacian $r(\lambda)$. For example, the diffusion kernel in [Kondor and Lafferty 2002] is obtained for $r(\lambda) = \exp(\sigma^2/\lambda)$ and the regularized Laplacian $\mathbf{L} + \mathbf{I}/\sigma^2$ [Zhu et al. 2003] is obtained for $r(\lambda) = \lambda + 1/\sigma^2$.

In the general manifold regularization framework [Belkin et al. 2006], the objective is the minimization of:

$$\frac{1}{n_l} \sum_{i=1}^{n_l} V(\mathbf{x}_i, y_i, \mathbf{f}) + \gamma_A \|\mathbf{f}\|_{\mathcal{H}}^2 + \gamma_I \|\mathbf{f}\|_I^2, \tag{45}$$

where the first term is the general form of the cost function on the labeled data, $\|\mathbf{f}\|_{\mathcal{H}}^2$ is a regularization term in the RKHS of the kernel $\mathbf{K}$ and $|\mathbf{f}\|_I^2$ is a regularization term of the geometry of the probability distribution. Laplacian Regularized Least

Squares (LapRLS) and Laplacian Support Vector Machines (LapSVM) [Gomez-Chova et al. 2008] are special cases of manifold regularization.

3.5. Inductive label inference

Graph-based label propagation methods are generally transductive methods, i.e., they operate only on existing data, without being able to handle the arrival of new data. Bengio et al. [Bengio et al. 2006] proposed a common framework in which different label propagation algorithms minimize a quadratic cost function with a closed-form solution obtained by solving a linear system of size equal to the number of available data. This cost function is then expanded in the inductive setting, assigning a label to the incoming data. More precisely, let us assume that all existing data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ have been labeled as $\mathbf{y} = \{\hat{y}_1, \ldots, \hat{y}_N\}$ through a label propagation method. When a new example $\mathbf{x}$ arrives, it is embodied in the graph, with a new weight matrix $\mathbf{W}_\mathbf{x}$. The objective is then to minimize the objective function:

$$\text{constant} + \mu \left(\sum_j W_\mathbf{x}(\mathbf{x}, \mathbf{x}_j)(\hat{y} - \hat{y}_j)^2 + \epsilon \hat{y}^2 \right), \tag{46}$$

with respect to the new label $\hat{y}$. The minimum of (46) is computed by setting the first derivative to zero:

$$\hat{y} = \frac{\sum_j W_\mathbf{x}(\mathbf{x}, \mathbf{x}_j)\hat{y}_j}{W_\mathbf{x}(\mathbf{x}, \mathbf{x}_j) + \epsilon}. \tag{47}$$

If the weight matrix $W_\mathbf{x}$ is extracted by using the k-NN function, then (47) is equivalent to k-NN classification. If $W_\mathbf{x}$ is estimated by employing the Gaussian kernel (4), then (47) is equivalent to the Nadaraya-Watson kernel regression [Bierens 1987].

3.6. Incorporating class prior knowledge

The classification results of label propagation methods can be enhanced in accuracy and robustness by incorporating class prior knowledge during label inference. More specifically, the accuracy of label propagation is increased with the imposition of additional constraints on the class proportions $p_1, \ldots, p_L$, defined as the percentage of data belonging to each class, estimated from the labeled data. In [Zhu and Ghahramani 2002], the authors proposed two methods for exploiting the class prior knowledge: class mass normalization and label bidding. In both methods, for each unlabeled example $\mathbf{x}_i$, a vector $\mathbf{Y}_i = [Y_{i1}, \ldots, Y_{iL}]^T$ is defined, whose j-th value corresponds to the probability that the example belongs to class j, with $\sum_{j=1}^{L} Y_{ij} = 1$. The mass of class j is defined as:

$$m_j = \frac{1}{n_u} \sum_{i=1}^{n_u} Y_{ij}. \tag{48}$$

In class mass normalization, the elements of the j-th column of $\mathbf{Y}$ are scaled by the factor $w_j = \frac{p_j}{m_j}$ and the label assignment for unlabeled sample $\mathbf{x}_i$ is performed according to $\text{argmax}_j\{w_j Y_{ij}\}$. Mass class normalization does not require the knowledge of the exact label (class) proportions. However, if the exact class proportions are known, a label bidding heuristic can be employed. Let us denote by c_j the number of unlabeled examples that are assigned the label l_j, with $\sum_j c_j = n_u$. In each iteration, the unlabeled example $\mathbf{x}_i$ with the highest class probability $\max_j\{Y_{ij}\}$ is found and assigned the label $l_{j_{max}}$, where $j_{max} = \text{argmax}_j\{Y_{ij}\}$, if the number of the already assigned labels $l_{j_{max}}$

does not exceed $c_{j_{max}}$. Otherwise it is ignored and the next highest class probability is searched.

3.7. Label propagation with multiple representations

So far, we considered that the labeled and unlabeled data have a single representation. However, in many real world applications, the data can be represented in more than one feature spaces. For each representation a new graph can be constructed. For example, in the scientific publication network, two graphs may be constructed having as nodes the articles: one that connects each article with its citations and another that connects articles that have at least one common author. In the case of 3D video segments, each video item (node) can be described in terms of color as well as depth (or disparity) feature vectors. The fusion of multiple data representations can be performed either on the graph construction level (early fusion), e.g., by concatenating the separate feature vectors into a global feature vector, or on the decision level (late fusion), e.g., by learning a classification algorithm for each data representation and fusing the classification results. Late fusion is also called "multi-modal fusion" of "multi-modality learning" [Wang et al. 2009a]. A study on early versus late fusion methods for semantic analysis of multi-modal video can be found in [Snoek et al. 2005]. In [Snoek et al. 2005], experimental results on 184 hours of video content showed that the late fusion framework had better performance for most semantic concepts, however with increased computational complexity with respect to early fusion methods.

In one of the first approaches in this area, Joachims et al. [Joachims et al. 2001] employed convex combinations of independent kernels $K(\mathbf{x}_1, \mathbf{x}_2) = \alpha K(\mathbf{x}_1, \mathbf{x}_2) + (1 - \alpha)K(\mathbf{x}_1, \mathbf{x}_2)$, $0 \leq \alpha \leq 1$. The kernels are considered independent, if they are derived from independent data representations. This method is based on the property that any convex combination of kernels produces a new kernel. In a similar notion, a convex combination of the graph Laplacians is employed in [Argyriou et al. 2005], [Tsuda et al. 2005] and [Sindhwani and Niyogi 2005]. These approaches do not discriminate between graphs relevant to the classification task and more irrelevant graphs, which provide no useful information. In order to alleviate this drawback, Kato et al. [Kato et al. 2009] and Wang er al. [Wang et al. 2009a] proposed a propagation method that constructs a convex combination of the graph Laplacians, by optimizing the weights via an iterative process, so that informative graphs are assigned larger coefficients. Moreover, a novel method for optimizing the weights of the convex graph Laplacian combination during the data representation step (i.e., data dimensionality reduction) is introduced in [Zoidi et al. 2014].

First in [Tong et al. 2005] and then in [Wang et al. 2007], [Wang et al. 2009a], the authors extended the single-graph regularization framework proposed in [Zhou et al. 2004] in the case of multiple graphs, as a weighted sum of multiple objective functions:

$$\arg\min_{\mathbf{f}, a_g} \left\{ \sum_g a_g \left(\sum_{i,j} W_{g,ij} \left(\frac{f_i}{\sqrt{D_{g,ii}}} - \frac{f_j}{\sqrt{D_{g,jj}}} \right)^2 + \mu_g \sum_i (f_i - y_i)^2 \right) \right\}, \quad \text{s.t. } \sum_g a_g = 1, \tag{49}$$

which is solved sequentially for the score function $\mathbf{f}$ and the weights a_g. Moreover, in [Tong et al. 2005] a sequential fusion scheme of two graphs is proposed, by sequentially minimizing the following two-stage optimization problem:

$$\mathbf{f}_1^* = \arg\min_{\mathbf{f}} = \mu \mathbf{f}^T (\mathbf{I} - \mathbf{S}^1)\mathbf{f} + (1 - \mu)(\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y}) \tag{50}$$

$$\mathbf{f}_2^* = \arg\min_{\mathbf{f}} = \eta \mathbf{f}^T (\mathbf{I} - \mathbf{S}^2)\mathbf{f} + (1 - \eta)(\mathbf{f} - \mathbf{f}_1^*)^T (\mathbf{f} - \mathbf{f}_1^*), \tag{51}$$

where $\mathbf{S}^1$ and $\mathbf{S}^2$ are the constraints from the first and second graph, respectively. The differences between the linear and sequential approach is in the way the similarity graphs are fused. In the linear case, the score function f is spread through the information from the two graphs and then, the results are fused. In the sequential case, first a label propagation is performed, based on the first similarity graph and the resulting labels are spread using the information of the second graph.

In another approach proposed in [Zhou and Burges 2007] each directed graph is considered to be a Markov chain with a unique stationary distribution similar to [Zhou et al. 2005]. Then they are combined in a mixture of Markov chains framework. In [Xiao et al. 2007], 3D points and 2D images are exploited for multiple view segmentation. Three similarity graphs are constructed, which measure the 3D point coordinate similarity, the 2D color similarity and the patch histogram similarity between two joint points. The joint points are vectors consisting of the coordinates of a 3D point and its corresponding patches in all images. The final graph, representing the joint similarity between two joint points is constructed by summing the three similarity graphs. In [Zhou et al. 2008], multi-graph label propagation for document recommendations is performed, by fusing information of the citation matrix, the author matrix and the venue matrix. An objective function is constructed for each modality and then, they are merged in a single objective function.

Two methods for combining information obtained from the left and right channel of a stereo video for facial image label propagation are introduced in [Zoidi et al. 2013]. In the first one, label propagation is performed through LNP [Wang and Zhang 2006] in the left and right channel separately, producing two classification matrices $\mathbf{F}^L$ and $\mathbf{F}^R$. Each stereo facial image is then assigned to the label that corresponds to the maximum column of the matrix:

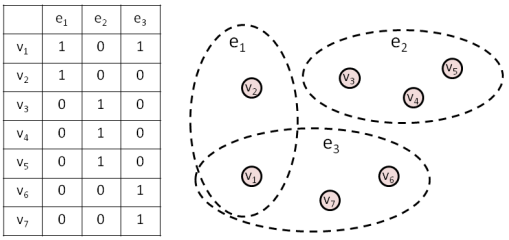
$$\mathbf{F}_{ij}^{max} = \max(\mathbf{F}_{ij}^L, \mathbf{F}_{ij}^R). \tag{52}$$

In the second merging technique, label propagation is performed on the average graph weight matrices of the left and right channel.

3.8. Label propagation with hypergraphs

In all label propagation methods examined so far, the relationships between data are expressed in data pairs. However, there are many real-world applications, in which the data relationships are more complex than pairwise relationships, concerning an arbitrary number of multimedia data (graph nodes). These situations arise when the data are represented by more than one labels. In such applications, the use of the typical pairwise relationships, represented by graphs, leads to loss of information and, consequently, to suboptimal solutions. These complex relationships between data can be effectively represented by hypergraphs.

A hypergraph is a graph, whose edges can connect more than two nodes [Berge 1989]. This means that, in hypergraphs, edges are subsets of nodes. A hypergraph $\mathcal{H}$ is defined as a pair $\mathcal{H} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{V}$ is a finite set of nodes and $\mathcal{E}$ is a set of nonempty subsets of nodes, called hyperedges, which represent the data relationships. In the case when hyperedges contain only two nodes, the hypergraph is equivalent to a simple graph. An example of a hypergraph is shown in Figure 4, where the set of nodes is $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ and the set of hyperedges is $\mathcal{E} = \{e_1, e_2, e_3\}$, where $e_1 = \{v_1, v_2, v_8\}$, $e_2 = \{v_5, v_6, v_7\}$ and $e_3 = \{v_2, v_3, v_4, v_6, v_8\}$. When the hyperedges are associated with possitive weights w_{ij}, the hypergraph is weighted and is denoted by $\mathcal{H} = (\mathcal{E}, \mathcal{V}, w)$. A hyperedge $e \in \mathcal{E}$ is incident with the node $v \in \mathcal{E}$ if $v \in e$. The



	e_1	e_2	e_3
v_1	1	0	1
v_2	1	0	0
v_3	0	1	0
v_4	0	1	0
v_5	0	1	0
v_6	0	0	1
v_7	0	0	1

Fig. 4. Example of a hypergraph.

hypergraph $\mathcal{H}$ is represented by the incidence matrix $\mathbf{H} \in \Re^{|\mathcal{V}|\times|\mathcal{E}|}$, where:

$$h(v,e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e. \end{cases} \tag{53}$$

The degree (strength) of the node $v \in \mathcal{V}$ is defined as the summation of the weights associated with its incident hyperedges $d(v) = \sum_{e\in\mathcal{E}} w(e)h(v,e)$. In a similar way, the degree of hyperedge $e \in \mathcal{E}$ is equal to its cardinality $\delta(e) = \sum_{v\in\mathcal{V}} h(v,e)$. The hypergraph adjacency matrix $\mathbf{A}$, as defined by Zhou et al. in [Zhou et al. 2007] according to the random walk model, is given by:

$$\mathbf{A} = \mathbf{HWH}^T - \mathbf{D}_v, \tag{54}$$

where $\mathbf{W}$ and $\mathbf{D}_v$ are the diagonal matrices of the hyperedge and node weights, respectively. The normalized hypergraph Laplacian matrix is defined accordingly [Zhou et al. 2007]:

$$\mathbf{L} = \mathbf{I} - \frac{1}{2}\mathbf{D}_v^{-1/2}\mathbf{HWH}^T\mathbf{D}_v^{-1/2} = \frac{1}{2}\left(\mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{AD}_v^{-1/2}\right). \tag{55}$$

An alternative definition of the hypergraph Laplacian based on the number of random walks can be found in [Rodríguez 2003].

Let us consider a weighted hypergraph $\mathcal{H} = (\mathcal{E}, \mathcal{V}, w)$, with a subset of nodes $\mathcal{S} \subset \mathcal{V}$ labeled with labels $\mathcal{L} = \{l_j\}_{j=1}^{L}$. Label propagation in hypergraphs refers to the task of label assignment to the unlabeled nodes, under the restriction that nodes belonging to the same hyperedge should be assigned the same label. In one of the earlier works, Zhou et al. [Zhou et al. 2007] presented a hypergraph transductive inference scheme that follows hypergraph clustering. Given a classification function $\mathbf{f} : \mathcal{V} \to \Re^{|\mathcal{V}|}$, the classification decision is given by a framework of the form:

$$\arg\min_{\mathbf{f}}\{R_{emp}(\mathbf{f}) + \mu\Omega(\mathbf{f})\}, \tag{56}$$

where $R_{emp}(\mathbf{f})$ is an empirical loss, $\Omega(\mathbf{f})$ is the clustering objective function and μ is a regularization parameter.

In [Corduneanu and Jaakkola 2004] and [Tsuda 2005], two information regularization frameworks for hypergraph label propagation are presented that employ label probability distributions, instead of deterministic labels. The idea behind the proposed methods is the minimization of the number of bits required to communicate labels for unlabeled data. The predicted label distributions of the unlabeled nodes are derived from the distributions of the labeled nodes and the node relationships in the hyperedges. The framework in [Corduneanu and Jaakkola 2004] minimizes the mixture-type information regularizer (m-regularizer), while the framework in [Tsuda 2005] minimizes the exponential-type information regularizer (e-regularizer), which is the dual of m-regularizer. The advantage of e-regularizer over m-regularizer is that it has

a closed form solution. In [Sun et al. 2008], multiple label propagation is performed through hypergraph spectral learning. The method is based on the property that the hypergraph spectrum captures the correlation among labels. Moreover, an approximate hypergraph spectral learning framework is introduced, for targeting large scale multi-label propagation problems. The framework is based on the approximation of the hypergraph Laplacian matrix $\mathbf{L} \in \Re^{n \times n}$ by $\mathbf{L} = \mathbf{H}\mathbf{H}^T$, where $\mathbf{H} \in \Re^{n \times k}$ has orthonormal columns. In [Chen et al. 2009], a novel method for multi-label propagation based on hypergraph regularization is presented, called Rank-HLapSVM. The method objective is to minimize the ranking loss, while having a large margin. It incorporates the hypergraph Laplacian regularizer $\text{tr}\{\mathbf{F}^T\mathbf{L}\mathbf{F}\}$ in the objective function of Ranking-SVM [Elisseeff and Weston 2001]:

$$\min_{\mathbf{F}} \frac{1}{2}\sum_i \|\mathbf{w}_i\|^2 + \frac{1}{2}\lambda \text{tr}\{\mathbf{F}^T\mathbf{L}\mathbf{F}\} + C\sum_i \frac{1}{|y_i||\bar{y}_i|} \sum_{(p,q)\in y_i \times \bar{y}_i} \xi_{ipq} \tag{57}$$

$$\text{s.t. } \langle \mathbf{w}_p - \mathbf{w}_q, x_i \rangle \geq 1 - \xi_{ipq},\ (p,q) \in y_i \times \bar{y}_i,\ \ \xi_{ipq} \geq 0, \tag{58}$$

where $y_i \subset \mathcal{L}$ is a subset of labels, $\bar{y}_i \subset \mathcal{L}$ is its complementary set and ξ_{ipq} are slack variables. In [Wang et al. 2009], multi-label propagation with multiple hypergraphs was employed for music style classification that integrates three types of information: audio signals, music style correlations and music tag information/correlations. The multiple hypergraphs are combined in a single hypergraph that models the correlations between different modalities, by constructing a hyperedge for each category that contains all the nodes that are relevant to the same category. Then, hypergraph Laplacian regularization of the form $\text{tr}\{\mathbf{F}^T\mathbf{L}\mathbf{F}\}$ is performed, similar to the simple graph case described in subsection 3.3. Hypergraph Laplacian regularization ensures that the label assignment function $\mathbf{F} \in \Re^{n \times L}$ is smooth on the hypergraph nodes. Hypergraph Laplacian regularization for semi supervised label propagation is also performed in [Ding and Yilmaz 2008], [Tian et al. 2009], with applications in image segmentation and gene expression classification, respectively. In [Ding and Yilmaz 2008], a random walk interpretation of hypergraph Laplacian regularization is also presented, as well as the extension of the normalized and the ratio cut (presented in subsection 3.3) to hypergraphs.

4. DIFFUSION PROCESSES IN RELATED AREAS

Diffusion is directly related to label propagation, which is essentially an information diffusion process over graphs/networks that can be modeled by the heat equation [Ding et al. 2007], [Ma et al. 2008]. Therefore, the crossfertilization of the two research topics is very important. In the following sections, the notions of diffusion in physics as well as in social networks are discussed.

4.1. Diffusion in physics

In physics, diffusion describes the flow of mass, energy or momentum in a medium, observed in a bunch of physical processes called *transport phenomena*, such as molecular diffusion and heat transfer [Alonso and Finn 1967]. Molecular diffusion refers to the flux of liquid or gass molecules from regions of high to regions of lower concentration, due to thermal energy dissipation. It can be considered as a gradual mixing of materials having different original concentrations, until an equilibrium of uniform concentration is reached. Similarly, heat conduction is a mode of kinetic energy transfer within and between bodies, due to a temperature gradient. The conduction takes place from bodies (or body parts) at a higher temperature to bodies (or body parts) at a lower temperature. At the equilibrium state (thermal equilibrium), the bodies

reach the same temperature. The evolution of temperature $T(x, y, z, t)$ within a homogeneous, finite, three-dimensional body is described by the following equation, known as heat equation:

$$\frac{\partial T}{\partial t} = \gamma(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}) \,, \tag{59}$$

where $T(x, y, z, t)$ denotes the spatiotemporal temperature diffusion over x, y, z and t, whereas γ denotes heat diffusivity, which is the quotient of thermal conductivity κ and heat capacity c. The theory of heat diffusion has been used in diffusion models of influence [Ma et al. 2008], as well as in label propagation [Wang et al. 2011].

4.2. Models of diffusion in social networks

In a social network context, diffusion refers to the flow of information through the individuals in a network. According to the theory by E. Rogers [Rogers 1962], diffusion describes how an idea or an innovation is adopted by units (e.g. persons, groups) belonging to a social system over time. With respect to the degree of eagerness to adopt an innovation (inovativeness), social units are divided into five categories: innovators, early adopters, early majority, late majority and, finally, laggards. Rogers theory, and especially the terminology regarding the adopter categories, has been adopted in several studies of information diffusion in social networks [Ma et al. 2008], [Kempe et al. 2005], [Kempe et al. 2003]. In order to model diffusion in social networks, various methods have been developed, which can easily be extended to label propagation in multimedia data graphs.

4.2.1. Game theoretic models. Game theory [von Neumann and Morgenstern 1944] can provide methods to build models of real-world social interactions, where the decisions of each individual depend not only on his/her choices, but also on the choices made by others. Each game-theoretic model is based on a game (social interaction), involving a set of players (individuals) and a payoff function that assigns a value to each player, according to his/her choices. Classic game theory studies how players should behave, in order to maximize their payoff. Evolutionary game theory studies the evolution of large populations of players that repeatedly play a game and are subjected to evolutionary changes. Learning game theory studies the dynamics of a population of individuals, who repeatedly play a game and adjust their behavior over time, as a result of their experience. Local interaction games extend the two-player coordination games, under the assumption that there is a large population of players interacting in a social network [Blume 1993], [Ellison 1993] and [Morris 2000].

4.2.2. Virus propagation models. The study of information diffusion often adopts disease-propagation models from epidemiology, due to the obvious analogy of information spread and epidemics [Pastor-Satorras and Vespignani 2001]. In the Susceptible-Infected-Recovered (SIR) model, the subject goes through the following states: first, he is susceptible to the disease then he becomes infected with probability p and finally recovers. After the subject recovers, he is no longer susceptible to the same disease. In the Susceptible-Infected-Susceptible (SIS) model, the subject goes through the following states: first, he is susceptible to the disease then he becomes infected with probability p and, after he recovers he becomes susceptible to the same disease. Regarding information diffusion in a social group, the SIS model can be interpreted as follows: first, the subject is ignorant to the information (susceptible) then he learns the information (infected) and after a period of time forgets it (susceptible)[Xu and Liu 2010].

Applications of virus propagation models in social networks can be found in [Opuszko and Ruhland 2013], [Woo et al. 2011] and [Yagan et al. 2012].

4.2.3. Threshold Models. They were originally proposed by Granovetter [Granovetter 1978] and Schelling [Schelling 1978]. A social network can be modeled with a graph, whose nodes are individuals and its edges represent relationships of influence between them. Each node is assigned a threshold value and each edge is assigned a weight, representing the probability that a node influences its neighbor. A node can be activated (i.e., adopt a behavior), if the fraction of its neighbors that are already active exceeds its threshold value. A common threshold model is the Linear Threshold Model. In this model, a set of nodes are initially active. Each edge is assigned a weight w_{uv} that reflects the influence of node v on u, for which $\sum w_{uv} \leq 1$. In addition, each node u has a threshold θ that depicts the intrinsic tendency of the node to adopt a state, given that its neighbors have already adopted it. At a time step t, a node u is activated only if the sum of weights of its active neighbors exceeds its threshold value θ_u. Once activated, a node remains at this state. If the threshold values are known, the process arising from the Linear Threshold Model is deterministic. However, this hypothesis can be lifted [Kempe et al. 2003], considering that thresholds are randomly drawn from a uniform distribution in $[0, 1]$, independently for each node. Kempe et al. introduced also a General Threshold Model, where the activation criterion is substituted by a monotone activation function f_u on the set of neighboring active nodes of u, taking values in the interval $[0, 1]$. Finally, in [Watts 2002], global cascades are studied, using a threshold model. Global cascades are cascades that affect a large portion of a network, occur rarely and are triggered by a small number of initial seeds. In [Watts 2002], the authors study common properties of global cascades regarding the network connectivity.

4.2.4. Cascade Models. These models study how new ideas spread in a network of individuals by employing a cascade scheme. The underlying assumption in cascade models is that the decision of an individual on adopting a new idea is strongly influenced by the recommendations and decisions of others. The probability that a node v will become active increases monotonically with the number of its active neighbors. A widely used simple cascade model is the Independent Cascade Model [Goldenberg et al. 2001], in which the probability that a node v is activated by an active neighbor u is independent of any previous failed attempts conducted by other nodes. Each active node u has the chance to activate a neighbor node v with a probability $p_v(u)$ at time t. A successful attempt results in the activation of node v at time $t + 1$. Once an active node has realized all attempts to influence its neighbors at time t, it then remains active but no longer contagious. Therefore, each active node attempts to activate neighboring nodes only once. In [Kempe et al. 2003], a generalization of the Independent Cascade Model is also suggested. In the General Cascade Model, the probability that a node u activates a neighbor v depends on the previous attempts for activation committed by its neighbors. The General Cascade Model assumes that the order, in which more than one nodes attempt to activate a node v at time t, does not affect the result. The set of nodes perform their attempts arbitrarily at time t, regardless of the order they do so. Another type of cascade model is the Triggering Model [Kempe et al. 2003], in which the nodes become active from a randomly chosen set of nodes, called ”triggering set”.

4.2.5. Applications of information diffusion models to other areas. The models of diffusion in social networks can be extended to label propagation in multimedia data graphs. A common issue in diffusion models is to maximize the spread of influence through the

network. This consists in finding an initial set of active nodes to start the diffusion process, such that the spread will be maximized. The problem was studied by Domingos and Richardson in [Domingos and Richardson 2001], in a viral marketing context. In terms of label propagation on multimedia data, influence maximization is equivalent to determine the initial set of labeled data that optimize the propagation criterion [Zoidi et al. 2013]. Diffusion methods can also be applied to collaborative filtering. Collaborative filtering algorithms are mainly used in recommendation systems, for making automatic predictions about the interests of a user, by collecting preferences or taste information from many users [Terveen and Hill 2001], [Shang et al. 2010]. These predictions are based on the preferences of other individuals in the network that share the same interests with the user. Label propagation can be adopted in recommendation systems, by treating recommendation as the process of label information propagation from labeled data (i.e., items with ratings) to unlabeled data (i.e., items without ratings) [Wang et al. 2011]. Finally, another field of application of information diffusion are citation networks. In [Shi et al. 2009], Shi et al. studied citation networks of publications in computer science from the perspective of information diffusion. The structural features of the information paths through these networks, as well as their impact on the information flow were analyzed.

5. EVALUATION METRICS

Several evaluation metrics exist, for measuring the performance of label propagation algorithms. The choice for the proper evaluation metric depends on the application. When the scope of label propagation is classification then, all classification accuracy measures can be employed for performance measurement. On the other hand, when label propagation is performed on large graphs with the aim of detecting the communities between the graph nodes, clustering measures are employed.

5.1. Classification evaluation metrics

Classification metrics measure the data percentage that has been assigned the correct label and require knowledge of the groundtruth (actual label of each sample). They are employed in label propagation algorithms that assign competitive labels on unlabeled data, i.e., each sample can be assigned one label from a set of mutually exclusive labels.

5.1.1. Binary evaluation metrics. Binary evaluation metrics are employed when the data belong to two classes. The possible outcome of label propagation is described in Table I. According to this table, the following classification measures are defined.

Table I. Possible label propagation outcome

		ground truth	
		actual label A	**actual label not** A
label propagation outcome	**assigned label** A	true positive (tp)	false positive (fp)
	assigned label not A	false negative (fn)	true negative (tn)

Classification accuracy. Classification accuracy is the simplest measure for validating classification algorithms. It is defined as:

$$\text{classification accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}. \tag{60}$$

Classification accuracy takes values in the range $[0, 1]$. When classification accuracy takes the value 1 it means that all data have been assigned the correct label.

Precision-Recall and F-score. In binary classification tasks, precision measures the purity of the data that have been assigned the label and recall measures the percentage of the retrieved data that should be assigned the label. More specifically, for label A we define:

$$\text{precision} = \frac{|\{\text{data assigned the label } A\} \cap \{\text{data that belong to class } A\}|}{|\{\text{data assigned the label } A\}|} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{61}$$

and

$$\text{recall} = \frac{|\{\text{data assigned the label } A\} \cap \{\text{data that belong to class } A\}|}{|\{\text{data that belong to class } A\}|} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \tag{62}$$

Precision and recall are combined in a single measure by taking their harmonic mean. The resulting measure is called F_β-score and is given by:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \text{precision} + \text{recall}} = \frac{(1 + \beta^2) \cdot \text{tp}}{(1 + \beta^2) \cdot \text{tp} + \beta^2 \cdot \text{fp} + \text{fn}}. \tag{63}$$

When $\beta = 1$, i.e., equation (63) is equal to the harmonic mean of precision and recall, equal importance is given to precision and recall and the popular F_1-score, also known as F-measure, is obtained.

Precision-Recall Break Even Point. The Precision-Recall Break Even Point (PRBEP) is the point in which recall is equal to precision. Let $\mathbf{f}_A \in \Re^M$ be the score vector for label A and τ_A be the threshold that determines whether the sample $\mathbf{x}_i$ is assigned the label A or not, according to the following rule:

$$l(\mathbf{x}_i) = \begin{cases} \text{A} & \text{if } f_{A,i} > \tau_A \\ \text{not A} & \text{otherwise} \end{cases}, \text{for } i = 1, \ldots, M. \tag{64}$$

PRBEP is then calculated by tuning the value of threshold τ_A so that precision and recall have equal values.

Receiver operating characteristic curve . The receiver operating characteristic (ROC) curve is a plot of the true positive rate (also known as sensitivity) over the false positive rate (also known as specificity):

$$\text{true positive rate} = \frac{\text{tp}}{\text{tp}+\text{fn}} \qquad \text{false positive rate} = \frac{\text{fp}}{\text{fp}+\text{tn}}. \tag{65}$$

The ROC curve is constructed by the true and false positive rates for varying values of τ_A (64). True and false positive rates take values between [0, 1]. The propagation accuracy is measured as the area under the receiver operating characteristic curve. When the area is 1, i.e., the label propagation algorithm is 100% accurate, the ROC curve consists of two straight lines, one vertical from point (0,0) to point (0,1) and one horizontal from point (0,1) to point (1,1). When the label propagation algorithm assigns the labels randomly, the ROC curve is a straight line from point (0,0) to point (1,1) and the area under it is equal to 0.5.

5.1.2. Multi-class evaluation metrics. Multi-class evaluation metrics are employed when the data belong to more than two classes. In this case, the multi-class classification problem is divided into multiple binary classification problems according to the one-against-all or the one-against-one validation method. Then, for each binary classification problem the binary evaluation metrics are computed and, the total evaluation metric is calculated by macro- or micro-averaging the respective binary metrics.

Let $\mathcal{L}$ be the set of L labels for the L classes and $F(\text{tp}_l, \text{fp}_l, \text{tn}_l, \text{fn}_l)$ one of the binary evaluation metrics for label $l \in \mathcal{L}$ presented in the previous section. The macro-averaged evaluation metric is calculated by averaging the binary evaluation metrics:

$$F_{macro} = \frac{1}{L} \sum_{l \in \mathcal{L}} F(\text{tp}_l, \text{fp}_l, \text{tn}_l, \text{fn}_l), \tag{66}$$

while the micro-averaged evaluation metric is calculated by:

$$F_{micro} = F\left(\sum_{l \in \mathcal{L}} \text{tp}_l, \sum_{l \in \mathcal{L}} \text{fp}_l, \sum_{l \in \mathcal{L}} \text{tn}_l, \sum_{l \in \mathcal{L}} \text{fn}_l \right). \tag{67}$$

The main difference between macro- and micro-averaging is that, in macro averaging the significance of each class is equal, while in micro-averaging the significance of each per-sample classification decision is equal.

5.2. Clustering evaluation metrics

In the case where label propagation is performed on large graphs, often obtained from social networks, with the scope of finding structures and communities among the graph nodes, label propagation can be viewed as a clustering task. Therefore, clustering metrics, such as modularity and cohesiveness, can be employed for evaluating the propagation performance.

5.2.1. Modularity. Modularity is an indicator of whether the graph nodes partition represents properly the network communities. It is measured by comparing the fraction of edges that connect nodes between different communities (clusters) over the total number of edges that exist in the network. Let B be the number of communities detected through label propagation and $\mathbf{B} \in \Re^{K \times K}$ be the symmetric matrix whose (i, j) value represents the fraction of edges that connect nodes of community i with nodes of community j. Modularity is then defined as the fraction of edges that connect nodes in the same community minus the fraction of edges between nodes in the same community in a network with the same partition but with randomly assigned edges between its nodes:

$$Q = \sum_{k=1}^{K} \left(B_{ii} - \sum_{j=1}^{K} B_{ij} \right). \tag{68}$$

Modularity takes values in the range from 0 (if the network edges were assigned randomly) to 1 (if strong connections exist between nodes in the same community). A second definition of modularity, based on the graph adjacency matrix $\mathbf{A} \in \Re^{N \times N}$ is:

$$Q = \frac{1}{4N} \sum_{ij \in \mathcal{C}} \left(A_{ij} - \frac{k_i k_j}{2N} \right), \tag{69}$$

where $\mathcal{C}$ is the set of node pairs in the same community and k_i the degree of node i. The matrix $\mathbf{D}$ with entries $D_{ij} = A_{ij} - \frac{k_i k_j}{2N}$ is called modularity matrix.

5.2.2. Cohesiveness. Cohesiveness measures how strong are the connections between nodes in the same community. More specific, a community is considered to be cohesive if the nodes of the community are more similar to nodes in the same community than to nodes in different communities. Let K be the number of categories (classes) in which the data belong to. The cohesiveness of community $\mathcal{C}_j$ is then defined as follows:

$$C = -\sum_{i=1}^{K} \frac{n_i}{|\mathcal{C}_j|} \log_2 \left(\frac{n_i}{|\mathcal{C}_j|} \right), \tag{70}$$

where n_i denotes the number of data in community $\mathcal{C}_j$ that belong to class i and $|\mathcal{C}_j|$ is the cardinality of community $\mathcal{C}_j$. When all data in the community belong to the same class the cohesiveness of the community is $C = 0$. Cohesiveness takes the smallest value $C = 1$ when the community consists of equal number of data from each class, i.e., the data are clustered randomly.

6. APPLICATIONS IN DIGITAL MEDIA

As digital devices are becoming more and more affordable and popular, interaction with multimedia content has come to be part of most people's daily routine, leading to an inevitable expansion in the volume of the data. Also, a vast amount of multimedia data is created, accessed and processed on the Web: on-line communities enable users to upload and share pictures (Flickr, Picassa, Photobucket), videos (Youtube, vimeo, Dailymotion) or music (Last.fm, soundcloud), as well as to annotate multimedia objects, according to their semantic content. Furthermore, users of social networking websites such as Facebook, Google+ and Twitter share, rank and annotate multimedia data every day. The growing popularity of the aforementioned communities and networks over the last years has given rise to a huge amount of on-line available multimedia data. The effective handling of large scale multimedia content for applications such as archival searching, indexing, or retrieval, has, therefore, attracted significant research interest.

An essential prerequisite for the success of the aforementioned applications is the annotation of the data, i.e. the assignment of labels (tags) characterizing their semantic content. As a matter of fact, usually only a very small percentage of the data residing in large multimedia collections and websites are annotated. Manual annotation by users is time consuming and often infeasible for large data collections. Additionally, user-provided annotations tend to be subjective. For the above reasons, the development of methods for automatically propagating the known labels of a small set of data to unlabeled data is of great importance. Label propagation has been widely used for semi-supervised annotation of the semantic concepts that appear in video sequences. These concepts may either describe entire videos, video frames, or segments of video frames. There exist several annotation tools based on label propagation algorithms for indexing multimedia data in large repositories [Lin et al. 2003b], [Lin et al. 2003a], [Lin et al. 2003c].

Label propagation in video can be performed in terms of pixels [Chen and Corso 2010], [Badrinarayanan et al. 2010], [Vijayanarasimhan and Grauman 2012], i.e., the label of initially labeled pixels in a handful of video frames is propagated to the pixels of the remaining video frames. In this case, the video pixels represent the graph nodes, while pixel labels describe the semantic concept of the structure it belongs to (e.g., an object in the frame). This procedure essentially leads to intra video frame segmentation and reduces greatly the effort required for the production of pixel-wise semantic labels [Chen and Corso 2011]. Several datasets have been proposed for benchmarking pixel label propagation algorithms [Brostow et al. 2009]. Pixel label propagation takes into account both the spatial relationships of pixels in the same frame, as well as the temporal relationship of the pixels in successive video frames [Badrinarayanan et al. 2010]. A common method for extracting temporal relationship between pixels is optical flow. However, optical flow fails in the case of occlusion and reappearance of an object. This problem is tackled in [Budvytis et al. 2010], with the introduction of a variational expectation maximization model. The performance of pixel label propagation methods depends highly on the initial set of labeled frames. In [Vijayanarasimhan and Grau-

man 2012], an active frame selection method is performed, for selecting the set of video frames from which label propagation will begin, so that the manual labeling effort will be minimized.

Label propagation can be also used for assigning labels that characterize entire video frames [Tang et al. 2009], [Qi et al. 2007] [Wang et al. 2009b], video snippets [Zhang et al. 2012], or identifying the persons that appear in the video frames, [Coppi et al. 2011]. The semantic labels may describe the type of video content (news report, sports, weather report, etc.), the scenes that appear in the video (city, forest, mountain sea, etc.), the shot type (outdoor, indoor, long shot, medium shot, close up, etc.), objects that appear in the videos (faces, cars, animals, etc.), actions that appear in the videos (explosion, human actions, crowd behavior, etc.). In these cases, the graph nodes represent the video frames, the video snippets, and regions of interest (bounding boxes) in the frames that enclose the persons' facial images/bodies, respectively. In such systems, video processing methods are employed, such as shot boundary detection algorithms and automatic object/face detection and tracking. The video shots can be divided into subshots, according to different events detected in the shots [Zhang et al. 2012]. Moreover, in the case of person identity label propagation in videos, the information about the co-appearance of persons in the video can be exploited for increasing the annotation accuracy, i.e., facial images that appear in the same frame obviously belong to different persons, therefore should be assigned different labels [Phi et al. 2011], [Zoidi et al. 2013]. A common benchmark dataset used in such systems is TRECVID [Amir et al. 2003], that consists of approximately 170 hours of TV news videos from 13 programs in 3 languages: English, Arabic and Chinese.

Similarly, image annotation methods employ label propagation in both pixel and image level. The goal of the methods that operate on the pixel level is to automate the segmentation process, rather than propagate labels across images. Starting from a set of initially labeled pixels, usually referred to as seeds, labels are propagated to the remaining image pixels [Grady 2006], [Rubinstein et al. 2012], [Wang et al. 2007], [Kuettel et al. 2012]. In [Rubinstein et al. 2012], label propagation is applied jointly on all the images of a dataset, taking into account image correspondences, instead of propagating the labels in each image individually. In this way, consistency of the annotations of similar entities across different images is enforced. Pixel label propagation also finds application in medical imaging. Segmentation of medical images, like the ones acquired by Magnetic Resonance Imaging (MRI) or Computerized Tomography (CT) scans, is an essential step during diagnosis. Again, segmentation is regarded as a label propagation problem, where labels from initially labeled pixels are propagated to unlabeled pixels [Grady and Funka-Lea 2004], [Yong et al. 2013].

In contrast to pixel label propagation, methods that operate on the image level utilize global descriptors for each image [Chen et al. 2010], [Houle et al. 2013], [Guillaumin et al. 2009]. Instead of relying merely on visual features, some methods use also textual information, combining, thus, image- and word-based graph learning [Liu et al. 2009], [Lee et al. 2013]. The advantage of these approaches is, that a better consistency between image similarity and label similarity is achieved. The majority of the image annotation methods assume that each tag is propagated separately. On the contrary, several recently proposed methods consider multi-label propagation [Chen et al. 2010],[Bao et al. 2011], [Lee et al. 2013]. Instead of propagating each label (tag) individually, these methods propagate the different labels simultaneously, taking the interactions between the tags into account, and allow for improved annotation results, as well as for more efficient computations. Usually, image annotation methods assume, that similar images have similar labels, without considering the fact that each label

characterizes only a local region of the image, while image similarity is computed globally. To deal with this fact, the method proposed in [Bao et al. 2011] takes into account the label locality, by considering the relationships between semantic regions of the images.

It is very often the case, that users provide either incorrect or incomplete annotations for the data. This results in the existence of ”noise” in the dataset labels, which limits the performance of the algorithms applied on it. Several methods have been proposed in order to improve the quality of the labels in image datasets, usually referred to also as retagging or tag ranking algorithms [Tang et al. 2011], [Liu et al. 2009], [Liu et al. 2011], [Wang et al. 2006], [Tang et al. 2014]. The method proposed in [Tang et al. 2011] for annotation of image data incorporates a strategy for refinement of the noisy training data, through a regularization scheme. In [Liu et al. 2011], image retagging is regarded as a multiple graph-based, multi-label learning problem. The proposed method takes into consideration the visual information of the images, the semantic correlation of the tags, as well as user-provided prior knowledge. Propagation is performed using multiple tag-specific similarity graphs (one for each tag), expressing the relationships between images associated with the same tag, as well as a label-similarity graph, which captures the semantic similarity between tags. The tag ranking method in [Tang et al. 2014], considers not only the relationships between entire images, but also the relationships among the visual content in salient regions, a fact that leads to more accurate results. Music re-tagging aims at suppressing the noise existing in tags which characterize a music object (i.e. an artist, album or track), such that the refined tags reflect better the semantic description of the music objects. The method in [Yang et al. 2012] exploits label propagation in order to refine the tags provided either by users or by automatic tagging systems. A label-refinement method is also proposed in [Wu et al. 2013], as part of a graph-based semi-supervised learning algorithm for music emotion recognition.

Another application of label propagation can be found in recommendation systems, which consist an important feature of on-line media sharing communities. Recommendation methods may rely on information related to the users' preferences, expressed through user-provided ratings. These methods are usually referred to as collaborative filtering methods. Alternatively, content-based recommendation methods exploit information associated to the multimedia content itself, derived either from meta-data or by extraction of features from the data. In recommendation systems in large video repositories, such as YouTube, label propagation is performed on multiple video graphs and the recommendation results are personalized for each user [Baluja et al. 2008], [Liu et al. 2007], [Tang et al. 2007]. Recommendation systems exploit co-view information, i.e., if the user watches video A and from co-view statistics it is known that other users who watched video A also saw videos B and C, then these videos will be recommended to the user. In co-view graphs, the nodes are the videos and the edge weights that connect two nodes represent the number of users that have watched the two videos. An alternative way for representing the relationships between the videos and the users is the user-video bipartite graph [Baluja et al. 2008]. A certain video is recommended for the user if the path between the video and the user is short, if there are multiple paths between the video and the user and if the paths between the video and the user avoid high-degree nodes (popular videos). Recommendation systems are also integrated in music sharing websites. The method presented in [Shao et al. 2009] uses both user related information and acoustic content features, in order to take advantage of the information contained in the two types of data. The authors propose a similarity measurement scheme, to calculate the similarity between music tracks and

subsequently, recommendation is regarded as a label propagation process from labeled to unlabeled tracks. Apart from multimedia objects, recommendation may regard social groups which share similar interests with a user. The method proposed in [Yu et al. 2011] consists such an example, where visual content of images along with text annotations are utilized in a label propagation scheme, in order to recommend social groups according to users' personal photo collections.

7. CONCLUSION

Label propagation is a semi-automatic annotation process, with a wide application range, varying from multimedia content annotation to the study of social networks. Most label propagation methods exploit a graph representation for the set of labeled and unlabeled data and their pairwise relationships. These methods were reviewed in this paper. The review focuses on the most important methods for graph construction and label inference that appeared in the last decade. Label propagation is essentially an information diffusion process. Therefore, information diffusion in certain domains, e.g., in innovation adoption and social networks have been reviewed as well, since it can have a multitude of applications in label propagation. Furthermore, the diffusion processes in physics have been presented, since they have greatly influenced the research in this area. In particular, diffusion methods derived from the study of social networks were analyzed. However, several obstacles have yet to be overcome, since the existing methods are sensitive to insufficient training data volumes, the proper choice of multimedia data distance functions and the curse of dimensionality.

REFERENCES

M. Alonso and E.J. Finn. 1967. *Fundamental University Physics*. Addison-Wesley Publishing.

A. Amir, M. Berg, S.F. Chang, W. Hsu, G. Iyengar, C.Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock, and others. 2003. IBM research TRECVID-2003 video retrieval system. *NIST TRECVID-2003* (2003).

A. Argyriou, M. Herbster, and M. Pontil. 2005. Combining Graph Laplacians for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 18*. MIT Press, 67–74.

V. Badrinarayanan, F. Galasso, and R. Cipolla. 2010. Label propagation in video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3265–3272.

S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*. 895–904.

B.K. Bao, B. Ni, Y. Mu, and S. Yan. 2011. Efficient Region-aware Large Graph Construction Towards Scalable Multi-label Propagation. *Pattern Recogn.* 44, 3 (2011), 598–606.

R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. 1994. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA.

M. Belkin, I. Matveeva, and P. Niyogi. 2004. Regularization and semi-supervised learning on large graphs. In *COLT*. Springer, 624–638.

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

M. Belkin, P. Niyogi, and V. Sindhwani. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research* 7 (2006), 2399–2434.

Y. Bengio, O. Delalleau, and N. Le Roux. 2006. Label Propagation and Quadratic Criterion. In *Semi-Supervised Learning*. MIT Press, 193–216.

C. Berge. 1989. *Hypergraphs: combinatorics of finite sets*. Vol. 45. North holland.

K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. 1999. When is nearest neighbor meaningful? In *Database Theory ICDT99*. Springer, 217–235.

H.J. Bierens. 1987. Kernel estimators of regression functions. In *Advances in Econometrics: Fifth World Congress of the Econometric Society*, Vol. 1. 99–144.

M. Bilenko, S. Basu, and R.J. Mooney. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 11.

A. Blum and S. Chawla. 2001. Learning from Labeled and Unlabeled Data using Graph Mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*. 19–26.

A. Blum, J. Lafferty, M.R. Rwebangira, and R. Reddy. 2004. Semi-supervised learning using randomized mincuts. In *Proceedings of the twenty-first international conference on Machine learning (ICML '04)*. ACM, 13–.

L. E. Blume. 1993. The Statistical Mechanics of Strategic Interaction. *Games and Economic Behavior* 5 (1993), 387–424.

I. Borg and P.J.F. Groenen. 2005. *Modern Multidimensional Scaling*. Springer.

G.J. Brostow, J. Fauqueur, and R. Cipolla. 2009. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* 30, 2 (2009), 88 – 97.

I. Budvytis, V. Badrinarayanan, and R. Cipolla. 2010. Label propagation in complex video sequences using semi-supervised learning.. In *BMVC*, Vol. 2257. 2258–2259.

A.Y.C. Chen and J.J. Corso. 2010. Propagating multi-class pixel labels throughout video frames. In *Western New York Image Processing Workshop (WNYIPW)*. 14–17.

A.Y.C. Chen and J.J. Corso. 2011. Temporally consistent multi-class video-object segmentation with the Video Graph-Shifts algorithm. In *IEEE Workshop on Applications of Computer Vision (WACV)*. 614–621.

G. Chen, J. Zhang, F. Wang, C. Zhang, and Y. Gao. 2009. Efficient multi-label classification with hypergraph regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1658–1665.

X. Chen, Y. Mu, S. Yan, and T.S. Chua. 2010. Efficient Large-scale Image Annotation by Probabilistic Collaborative Multi-label Propagation. In *Proceedings of the International Conference on Multimedia (MM '10)*. ACM, 35–44.

H. Cheng, Z. Liu, and J. Yang. 2009. Sparsity induced similarity measure for label propagation. In *IEEE 12th International Conference on Computer Vision*. 317 –324.

D. Coppi, S. Calderara, and R. Cucchiara. 2011. People appearance tracing in video by spectral graph transduction. In *IEEE International Conference on Computer Vision Workshops*. 920–927.

A. Corduneanu and T. Jaakkola. 2004. Distributed information regularization on graphs. *Neural Information Proccessing Systems (NIPS)* (2004).

S.I. Daitch, J.A. Kelner, and D.A. Spielman. 2009. Fitting a graph to vector data. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, 201–208.

J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning (ICML '07)*. ACM, 209–216.

P.S. Dhillon, P.P. Talukdar, and K. Crammer. 2010. Learning better data representation using inference-driven metric learning. In *Proceedings of the ACL 2010 Conference Short Papers (ACLShort '10)*. 377–381.

C. Ding, H. D. Simon, R. Jin, and T. Li. 2007. A learning framework using Green's function and kernel regularization with application to recommender system. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 260–269.

L. Ding and A. Yilmaz. 2008. Image segmentation as learning on hypergraphs. In *Seventh International Conference on Machine Learning and Applications*. IEEE, 247–252.

P. Domingos and M. Richardson. 2001. Mining the Network Value of Customers. In *Proc. of KDD*. 57–66.

A. Elisseeff and J. Weston. 2001. A kernel method for multi-labelled classification. *Advances in neural information processing systems* 14 (2001), 681–687.

G. Ellison. 1993. Learning, Local Interaction, and Coordination. *Econometrica* 61, 5 (1993), 1047–1071.

P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski. 2006. Taming the curse of dimensionality in kernels and novelty detection. In *Applied soft computing technologies: The challenge of complexity*. Springer, 425–438.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. 2004. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*. MIT Press, 513–520.

J. Goldenberg, B. Libai, and E. Muller. 2001. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters* 12, 3 (2001), 211–223.

L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe. 2008. Semisupervised Image Classification With Laplacian Support Vector Machines. *IEEE Geoscience and Remote Sensing Letters* 5, 3 (july 2008), 336 –340.

L. Grady. 2006. Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 28, 11 (2006), 1768–1783.

L. Grady and G. Funka-Lea. 2004. Multi-label Image Segmentation for Medical Applications Based on Graph-Theoretic Electrical Potentials. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*. Lecture Notes in Computer Science, Vol. 3117. Springer Berlin Heidelberg, 230–245.

M. Granovetter. 1978. Threshold models of collective behavior. *Amer. J. Sociology* 83, 6 (1978), 1420–1433.

S. Gregory. 2010. Finding overlapping communities in networks by label propagation. *New Journal of Physics* 12, 10 (2010), 103018.

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. 2009. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *IEEE 12th International Conference on Computer Vision*. 309–316.

L. Hagen and A.B. Kahng. 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 11, 9 (sep 1992), 1074 – 1085.

R.A. Heckemann, J.V. Hajnal, P. Aljabar, D. Rueckert, and A. Hammers. 2006. Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *NeuroImage* 33, 1 (2006), 115 – 126.

S.C.H. Hoi, W. Liu, and S.F. Chang. 2008. Semi-supervised distance metric learning for collaborative image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–7.

M.E. Houle, V. Oria, S. Satoh, and J. Sun. 2013. Annotation Propagation in Image Databases Using Similarity Graphs. *ACM Trans. Multimedia Comput. Commun. Appl.* 10, 1 (2013), 7:1–7:21.

T. Hwang and R. Kuang. 2010. A Heterogeneous Label Propagation Algorithm for Disease Gene Discovery. In *SDM'10*. 583–594.

T. Jebara, J. Wang, and S.F. Chang. 2009. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 441–448.

T. Joachims. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of the international conference on Machine learning (ICML '03)*. ACM, 290–297.

T. Joachims, N. Cristianini, and J. Shawe-Taylor. 2001. Composite Kernels for Hypertext Categorisation. In *Proceedings of the International Conference on Machine Learning (ICML*. Morgan Kaufmann Publishers, 250–257.

I.T. Jolliffe. 2002. *Principal Component Analysis, Second Edition*. Springer.

T. Kato, H. Kashima, and M. Sugiyama. 2009. Robust Label Propagation on Multiple Networks. *IEEE Transactions on Neural Networks* 20, 1 (2009), 35 –44.

D. Kempe, J. Kleinberg, and É. Tardos. 2003. Maximizing the spread of influence through a social network. In *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*.

D. Kempe, J. Kleinberg, and É. Tardos. 2005. Influential Nodes in a Diffusion Model for Social Networks. In *Proceedings of the 32Nd International Conference on Automata, Languages and Programming (ICALP'05)*. 1127–1138.

R.I. Kondor and J. Lafferty. 2002. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the ICML*. 315–322.

D. Kuettel, M. Guillaumin, and V. Ferrari. 2012. Segmentation Propagation in Imagenet. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII (ECCV'12)*. Springer-Verlag, 459–473.

W.Y. Lee, L.C. Hsieh, G.L. Wu, and W. Hsu. 2013. Graph-based Semi-supervised Learning with Multimodality Propagation for Large-scale Image Datasets. *J. Vis. Comun. Image Represent.* 24, 3 (April 2013), 295–302.

S. Letovsky and S. Kasif. 2003. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics* 19 (2003), i197–i204. Issue 1.

C.Y. Lin, B.L. Tseng, M. Naphade, A. Natsev, and J.R. Smith. 2003c. VideoAL: a novel end-to-end MPEG-7 video automatic labeling system. In *IEEE International Conference on Image Processing*, Vol. 3. III–53–6 vol.2.

C.Y. Lin, B.L. Tseng, and J.R. Smith. 2003a. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *Proceedings of the TRECVID 2003 Workshop*.

C.Y. Lin, B.L. Tseng, and J.R. Smith. 2003b. VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learning. In *IEEE International Conference on Multimedia and Expo*.

D. Liu, X.S. Hua, L. Yang, M. Wang, and H.J. Zhang. 2009. Tag Ranking. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, New York, NY, USA, 351–360.

D. Liu, S. Yan, X.S. Hua, and H.J. Zhang. 2011. Image Retagging Using Collaborative Tag Propagation. *IEEE Transactions on Multimedia* 13, 4 (2011), 702–712.

J. Liu, W. Lai, X.S. Hua, Y. Huang, and S. Li. 2007. Video search re-ranking via multi-graph propagation. In *Proceedings of the 15th international conference on Multimedia (MULTIMEDIA '07)*. 208–217.

J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. 2009. Image Annotation via Graph Learning. *Pattern Recogn.* 42, 2 (2009), 218–228.

J. Long, J. Yin, W. Zhao, and E. Zhu. 2008. Graph-Based Active Learning Based on Label Propagation. In *Modeling Decisions for Artificial Intelligence*. Lecture Notes in Computer Science, Vol. 5285. Springer Berlin / Heidelberg, 179–190.

H. Ma, H. Yang, M.R. Lyu, and I. King. 2008. Mining Social Networks Using Heat Diffusion Processes for Marketing Candidates Selection. In *CIKM*.

M. Maier, U. Von Luxburg, and M. Hein. 2008. Influence of graph construction on graph-based clustering measures. In *Neural Information Processing Systems*. 1025–1032.

S. Morris. 2000. Contagion. *Review of Economic Studies* 67 (2000).

M.E.J. Newman. 2001. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E* 64 (2001), 016132. Issue 1.

N. Nguyen and Y. Guo. 2008. Metric learning: A support vector approach. *Machine Learning and Knowledge Discovery in Databases* (2008), 125–136.

Z.Y. Niu, D.H. Ji, and C.L. Tan. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*. 395–402.

T. Opsahl, F. Agneessens, and J. Skvoretz. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks* 32, 3 (2010), 245 – 251.

M. Opuszko and J. Ruhland. 2013. Impact of the Network Structure on the SIR Model Spreading Phenomena in Online Networks. In *ICCGI 2013: The Eighth International Multi-Conference on Computing in the Global Information Technology*.

M. Orbach and K. Crammer. 2012. Graph-Based Transduction with Confidence. In *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II (ECML PKDD'12)*. Springer-Verlag, Berlin, Heidelberg, 323–338.

R. Pastor-Satorras and A. Vespignani. 2001. *Epidemic Dynamics and Endemic States in Complex Networks*. Physical Review E.

T.P. Phi, T. Tuytelaars, and M.F. Moens. 2011. Naming People in News Videos with Label Propagation. *IEEE MultiMedia* 18, 3 (march 2011), 44 –55.

G.J. Qi, X.S. Hua, Y. Rui, J. Tang, T. Mei, and H.J. Zhang. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th international conference on Multimedia*. 17–26.

D. Rao and D. Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*. 675–682.

R. Rao. 2002. *Probabilistic Models of the Brain: Perception and Neural Function*. MIT Press.

J.A. Rodríguez. 2003. On the Laplacian spectrum and walk-regular hypergraphs. *Linear and Multilinear Algebra* 51, 3 (2003), 285–297.

E. Rogers. 1962. *Diffusion of innovations*. Free Press, New York.

S.T. Roweis and L.K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (2000), pp. 2323–2326.

M. Rubinstein, C. Liu, and W.T. Freeman. 2012. Annotation Propagation in Large Image Databases via Dense Image Correspondence. In *Computer Vision ECCV 2012*. Lecture Notes in Computer Science, Vol. 7574. Springer Berlin Heidelberg, 85–99.

H. Rue and L. Held. 2005. *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall.

O. Russakovsky, J. Deng, Z. Huang, A. C. Berg, and L. Fei-Fei. 2013. Detecting avocados to zucchinis: what have we done, and where are we going?. In *International Conference on Computer Vision (ICCV)*.

V. Satuluri and S. Parthasarathy. 2009. Scalable graph clustering using stochastic flows: applications to community discovery. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 737–746.

L.K. Saul and S.T. Roweis. 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research* 4 (2003), 119–155.

T. Schelling. 1978. *Micromotives and Macrobehavior*. Norton.

B.D. Shai, L. Tyler, and P. Dávid. 2008. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *In: 21st Annual Conference on Learning Theory*.

M.S. Shang, Z.K. Zhang, T. Zhoub, and Y.C. Zhang. 2010. Collaborative filtering with diffusion-based similarity on tripartite graphs. *Physica A* 389, 6 (2010), 1259–1264.

B. Shao, D. Wang, T. Li, and M. Ogihara. 2009. Music Recommendation Based on Acoustic Features and User Access Patterns. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 8 (2009), 1602–1611.

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

X. Shi, B. Tseng, and L. Adamic. 2009. Information Diffusion in Computer Science Citation Networks. In *Proceedings of the International Conference on Weblogs and Social Media*.

V. Sindhwani and P. Niyogi. 2005. A co-regularized approach to semi-supervised learning with multiple views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*.

A. Singh, R.D. Nowak, and X. Zhu. 2008. Unlabeled data: Now it helps, now it doesn't. In *NIPS'08*. 1513–1520.

A. Smola and R. Kondor. 2003. Kernels and Regularization on Graphs. In *Proc. Conf. on Learning Theory and Kernel Machines*. 144–158.

C.G.M. Snoek, M. Worring, and A.W.M. Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA '05)*. 399–402.

M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP (EMNLP '11)*. 53–63.

A. Subramanya and J. Bilmes. 2011. Semi-Supervised Learning with Measure Propagation. *J. Mach. Learn. Res.* 12 (2011), 3311–3370.

L. Sun, S. Ji, and J. Ye. 2008. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 668–676.

M. Szummer and T. Jaakkola. 2002. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems*. MIT Press, 945–952.

P.P. Talukdar. 2009. *Topics in Graph Construction for Semi-Supervised Learning*. Technical Report, University of Pennsylvania.

P. P. Talukdar and K. Crammer. 2009. New Regularized Algorithms for Transductive Learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II (ECML PKDD '09)*. Springer-Verlag, Berlin, Heidelberg, 442–457.

J. Tang, R. Hong, S. Yan, T.S. Chua, G.J. Qi, and R. Jain. 2011. Image annotation by kNN-sparse graph-based label propagation over noisily tagged web images. *ACM Trans. Intell. Syst. Technol.* 2, 2 (2011), 14:1–14:15.

J. Tang, X.S. Hua, G.-J. Qi, T. Mei, and X. Wu. 2007. Anisotropic Manifold Ranking for Video Annotation. In *IEEE International Conference on Multimedia and Expo*. 492–495.

J. Tang, X.-S. Hua, G.-J. Qi, Y. Song, and X. Wu. 2008. Video Annotation Based on Kernel Linear Neighborhood Propagation. *IEEE Transactions on Multimedia* 10, 4 (june 2008), 620 –628.

J. Tang, X.-S. Hua, M. Wang, Z. Gu, G.-J. Qi, and X. Wu. 2009. Correlative Linear Neighborhood Propagation for Video Annotation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39, 2 (april 2009), 409 –416.

J.. Tang, M. Li, Z. Li, and C. Zhao. 2014. Tag ranking based on salient region graph propagation. *Multimedia Systems* (2014), 1–9.

J. Tang, S. Yan, R. Hong, G.J. Qi, and T.S. Chua. 2009. Inferring semantic concepts from community-contributed images and noisy tags. In *Proceedings of the 17th ACM international conference on Multimedia (MM '09)*. ACM, 223–232.

J.B. Tenenbaum, V. De Silva, and J.C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500 (2000), 2319–2323.

L. Terveen and W. Hill. 2001. *Beyond Recommender Systems: Helping People Help Each Other*. HCI In The New Millennium, Jack Carroll, ed., Addison-Wesley.

Z. Tian, T. Hwang, and R. Kuang. 2009. A hypergraph-based learning algorithm for classifying gene expression and arrayCGH data with prior knowledge. *Bioinformatics* 25, 21 (2009), 2831–2838.

H. Tong, J. He, M. Li, C. Zhang, and W.Y. Ma. 2005. Graph based multi-modality learning. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 862–871.

I.W. Tsang and J.T. Kwok. 2006. Large-scale sparsified manifold regularization. In *Advances in Neural Information Processing Systems (NIPS) 19*.

K. Tsuda. 2005. Propagating distributions on a hypergraph by dual information regularization. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 920–927.

K. Tsuda, H. Shin, and B. Schölkopf. 2005. Fast protein classification with multiple networks. *Bioinformatics* 21, 2 (2005), 59–65.

S. Vijayanarasimhan and K. Grauman. 2012. Active frame selection for label propagation in videos. In *Proceedings of the 12th European conference on Computer Vision - Volume Part V (ECCV'12)*. 496–509.

J. von Neumann and O. Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

C. Wang, F. Jing, L. Zhang, and H.J. Zhang. 2006. Image Annotation Refinement Using Random Walk with Restarts. In *Proceedings of the 14th Annual ACM International Conference on Multimedia (MULTIMEDIA '06)*. ACM, 647–650.

D. Wang, I. King, and K.S. Leung. 2011. "Like Attracts Like!"– A Social Recommendation Framework Through Label Propagation. In *Proceedings of SIGIR2011 Workshop on Social Web Search and Mining: Content Analysis Under Crisis*.

F. Wang, X. Wang, and T. Li. 2007. Efficient label propagation for interactive image segmentation. In *Sixth International Conference on Machine Learning and Applications, 2007. ICMLA 2007.* 136–141.

F. Wang, X. Wang, B. Shao, T. Li, and M. Ogihara. 2009. Tag integrated multi-label music style classification with hypergraph. *Proc. 10th International Society for Music Information Retrieval* (2009), 363–368.

F. Wang and C. Zhang. 2006. Label propagation through linear neighborhoods. In *Proceedings of the 23rd international conference on Machine learning (ICML '06)*. ACM, 985–992.

J. Wang, F. Wang, C. Zhang, H.C. Shen, and L. Quan. 2009. Linear Neighborhood Propagation and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 9 (sept. 2009), 1600 –1615.

J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li. 2012. Scalable k-NN graph construction for visual descriptors. In *CVPR*.

M. Wang, X.S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song. 2009a. Unified Video Annotation via Multigraph Learning. *IEEE Transactions on Circuits and Systems for Video Technology* 19, 5 (2009), 733–746.

M. Wang, X.S. Hua, J. Tang, and R. Hong. 2009b. Beyond Distance Measurement: Constructing Neighborhood Similarity for Video Annotation. *IEEE Transactions on Multimedia* 11, 3 (2009), 465–476.

M. Wang, X.S. Hua, X. Yuan, Y. Song, and L.R. Dai. 2007. Optimizing multi-graph learning: towards a unified video annotation scheme. In *Proceedings of the 15th international conference on Multimedia*. ACM, 862–871.

S. Wasserman and K. Faust. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.

D.J. Watts. 2002. A simple model of global cascades on random networks. In *Proceedings of the National Academy of Sciences*. 5766–5771.

K.Q. Weinberger and L.K. Saul. 2009. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research* 10 (2009), 207–244.

J. Weston, C. Leslie, E. Ie, D. Zhou, and A. Elisseeff. 2005. Semi-supervised protein classification using cluster kernels. *Bioinformatics* 21 (2005), 3241–3247. Issue 15.

J. Woo, J. Son, and H. Chen. 2011. An SIR model for violent topic diffusion in social media. In *2011 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 15–19.

B. Wu, E. Zhong, H. Hu, A. Horner, and Q. Yang. 2013. SMART: Semi-Supervised Music Emotion Recognition with Social Tagging. In *Proceedings of 2013 SIAM International Conference on Data Mining (SDM'2013)*. ACM.

J. Xiao, J. Wang, P. Tan, and L. Quan. 2007. Joint affinity propagation for multiple view segmentation. In *IEEE 11th International Conference on Computer Vision*. IEEE, 1–7.

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. 2002. Distance Metric Learning, with Application to Clustering with Side-information. In *Advances in Neural Information Processing Systems 15*, Vol. 15. 505–512.

B. Xu and L. Liu. 2010. Information diffusion through online social networks. In *IEEE International Conference on Emergency Management and Management Sciences*. 53 –56.

O. Yagan, D. Qian, J. Zhang, and D. Cochran. 2012. Information diffusion in overlaying social-physical networks. In *CISS*. 1–6.

R. Yan, L. Yang, and A. Hauptmann. 2003. Automatically labeling video data using multi-class active learning. In *Ninth IEEE International Conference on Computer Vision*. IEEE, 516–523.

R. Yan, J. Zhang, L. Yang, and A. Hauptmann. 2006. A discriminative learning framework with pairwise constraints for video object classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 4 (2006), 578–593.

L. Yang. 2006. Distance Metric Learning: A Comprehenive Survey. (2006). http://www.cs.cmu.edu/~liuy/frame_survey_v2.pdf

L. Yang, D. Ji, G. Zhou, Y. Nie, and G. Xiao. 2006. Document re-ranking using cluster validation and label propagation. In *Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06)*. ACM, 690–697.

Y.H. Yang, D. Bogdanov, P. Herrera, and M. Sordo. 2012. Music Retagging Using Label Propagation and Robust Principal Component Analysis. In *Proceedings of the 21st International Conference Companion on World Wide Web (WWW '12 Companion)*. ACM, 869–876.

Z. Yong, L. Weishi, Z. Yang, Z. Gang, Q. Dongxiang, Z. Qi, H. Ying, W. Haifeng, H. Xiaobo, and H. Jiaming. 2013. Brain MRI Segmentation with Label Propagation. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 2, 5 (2013), 158–163.

J. Yu, X. Jin, J. Han, and J. Luo. 2011. Collection-based Sparse Label Propagation and Its Application on Social Group Suggestion from Photos. *ACM Trans. Intell. Syst. Technol.* 2, 2, Article 12 (2011), 21 pages.

T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu. 2012. A Generic Framework for Video Annotation via Semi-Supervised Learning. *IEEE Transactions on Multimedia* 14, 4 (2012), 1206–1219.

D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. MIT Press, 321–328.

D. Zhou and C.J.C. Burges. 2007. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th international conference on Machine learning (ICML '07)*. ACM, 1159–1166.

D. Zhou, J. Huang, and B. Schölkopf. 2005. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd international conference on Machine learning (ICML '05)*. ACM, 1036–1043.

D. Zhou, J. Huang, and B. Schölkopf. 2007. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems* 19 (2007), 1601.

D. Zhou and B. Schölkopf. 2004. Learning from labeled and unlabeled data using random walks. In *Proceedings of the 26th DAGM Symposium on Pattern Recognition*. Springer, 237–244.

D. Zhou, S. Zhu, K. Yu, X. Song, B.L. Tseng, H. Zha, and C.L. Giles. 2008. Learning multiple graphs for document recommendations. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 141–150.

G.D. Zhou and F. Kong. 2009. Global learning of noun phrase anaphoricity in coreference resolution via label propagation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2 (EMNLP '09)*. 978–986.

X. Zhu. 2008. *Semi-Supervised Learning Literature Survey*. Technical Report, University of Wisconsin - Madison.

X. Zhu and Z. Ghahramani. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation*. Technical Report. School of CS, CMU.

X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML*. 912–919.

X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. 2005. Nonparametric Transforms of Graph Kernels for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, Vol. 17. MIT Press, 1641–1648.

X. Zhu, J. Lafferty, and Z. Ghahramani. 2003. *Semi-Supervised Learning: From Gaussian Fields to Gaussian Processes*. Technical Report. School of CS, CMU.

O. Zoidi, N. Nikolaidis, and I. Pitas. 2013. Exploiting Clustering and Stereo Information in Label Propagation of Facial Images. In *IEEE Symposium Series on Computational Intelligence*.

O. Zoidi, A. Tefas, N. Nikolaidis, and I. Pitas. 2014. Person identity label propagation in stereo videos. *IEEE Transactions on Multimedia* 16, 5 (2014), 1358 – 1368.