Graph-Based Regularization for Transductive Class-Membership Prediction

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

LACAM Laboratory – Dipartimento di Informatica
Università degli Studi di Bari Aldo Moro – via E. Orabona, 4 - 70125 Bari - Italia
firstName.lastName@uniba.it

Abstract. Considering the increasing availability of structured machine processable knowledge in the context of the Semantic Web, only relying on purely deductive inference may be limiting. This work proposes a new method for similarity-based class-membership prediction in Description Logic knowledge bases. The underlying idea is based on the concept of *propagating* class-membership information among similar individuals; it is non-parametric in nature and characterized by interesting complexity properties, making it a potential candidate for large-scale transductive inference. We also evaluate its effectiveness with respect to other approaches based on inductive inference in SW literature.

1 Introduction

Standard Semantic Web (SW) reasoning services rely on purely deductive inference. However, this may be limiting, e.g. due to the complexity of reasoning tasks, availability and correctness of structured knowledge. Approximate deductive and inductive inference were discussed as a possible approach to try to overcome such limitations [25]. Various proposals to extend inductive inference methods towards SW formalisms have been discussed in SW literature: inductive methods can perform some sort of approximate and uncertain reasoning and derive conclusions which are not derivable or refutable from the knowledge base [25].

This work proposes a novel method for transductive inference on Description Logic (DL) representations. In the class-membership prediction task, discriminative methods proposed so far ignore unlabeled problem instances (individuals for which the value of such class-membership is unknown); however, accounting for unlabeled instances during learning can provide more accurate results if some conditions are met [6, 35]. Generative methods, on the other hand, try to model a joint probability distribution on both instances and labels, thus facing a possibly harder learning problem than only predicting the most probable label for any given instance.

In Sect. 2 we will shortly survey related works, and introduce a variant to the classic class-membership prediction problem. In Sect. 3 we will introduce the proposed method: the assumptions it relies on, and how it can be used

for class-membership prediction on large and Web scale ontological knowledge bases. In Sect. 4 we will provide empirical evidence for the effectiveness of the proposed method with respect to other methods in SW literature. In Sect. 5 we summarize the proposed approach, outline its limitations and discuss possible future research directions.

2 Preliminaries

A variety of approaches have been proposed in the literature for solving the class-membership prediction problem, either *discriminative* or *generative* [21]. Assuming instances are i.i.d. samples from a distribution P ranging over a space $X \times Y$ (where X is the space of instances and Y a set of labels), *generative* prediction methods first build an estimate $\hat{P}$ of the joint probability distribution $P(X, Y)$, and then use it to infer $\hat{P}(Y \mid x) = \hat{P}(Y, x)/\hat{P}(x)$ for a given, unlabeled instance $x \in X$. On the other hand, *discriminative* methods simply aim at estimating when $P(y \mid x) \geq 0.5$, for any given $(x, y) \in X \times Y$ (thus facing a possibly easier problem than estimating a joint probability distribution over $X \times Y$). The following shortly surveys class-membership prediction methods proposed so far.

2.1 Discriminative Methods

Some of the approaches proposed for solving the class-membership prediction problem are similarity-based. For instance, methods relying on the k-Nearest Neighbors (k-NN) algorithm are discussed in [8, 25]. A variety of (dis-)similarity measures between either individuals or concepts have been proposed: according to [5], they can be based on *features* (where objects are characterized by a set of features, such as in [16]), on the *semantic-network* structure (where background information is provided in the form of a semantic network, such as in [10, 17]) or on the *information content* (where both the semantic network structure and population are considered, such as in [9]).

Kernel-based algorithms [27] have been proposed for various learning tasks from DL-based representations. This is made possible by the existence of a variety of kernel functions, either for concepts or individuals (such as [11, 4, 13]).

By (implicitly) projecting instances into an high-dimensional feature space, kernel functions allow to adapt a multitude of machine learning algorithms to structured representations. SW literature includes methods for inducing robust classifiers [12] or learning to rank [14] from DL knowledge bases using kernel methods.

2.2 Generative Methods

A generative model for learning from formal ontologies is proposed in [26]: each individual is associated to a *latent variable* (similar to a *cluster indicator*) which influences its attributes and the relations it participates in. It proposes using a

Nonparametric Bayesian model for automatically selecting the number of possible values for such latent variables, with an inference method based on Markov Chain Monte Carlo where posterior sampling is constrained by a predefined set of DL axioms.

A different generative model is proposed in [23]: it focuses on learning theories in a probabilistic extension of the $\mathcal{ALC}$ DL named CR$\mathcal{ALC}$, and using DL refinement operators to efficiently explore the space of concepts. It is inspired by literature on Bayesian Logic Programs [19].

2.3 Semi-Supervised and Transductive Learning

Classic discriminative learning methods tend to ignore unlabeled instances. However, real life scenarios are usually characterized by an abundance of unlabeled instances and a few labeled ones [6, 35]. This may also be the case for class-membership prediction from formal ontologies: class-membership relations may be difficult to obtain during ontology engineering tasks (e.g. due to availability of domain experts) and inference (e.g. since deciding instance-membership may have an intractable time complexity in some languages).

Using unlabeled instances during learning is generally known in the machine learning community as *Semi-Supervised Learning* [6, 35] (SSL). A variant to this setting is known as *Transductive Learning* [31] and refers to finding a labeling only to unlabeled instances provided in the training phase, without necessarily generalizing to unseen instances (and thus resulting into a possibly *simpler* learning problem). If the marginal distribution of instances P_X is informative with respect to the conditional probability distribution $P(Y \mid x)$, accounting for unlabeled instances during learning can provide more accurate results [6, 35].

A possible approach is including terms dependent from P_X into the objective function. This results in the two fundamental assumptions [6]:

- **Cluster assumption** – The joint probability distribution $P(X, Y)$ is structured in such a way that points in the same *cluster* are likely to have the same label.
- **Manifold assumption** – Assume that the distribution P_X is supported on a low-dimensional manifold: then, $P(Y \mid x)$ *varies smoothly*, as a function of x, with respect to the underlying structure of the manifold.

In the following sections, we discuss a similarity-based, non-parametric and computationally efficient method for predicting missing class-membership relations. This method is discriminative in nature, but also accounts for unknown class-membership during learning.

We will face a slightly different version of the classic class-membership prediction problem, namely *transductive class-membership prediction*. It is inspired to the *Main Principle* in [31]: "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more

general intermediate problem". In this setting, the learning algorithm only aims at estimating the class-membership relation of interest for a given training set of individuals, without necessarily being able to generalize to individuals outside such set.

In this work, we formalize the transductive class-membership prediction problem as a cost minimization problem: given a set of training individuals $\mathrm{Ind}_C(\mathcal{K})$ whose class-membership relation to a target concept C is either known or unknown, find a function $f^* : \mathrm{Ind}_C(\mathcal{K}) \to \{+1, -1\}$ defined over training individuals and returning a value $+1$ (resp. -1) if the individual likely to be a member of C (resp. $\neg C$), minimizing a given cost function. More formally:

Definition 1. *(Transductive Class-Membership Prediction) The Transductive Class-Membership Prediction problem can be formalized as follows:*

- **Given:**
  - *a* target *concept C;*
  - *a set of training individuals $Ind_C(\mathcal{K})$ in a knowledge base $\mathcal{K}$ partitioned in positive, negative and neutral examples or, more formally, such that:*
    $Ind_C^+(\mathcal{K}) = \{a \in Ind_C(\mathcal{K}) \mid \mathcal{K} \models C(a)\}$ *positive examples,*
    $Ind_C^-(\mathcal{K}) = \{a \in Ind_C(\mathcal{K}) \mid \mathcal{K} \models \neg C(a)\}$ *negative examples,*
    $Ind_C^0(\mathcal{K}) = \{a \in Ind_C(\mathcal{K}) \mid \mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a)\}$ *neutral examples;*
  - *A* cost function $cost(\cdot) : \mathcal{F} \mapsto \mathbb{R}$*, specifying the* cost *associated to a set of class-membership relations assigned to training individuals by $f \in \mathcal{F}$, where $\mathcal{F}$ is a space of labeling functions of the form $f : Ind_C(\mathcal{K}) \mapsto \{+1, -1\}$;*
- **Find** *a labeling function $f^* \in \mathcal{F}$ minimizing the given cost function with respect to training individuals $Ind_C(\mathcal{K})$:*

$$f^* \leftarrow \arg\min_{f \in \mathcal{F}} cost(f).$$

The function f^* can then be used to estimate the class-membership relation with respect to the target concept C for all training individuals $a \in \mathrm{Ind}_C(\mathcal{K})$: it will return $+1$ (resp. -1) if an individual is likely to be a member of C (resp. $\neg C$). Note that the function is defined on the whole set of training individuals; therefore it can possibly contradict already known class-membership relations (thus being able to handle noisy knowledge). If $\mathrm{Ind}_C(\mathcal{K})$ is finite, the space of labeling functions $\mathcal{F}$ is also finite, and each function $f \in \mathcal{F}$ can be equivalently expressed as a vector in $\{-1, +1\}^n$, where $n = |\mathrm{Ind}_C(\mathcal{K})|$.

3 Propagating Class-Membership Information Among Individuals

This section discusses a *graph-based semi-supervised* [35] method for class-membership prediction from DL representations. The proposed method relies on a weighted *semantic similarity graph*, where nodes represent positive,

negative and neutral examples of the transductive class-membership prediction problem, and weighted edges define similarity relations among such individuals.

More formally, let $\mathcal{K}$ be a knowledge base, $\mathrm{Ind}_C(\mathcal{K})$ a set of training individuals with respect to a target concept C in $\mathcal{K}$, and $Y = \{-1, +1\}$ a space of labels each corresponding to a type of class-membership relation with respect to C. Each training individual $a \in \mathrm{Ind}_C(\mathcal{K})$ is associated to a label, which will be $+1$ (resp. -1) if $\mathcal{K} \models C(a)$ (resp. $\mathcal{K} \models \neg C(a)$), and will be unknown otherwise, thus representing an unlabeled instance. For defining a cost over functions $f \in \mathcal{F}$, the proposed method relies on *regularization by graph*: the learning process aims at finding a labeling function that is both consistent with given labels, and changes smoothly between similar instances (where similarity relations are encoded in the semantic similarity graph). This can be formalized through a *regularization framework*, using a measure of the consistency to the given labels as a loss function, and a measure of smoothness among the similarity graph as a regularizer.

Several cost functions have been proposed in SSL literature. An appealing class of functions, from the side of computational cost, relies on the *quadratic cost criterion* framework [6, ch. 11]: for this class of functions, a closed form solution to the cost minimization problem can be found efficiently (subsection 3.2).

3.1 Semantic Similarity Graph

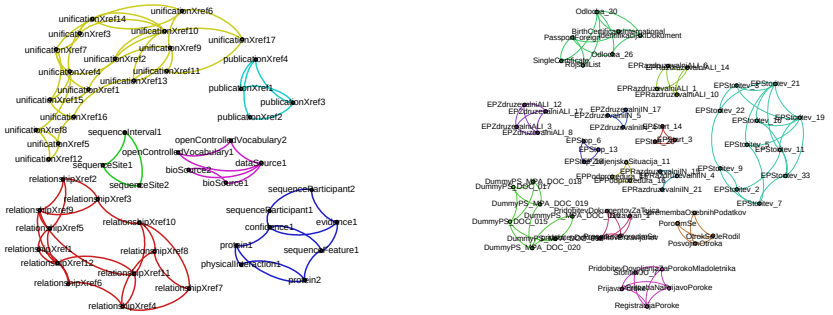
A similarity graph can be represented with a weight matrix $\mathbf{W}$, where the value of $\mathbf{W}_{ij}$ represents the strength of the similarity relation between two training examples x_i and x_j. In graph-based SSL literature, $\mathbf{W}$ is often obtained either as a Nearest Neighbor (NN) graph (where each instance is connected to the k most similar instances in the graph, or to those with a distance under a radius ϵ); or by means of a kernel function, such as the Gaussian kernel.

Finding the best way to construct $\mathbf{W}$ is an active area of research. In [6, ch. 20] authors discuss a method to combine multiple similarity measures in the context of protein function prediction, while [18, 32, 1] propose different methods for data-driven similarity graph construction.

When empirically evaluating the proposed method, we employ the family of dissimilarity measures between individuals in a DL knowledge base defined in [25], since it does not constrain to any particular family of DLs. We refer to the resulting similarity graph among individuals in a formal ontology as the *semantic similarity graph*. Given a set of concept descriptions $F = \{F_1, \ldots, F_n\}$ and a weight vector $\mathbf{w}$, such family of dissimilarity measures $d_p^F : Ind_C(\mathcal{K}) \times Ind_C(\mathcal{K}) \mapsto [0, 1]$ is defined as:

$$\delta_i(x, y) = \begin{cases} 0 \text{ if } (\mathcal{K} \models F_i(x) \wedge \mathcal{K} \models F_i(y)) \vee (\mathcal{K} \models \neg F_i(x) \wedge \mathcal{K} \models \neg F_i(y)) \\ 1 \text{ if } (\mathcal{K} \models F_i(x) \wedge \mathcal{K} \models \neg F_i(y)) \vee (\mathcal{K} \models \neg F_i(x) \wedge \mathcal{K} \models F_i(y)) \\ u_i \text{ otherwise} \end{cases} \quad (1)$$

where $x, y \in Ind_C(\mathcal{K})$ and $p > 0$.



(a) 3-NN graph, BioPAX (Proteomics)

(b) 3-NN graph, Leo

Fig. 1: k-Nearest Neighbor Semantic Similarity graphs for individuals BioPAX (Proteomics) ontology (left) and for the Leo ontology (right), obtained using the dissimilarity measure in [25]: F was defined as the set of atomic concepts in the ontology (each weighted with its normalized entropy [25]) and $p = 2$.

Two examples of (k-NN) semantic similarity graphs among all individuals in the ontologies BioPax (Proteomics) and Leo, obtained using the aforementioned dissimilarity measure, are provided in Fig. 1.

3.2 Quadratic Cost Criteria

In quadratic cost criteria [6, ch. 11], the original label space $\{-1, +1\}$ (binary classification case) is relaxed to $[-1, +1]$. This allows expressing the confidence associated to a labeling (and possibly provide an indicator of $P(Y \mid x)$). For such a reason, in the proposed method, elements of the functions space $\mathcal{F}$ can be relaxed to the form $f : \mathrm{Ind}_C(\mathcal{K}) \mapsto [-1, +1]$.

As in subsection 2.3, labeling functions can be equivalently represented as vectors $\mathbf{y} \in [-1, +1]^n$. Let $\hat{\mathbf{y}} \in [-1, +1]^n$ be a possible labeling for a set of n instances. We can see $\hat{\mathbf{y}}$ as a $(l + u) = n$ dimensional vector, where the first l indexes refer to already labeled instances, and the last u to unlabeled instances: $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_l, \hat{\mathbf{y}}_u]$.

Consistency of $\hat{\mathbf{y}}$ with respect to original labels can be formulated in the form of a quadratic cost: $\sum_{i=1}^{l} (\hat{y_i} - y_i)^2 = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2$.

To regularize the labellings with respect to the graph structure, the *graph Laplacian* [6] can be used. Let $\mathbf{W}$ be the adjacency (weight) matrix corresponding to the similarity graph G and let $\mathbf{D}$ be the diagonal matrix obtained from $\mathbf{W}$ as $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{W}_{ij}$ (i.e. by summing the elements in each column of $\mathbf{W}$).

Hence, two alternative definitions for the graph Laplacian can be considered [6]:

Unnormalized graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
Normalized graph Laplacian: $\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$.

Another regularization term in the form of $||\hat{\mathbf{y}}||^2$ (or $||\hat{\mathbf{y}}_u||^2$, as in [33]) can be added to the final cost function to prefer smaller values in $\hat{\mathbf{y}}$. This is useful e.g. to prevent arbitrary labellings in a connected component of the semantic similarity graph containing no labeled instances.

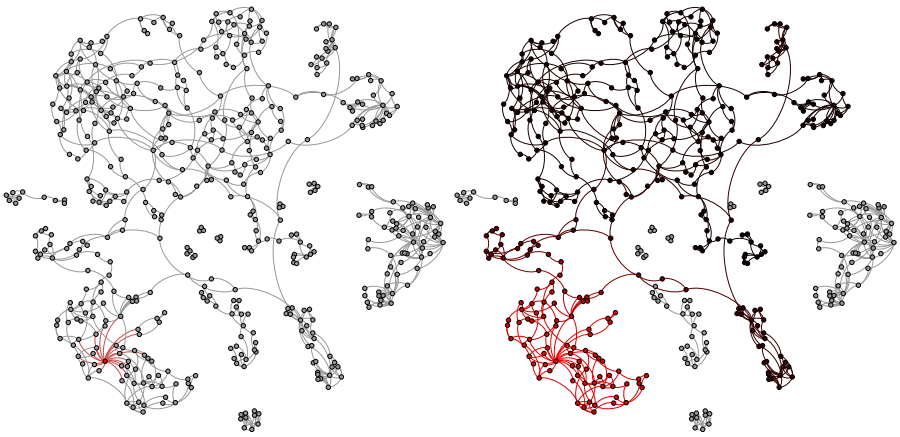
Putting the pieces together, we obtain two quadratic cost criteria discussed in the literature, namely Regression on Graph [2] (RG) and the Consistency Method [33] (CM):

Regression on Graph where the cost function can be written as:

$$cost(\hat{\mathbf{y}}) = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2 + \mu \hat{\mathbf{y}}^T \mathbf{L} \hat{\mathbf{y}} + \mu\epsilon ||\hat{\mathbf{y}}||^2; \tag{2}$$

Consistency Method where the cost function can be written as:

$$cost(\hat{\mathbf{y}}) = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2 + \mu \hat{\mathbf{y}}^T \mathcal{L} \hat{\mathbf{y}} + ||\hat{\mathbf{y}}_u||^2. \tag{3}$$



(a) Before propagation (b) After propagation

Fig. 2: Example of information propagation, from a single individual, to nearby individuals in a sample similarity graph.

We will now derive a closed form solution for the problem of finding a (global) minimum for the quadratic cost criterion in RG; a similar process is also valid in the case of CM. Its first order derivative is defined as follows:

$$\frac{1}{2} \frac{\partial cost(\hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}} = (\mathbf{S} + \mu \mathbf{L} + \mu\epsilon \mathbf{I})\hat{\mathbf{y}} - \mathbf{S}\mathbf{y},$$

where $\mathbf{S} = diag(\mathbf{s}_1, \ldots, \mathbf{s}_n)$, with $\mathbf{s}_i = 1$ iff $i \leq l$ and 0 otherwise. Its second order derivative is a positive definite matrix if $\epsilon > 0$, since $\mathbf{L}$ is positive semi-definite. Therefore, setting the first order derivative to 0 leads to a global minimum:

$$\hat{\mathbf{y}} = (\mathbf{S} + \mu\mathbf{L} + \mu\epsilon\mathbf{I})^{-1}\mathbf{S}\mathbf{y}, \tag{4}$$

showing that $\hat{\mathbf{y}}$ can be obtained either by matrix inversion or by solving a (possibly sparse) linear system.

Complexity of Inference The linear system in Eq. 4 can be computed efficiently, with a nearly-linear time complexity in the number of non-zero elements in the coefficient matrix. Indeed, computing $\hat{\mathbf{y}}$ can be reduced to solving a linear system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} = (\mathbf{S} + \mu\mathbf{L} + \mu\epsilon\mathbf{I})$, $\mathbf{b} = \mathbf{S}\mathbf{y}$ and $\mathbf{x} = \hat{\mathbf{y}}$. A linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}^{n\times n}$ can be solved in nearly linear time if the coefficient matrix $\mathbf{A}$ is *symmetric diagonally dominant*[1] (SDD). An algorithm for solving SDD linear systems is proposed in [7]: its time-complexity is $\approx \mathrm{O}\big(m \log^{1/2} n\big)$, where m is the number of non-zero entries in $\mathbf{A}$ and n is the number of variables in the system of linear equations. This result applies to the calculation in Eq. 4, since the graph Laplacian $\mathbf{L}$ is SDD [30], and thus the coefficient matrix $\mathbf{A}$ is SDD. An efficient parallel solver for SDD linear systems is proposed in [24].

Interpretation as a Probabilistic Graphical Model The terms enforcing similar labels among nearby individuals and the regularizer in the cost functions in Eq. 2 and Eq. 3 can be seen as *energy functions* [20] over $\hat{\mathbf{y}}$ in the form:

$$E(\hat{\mathbf{y}}) = \hat{\mathbf{y}}^T\tilde{\mathbf{L}}\hat{\mathbf{y}}, \text{ with } \hat{\mathbf{y}} \in \mathbb{R}^n, \tag{5}$$

where $\tilde{\mathbf{L}} = \mu(\mathbf{L} + \epsilon\mathbf{I})$ in Eq. 2 and $\tilde{\mathbf{L}} = \mu\mathcal{L} + \mathbf{I}$ in Eq. 3. The energy function in Eq. 5 corresponds to a *Gaussian Random Field* [20] (GRF):

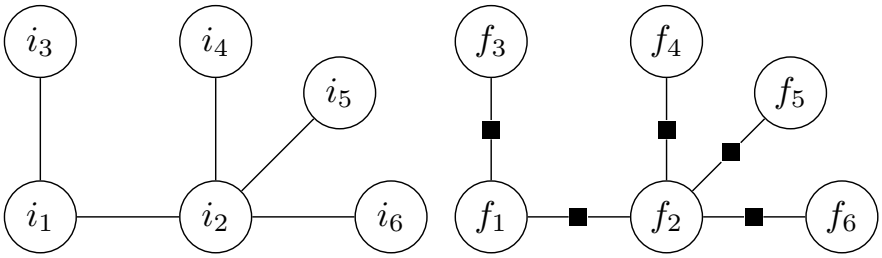
$$p(\hat{\mathbf{y}}) = \frac{1}{Z}\exp\Big[-\beta E(\hat{\mathbf{y}})\Big] = \frac{1}{Z}\exp\Big[-\beta\hat{\mathbf{y}}^T\tilde{\mathbf{L}}\hat{\mathbf{y}}\Big], \tag{6}$$

where Z is a normalization factor and β is an "inverse temperature parameter". The GRF in Eq. 6 defines a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ on the continuous labellings $\hat{\mathbf{y}}$, where $\boldsymbol{\Omega} = (2\beta\tilde{\mathbf{L}})$ and $\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1}$ represent respectively its *information* (or *precision*) and *covariance* matrix. Such matrices encode the independence relations among variables in the multivariate Gaussian distribution.

Given that $\hat{\mathbf{y}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\hat{\mathbf{y}}_i$ and $\hat{\mathbf{y}}_j$ are independent iff $\boldsymbol{\Sigma}_{ij} = 0$ (i.e. $\hat{\mathbf{y}}_i \perp \perp \hat{\mathbf{y}}_j$ iff $\boldsymbol{\Sigma}_i j = 0$), while $\hat{\mathbf{y}}_i$ and $\hat{\mathbf{y}}_j$ are independent conditioned on all the other variables iff $\boldsymbol{\Omega}_{ij} = 0$ (i.e. $\hat{\mathbf{y}}_i \perp\!\!\!\perp \hat{\mathbf{y}}_j \mid \hat{\mathbf{y}} - \{\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_j\}$ iff $\boldsymbol{\Omega}_{ij} = 0$).

It is interesting to note that the information matrix $\boldsymbol{\Omega}$ (and hence the graph Laplacian of the similarity matrix) directly defines a minimal I-map Gaussian Markov random field (GMRF) for the distribution p [20], where non-zero entries in the matrix can be directly translated to edges in the GMRF.

[1] A matrix $\mathbf{A}$ is SDD iff $\mathbf{A}$ is symmetric (i.e. $\mathbf{A} = \mathbf{A}^T$) and $\forall i : \mathbf{A}_{ii} \geq \sum_{i\neq j} |\mathbf{A}_{ij}|$.



(a) Sample Similarity Graph among a set of 6 individuals in a Knowledge Base

(b) Corresponding GMRF defined over the soft-labels of the individuals

Summary This work leverages quadratic cost criteria to efficiently solve the transductive class-membership prediction problem. Finding a minimum $\hat{\mathbf{y}}$ for a predefined cost criterion is equivalent to finding a labeling function f^* in the form $f^* : \mathrm{Ind}_C(\mathcal{K}) \mapsto [-1, +1]$, where the labeling returned for a generic training individual $a \in Ind_C(\mathcal{K})$ correspond to the value in $\hat{\mathbf{y}}$ in the position mapped to a. This can be done by representing the set of training individuals $Ind_C(\mathcal{K})$ as a partially labeled vector $\mathbf{y}$ of length $|Ind_C(\mathcal{K})| = n$, such that the first l (resp. last u) components correspond to positive and negative (resp. neutral) examples in $Ind_C(\mathcal{K})$. Such $\mathbf{y}$ can be then used to measure the consistency with original labels in a quadratic cost criterion; while the semantic similarity graph can be employed to enforce smoothness in class-membership predictions among similar training individuals. An advantage of quadratic cost criteria is that their minimization ultimately reduces to solving a large sparse linear system with a SDD coefficient matrix. For large-scale datasets, a subset selection method is discussed in [6, ch. 18], which allows to greatly reduce the size of the original linear system.

4 Empirical Evaluations

In this section, we evaluate several (inductive and transductive) methods for class-membership prediction, with the aim of comparing the methods discussed in section 3 with respect to other methods in SW literature. We are reporting evaluations for the Regularization on Graph [2] (RG) and the Consistency Method [33] (CM); Label Propagation [34] (LP); three kinds of Support Vector Machines [27] (SVM), namely Hard-Margin SVM (HM-SVM), Soft-Margin SVM with L_1 norm (SM-SVM) and Laplacian SVM [3] (LapSVM); and $\sqrt{l}$-Nearest Neighbors for class-membership prediction [25].

4.1 Description of Evaluated Methods

LP is a graph-based SSL algorithm relying on the idea of propagating labeling information among similar instances through an iterative process involving matrix operations. It can be equivalently formulated under the quadratic criterion

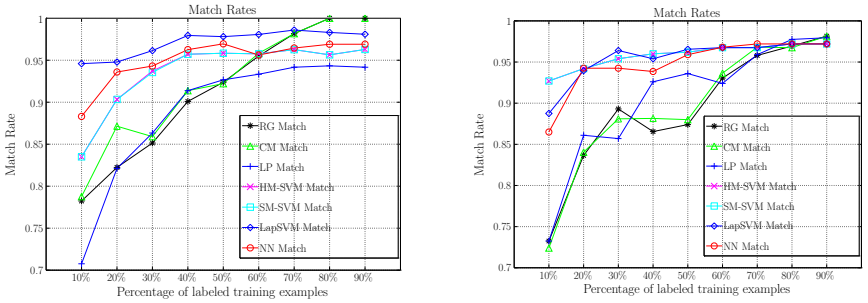
framework [6, ch. 11]. More formally it associates, to each unlabeled instance in the graph, the probability of performing a random walk until a positively (resp. negatively) example is found.

We also evaluated Support Vector Machines (SVM), which have been proposed for inducing robust classifiers from ontological knowledge bases [13, 25]. SVM classifiers come in different flavors: the classic HM-SVM binary classifier aims at finding the hyperplane in the feature space separating the instances belonging to different classes, which maximizes the *geometric margin* between the hyperplane and nearest training points. The SM-SVM classifier is a relaxation of HM-SVM, which allows for some misclassification in training instances (by relaxing the need of having perfectly linearly separable training instances in the feature space). LapSVM is a semi-supervised extension of the SM-SVM classifier: given a set of labeled instances and a set of unlabeled instances, it aims at finding an hyperplane that is also smooth with respect to the (estimated) geometry of instances. More formally, let $(\mathbf{x}_l, \mathbf{y}_l)$ (resp. $\mathbf{x}_u$) be a set of labeled (resp. unlabeled) instances. LapSVM finds a function f in a space of functions $\mathcal{H}_K$ determined by the kernel K (called *Reproducing Kernel Hilbert Space* [27]) minimizing $\frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_L ||f||^2_{\mathcal{H}_K} + \gamma_{\mathcal{M}} ||f||^2_{\mathcal{M}}$, where V represents a costs function of errors committed by f on labeled samples (typically the hinge loss function $\max\{0, 1 - y_i f(x_i)\}$), $||\cdot||_{\mathcal{H}_K}$ imposes smoothness conditions on possible solutions [27] and $||\cdot||^2_{\mathcal{M}}$, intuitively, penalizes rapid changes in the classification function between close instances in the similarity graph. It generalizes HM-SVM ($\gamma_L \to 0, \gamma_{\mathcal{M}} = 0$) and SM-SVM ($\gamma_{\mathcal{M}} = 0$). Our implementation of LapSVM follows the algorithm proposed in [3]; for HM-SVM, SM-SVM and LapSVM, we solve the underlying convex optimization problems using the Gurobi optimizer [15].

RG, CM, LP and LapSVM all rely on a semantic similarity graph $\mathbf{W}$ as a representation of the geometry of instances. We first calculate distances employing the dissimilarity measure defined in [25] and outlined in eq. 1, with $p = 2$; then we obtain $\mathbf{W}$ by building a k-Nearest Neighbor graph using such distances (since sparsity in $\mathbf{W}$ influences the scalability of quadratic cost criteria, as written in subsection 3.2). When building the neighborhood of a node, we handled the cases in which nodes had the same distance by introducing a random ordering between such nodes. The Kernel function used for Hard-Margin SVM, Soft-Margin SVM and Laplacian SVM are also defined in [25], and directly correlated with the aforementioned dissimilarity measure in eq. 1 (given a committee of concepts F and the parameters $\mathbf{w}$ and p, the dissimilarity was originally obtained as $1 - k(a, b)$, where $k(a, b)$ is the value of the kernel function on a pair of individuals (a, b) in the knowledge base). We also provide a first evaluation for the k-NN algorithm (with $k = \sqrt{l}$, where l is the number of labeled instances, as discussed in [25]): we simply choose the majority class among the $\sqrt{l}$ most similar individuals to label each unlabeled instance.

Ontology	Expressivity	#Axioms	#Inds.	#Classes	#ObjProps.
BioPax (Proteomics)	$\mathcal{ALCHN}(\mathcal{D})$	773	49	55	47
Family-Tree	$\mathcal{SROIF}(\mathcal{D})$	2059	368	22	52
Leo	$\mathcal{ALCHIF}(\mathcal{D})$	430	61	32	26
MDM0.73	$\mathcal{ALCHOF}(\mathcal{D})$	1098	112	196	22
Wine	$\mathcal{SHOIN}(\mathcal{D})$	1046	218	142	21

Table 1: Ontologies considered in the experiments.



(a) Leo ontology

(b) BioPAX (Proteomics) ontology

Fig. 4: Variation of average Match Rates with respect to the number of folds used in the training step, during a k-Fold Cross Validation (with $k = 10$).

4.2 Evaluations

Starting from a set of real ontologies [2] (outlined in Table 1), we generated a set of 20 random query concepts for each ontology [3], so that the number of individuals belonging to the target query concept C (resp. $\neg C$) was at least of 10 elements and the number of individuals in C and $\neg C$ was in the same order of magnitude. A DL reasoner [4] was employed to decide on the theoretical concept-membership of individuals to query concepts. We employ the evaluation metrics in [8], which take into account the peculiarities deriving by the presence of missing knowledge:

Match Case of an individual that got the same label by the reasoner and the inductive classifier.

Omission Error Case of an individual for which the inductive method could not determine whether it was relevant to the query concept or not while it was found relevant by the reasoner.

Commission Error Case of an individual found to be relevant to the query concept while it logically belongs to its negation or vice-versa.

[2] From TONES Repository: `http://owl.cs.manchester.ac.uk/repository/`

[3] Using the methods available at `http://lacam.di.uniba.it/~nico/research/ontologymining.html`

[4] Pellet v2.3.0 – `http://clarkparsia.com/pellet/`

Leo	Match	Omission	Commission	Induction
RG	1 ± 0	0 ± 0	0 ± 0	0 ± 0
CM	1 ± 0	0 ± 0	0 ± 0	0 ± 0
LP	0.942 ± 0.099	0.007 ± 0.047	0.052 ± 0.091	0 ± 0
SM-SVM	0.963 ± 0.1	0 ± 0	0.037 ± 0.1	0 ± 0
LapSVM	0.978 ± 0.068	0 ± 0	0.022 ± 0.068	0 ± 0
$\sqrt{l}$-NN	0.971 ± 0.063	0 ± 0	0.029 ± 0.063	0 ± 0
BioPAX (Proteomics)	Match	Omission	Commission	Induction
RG	0.986 ± 0.051	0.004 ± 0.028	0.008 ± 0.039	0.002 ± 0.02
CM	0.986 ± 0.051	0.002 ± 0.02	0.01 ± 0.044	0.002 ± 0.02
LP	0.982 ± 0.058	0.002 ± 0.02	0.014 ± 0.051	0.002 ± 0.02
SM-SVM	0.972 ± 0.075	0 ± 0	0.026 ± 0.068	0.002 ± 0.02
LapSVM	0.972 ± 0.075	0 ± 0	0.026 ± 0.068	0.002 ± 0.02
$\sqrt{l}$-NN	0.972 ± 0.075	0 ± 0	0.026 ± 0.068	0.002 ± 0.02
MDM0.73	Match	Omission	Commission	Induction
RG	0.953 ± 0.063	0.003 ± 0.016	0.011 ± 0.032	0.015 ± 0.039
CM	0.953 ± 0.063	0.001 ± 0.009	0.013 ± 0.036	0.018 ± 0.04
LP	0.942 ± 0.065	0 ± 0	0.026 ± 0.046	0.033 ± 0.054
SM-SVM	0.793 ± 0.252	0 ± 0	0.174 ± 0.255	0.033 ± 0.054
LapSVM	0.915 ± 0.086	0 ± 0	0.052 ± 0.065	0.033 ± 0.054
$\sqrt{l}$-NN	0.944 ± 0.069	0 ± 0	0.023 ± 0.051	0.033 ± 0.054
Wine	Match	Omission	Commission	Induction
RG	0.24 ± 0.03	0 ± 0.005	0.007 ± 0.017	0.5 ± 0.176
CM	0.242 ± 0.028	0 ± 0.005	0.005 ± 0.015	0.326 ± 0.121
LP	0.239 ± 0.035	0 ± 0.005	0.008 ± 0.021	0.656 ± 0.142
SM-SVM	0.235 ± 0.036	0 ± 0	0.012 ± 0.024	0.753 ± 0.024
LapSVM	0.238 ± 0.033	0 ± 0	0.009 ± 0.021	0.753 ± 0.024
$\sqrt{l}$-NN	0.241 ± 0.031	0 ± 0	0.006 ± 0.018	0.753 ± 0.024

Table 2: Match, Omission, Commission and Induction [25] results for a k-Fold Cross Validation ($k = 10$) on 20 randomly generated queries. For each experiment, the best parameters within the training were found using a k-Fold Cross Validation ($k = 10$).

Induction Case of an individual found to be relevant to the query concept or to its negation, while either case is not logically derivable from the knowledge base.

Before evaluating on the test set, parameter tuning was performed for each of the methods via a k-Fold Cross Validation ($k = 10$) within the training set, for finding the parameters with lower classification error in cross-validation. For LapSVM, the $(\gamma_L, \gamma_{\mathcal{M}})$ parameters were varied in $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$, while for SM-SVM, which follows the implementation in [27, pg. 223], the C parameter was allowed to vary in $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$. Similarly, the (μ, ϵ) parameters in RG and CM where varied in $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$. The parameter k for building the k-NN semantic similarity graph, used by LapSVM, RG, CM and LP, was varied in $\{2, 4, 8, 16\}$. We did not carefully choose the concept committee F defining the dissimilarity measure: we simply used the set of atomic concepts in the ontology, thus ignoring any prior knowledge about the structure of the target concept C or the presence of statistical correlations in the knowledge base. Each concept in the committee F was weighted with its normalized entropy [25]. RG, CM and LP give an indication of the uncertainty associated to a specific labeling by associating values in the set $[-1, +1]$ to each node. A labelingx ≈ 0 (specifically, when the label was in the set $[-10^{-4}, 10^{-4}]$ we decided to leave the node unlabeled, so to try to provide more robust estimates of labels (and thus a possibly lower commission error and match rates and higher omission error rates). This may happen e.g. when there are no labeled examples within a connected component of the semantic similarity graph.

In Tab. 2 we report average index rates and standard deviations for each of the ontologies in Tab. 1; the only exceptions is for the FAMILY-TREE ontology, which provided 0.76 ± 0.13 match rates and 0.24 ± 0.13 induction rates for all methods (except for LP, where the induction rates were 0.21 ± 0.14. In general, LapSVM outperformed the other two non-SSL SVM classification methods. This happened with varying quantities of unlabeled data; this is shown for example in the behavior of match rates in Fig. 4a, where results obtained in a k-Fold Cross Validation using a varying quantity of labeled instances. However, standard SVM training is $O(m^3)$ in general, where m is the number of training instances; therefore, some extra effort may be necessary to make SVM methods scale on SW knowledge bases. Such results may provide some empirical evidence that inductive methods for formal ontologies may take benefit from also accounting for unlabeled instances during learning.

4.3 Limitations

A fundamental problem in graph-based SSL methods relies in the construction of the similarity graph [6, 35], which is known to have a strong impact on the effectiveness of SSL methods. In this work, we identified similarity relations among individuals using a measure defined in [25] together with a set of atomic concepts defined in the ontology. However, this might not always be effective

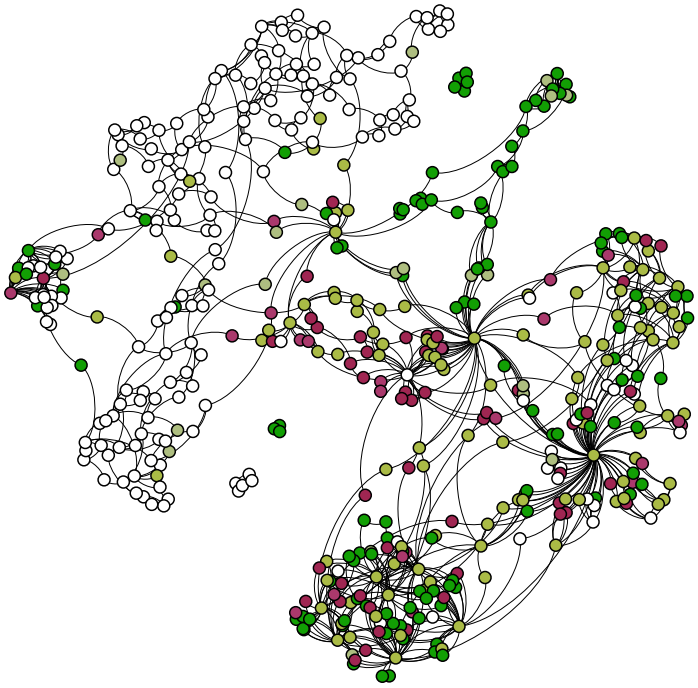


Fig. 5: Semantic Similarity Graph for the persons in the AIFB PORTAL ontology, after removing predicates encoding research group affiliations: each color corresponds to a distinct research group affiliation (white was used when no affiliation was available).

(consider e.g. a shallow ontology, where only a few properties of each individual are described by means of atomic concepts).

A possible approach to leverage the relational graph structure between individuals in a DL ontology, would be the use of *graph* and *RDF kernels*, such as the one defined in [4, 22, 28]. By implicitly mapping individuals into a feature space, a kernel function $k(\cdot, \cdot)$ naturally induces a Euclidean distance in the kernel feature space [27]:

$$||\phi(x_i) - \phi(x_j)||^2 = k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j),$$

where ϕ is a function mapping each instance to some feature space. However, this might not always be effective: Fig. 5 shows a Semantic Similarity Graph constructed among persons in the AIFB PORTAL Ontology [5], using the RDF kernel described in [22] (ignoring research group affiliations), where each different colors corresponds to a distinct research group. Similarity relations among individuals inferred by such a kernel do not accurately reflect similarities in re-

[5] `http://www.aifb.kit.edu/web/Wissensmanagement/Portal`, as of 21 Feb. 2012

search group affiliations, suggesting that the choice of a kernel could be task dependent in some contexts.

5 Conclusion and Future Works

This work proposes a method for transductive class-membership prediction based on graph-based regularization from DL representations. It leverages neutral examples by propagating class-membership information among similar individuals in the training set. The proposed method relies on quadratic cost criteria, whose optimization can be reduced to solving a (possibly sparse) symmetric and diagonally dominant linear system. This is a well-known problem in the literature, with a nearly linear time complexity in the number of non-zero entries in the coefficient matrix.

The similarity graph is known to have a strong influence on the effectiveness of graph-based SSL methods [35], suggesting that the graph construction process might be guided by the prediction task at hand. The construction of the similarity graph for class-membership learning tasks can be influenced by factors such as the structure of the target concept C, or by finding statistical correlation within the knowledge base. Also, it is not clear whether continuous labels assigned by the proposed methods may correspond to posterior probability estimates from the statistical point of view. In future work, we aim at investigating the aforementioned two aspects of graph-based transductive and semi-supervised class-membership prediction from DL representations.

References

[1] Alexandrescu, A., Kirchhoff, K.: Data-Driven Graph Construction for Semi-Supervised Graph-Based Learning in NLP. In: Sidner, C., et al. (eds.) HLT-NAACL. pp. 204–211. The Association for Computational Linguistics (2007)

[2] Belkin, M., Matveeva, I., Niyogi, P.: Regularization and Semi-supervised Learning on Large Graphs. In: Shawe-Taylor, J., et al. (eds.) COLT. LNCS, vol. 3120, pp. 624–638. Springer (2004)

[3] Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006)

[4] Bloehdorn, S., Sure, Y.: Kernel Methods for Mining Instance Data in Ontologies. In: Aberer, K., et al. (eds.) The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. LNCS, vol. 4825, pp. 58–71. Springer (2007)

[5] Borgida, A., Walsh, T., Hirsh, H.: Towards measuring similarity in description logics. In: Horrocks, I., et al. (eds.) Description Logics. CEUR Workshop Proceedings, vol. 147. CEUR-WS.org (2005)

[6] Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge, MA (2006)

[7] Cohen, M.B., Kyng, R., Miller, G.L., Pachocki, J.W., Peng, R., Rao, A., Xu, S.C.: Solving sdd linear systems in nearly $m\log^{1/2}n$ time. In: Shmoys [29], pp. 343–352

[8] d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: An inductive approach. In: Bechhofer, S., et al. (eds.) The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings. LNCS, vol. 5021, pp. 288–302. Springer (2008)
[9] d'Amato, C., Fanizzi, N., Esposito, F.: A Semantic Similarity Measure for Expressive Description Logics. CoRR abs/0911.5043 (2009)
[10] d'Amato, C., Staab, S., Fanizzi, N.: On the Influence of Description Logics Ontologies on Conceptual Similarity. In: Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns. pp. 48–63. EKAW '08, Springer, Berlin, Heidelberg (2008)
[11] Fanizzi, N., d'Amato, C.: Inductive concept retrieval and query answering with semantic knowledge bases through kernel methods. In: Proceedings of the 11th international conferenceon Knowledge-based intelligent information and engineering systems: Part I. pp. 148–155. KES'07, Springer (2007)
[12] Fanizzi, N., d'Amato, C., Esposito, F.: Reduce: A reduced coulomb energy network method for approximate classification. In: Proceedings of the 6th European Semantic Web Conference (ESWC'09). pp. 323–337. Springer
[13] Fanizzi, N., d'Amato, C., Esposito, F.: Statistical Learning for Inductive Query Answering on OWL Ontologies. In: Proceedings of the 7th International Conference on The Semantic Web. pp. 195–212. ISWC '08, Springer (2008)
[14] Fanizzi, N., d'Amato, C., Esposito, F.: Towards learning to rank in description logics. In: Coelho, H., et al. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 985–986. IOS Press (2010)
[15] Gurobi Optimization, I.: Gurobi optimizer reference manual (2012)
[16] Hu, B., Dasmahapatra, S., Lewis, P.: Semantic metrics. Int. J. Metadata Semant. Ontologies 2(4), 242–258 (Jul 2007)
[17] Janowicz, K., Wilkes, M.: SIM-DLA: A Novel Semantic Similarity Measure for Description Logics Reducing Inter-concept to Inter-instance Similarity. In: Aroyo, L., et al. (eds.) ESWC. LNCS, vol. 5554, pp. 353–367. Springer (2009)
[18] Kapoor, A., Qi, Y.A., Ahn, H., Picard, R.W.: Hyperparameter and Kernel Learning for Graph Based Semi-Supervised Classification. In: Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada] (2005)
[19] Kersting, K., Raedt, L.D.: Bayesian logic programming: Theory and tool. In: Getoor, L., Taskar, B. (eds.) An Introduction to Statistical Relational Learning. MIT Press (2007)
[20] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
[21] Lasserre, J., Bishop, C.M.: Generative or Discriminative? Getting the Best of Both Worlds. Bayesian Statistics 8, 3–24 (2007)
[22] Lösch, U., Bloehdorn, S., Rettinger, A.: Graph Kernels for RDF Data. In: Simperl, E., et al. (eds.) ESWC. LNCS, vol. 7295, pp. 134–148. Springer (2012)
[23] Ochoa-Luna, J.E., Cozman, F.G.: An algorithm for learning with probabilistic description logics. In: Bobillo, F., et al. (eds.) URSW. pp. 63–74 (2009)
[24] Peng, R., Spielman, D.A.: An efficient parallel solver for sdd linear systems. In: Shmoys [29], pp. 333–342
[25] Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the Semantic Web - Statistical Learning for Next Generation Knowledge Bases. Data Mining and Knowledge Discovery - Special Issue on Web Mining (2012)

[26] Rettinger, A., Nickles, M., Tresp, V.: Statistical Relational Learning with Formal Ontologies. In: Buntine, W.L., et al. (eds.) ECML/PKDD (2). LNCS, vol. 5782, pp. 286–301. Springer (2009)
[27] Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)
[28] Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman Graph Kernels. Journal of Machine Learning Research 12, 2539–2561 (2011)
[29] Shmoys, D.B. (ed.): Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014. ACM (2014)
[30] Spielman, D.A.: Algorithms, Graph Theory, and Linear Equations in Laplacian Matrices. In: Proceedings of the International Congress of Mathematicians 2010 (ICM 2010). pp. 2698–2722 (2010)
[31] Vapnik, V.N.: Statistical learning theory. Wiley, 1 edn. (Sep 1998)
[32] Zhang, X., Lee, W.S.: Hyperparameter Learning for Graph Based Semi-supervised Learning Algorithms. In: Schölkopf, B., et al. (eds.) Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006. pp. 1585–1592. MIT Press (2006)
[33] Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Thrun, S., et al. (eds.) Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]. MIT Press (2003)
[34] Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. rep., CMU CALD tech report CMU-CALD-02 (2002)
[35] Zhu, X.: Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison (2005)