Pattern Recognition Letters 37 (2014) 4–14

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Pattern classification and clustering: A review of partially supervised learning approaches

Friedhelm Schwenker [a,*], Edmondo Trentin [b]

[a] Institute of Neural Information Processing, Ulm University, 89069 Ulm, Germany
[b] Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Via Roma 56, 53100 Siena, Italy

ARTICLE INFO

Article history:
Available online 30 October 2013

Keywords:
Partially supervised learning
Semi-supervised learning
Active learning
Transductive learning
Multi-view learning
Neural network

ABSTRACT

The paper categorizes and reviews the state-of-the-art approaches to the partially supervised learning (PSL) task. Special emphasis is put on the fields of pattern recognition and clustering involving partially (or, weakly) labeled data sets. The major instances of PSL techniques are categorized into the following taxonomy: (i) active learning for training set design, where the learning algorithm has control over the training data; (ii) learning from fuzzy labels, whenever multiple and discordant human experts are involved in the (complex) data labeling process; (iii) semi-supervised learning (SSL) in pattern classification (further sorted out into: self-training, SSL with generative models, semi-supervised support vector machines; SSL with graphs); (iv) SSL in data clustering, using additional constraints to incorporate expert knowledge into the clustering process; (v) PSL in ensembles and learning by disagreement; (vi) PSL in artificial neural networks. In addition to providing the reader with the general background and categorization of the area, the paper aims at pointing out the main issues which are still open, motivating the ongoing investigations in PSL research.

1. Introduction

The development of robust pattern classifiers from a limited training set $\mathcal{T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of observations (i.e., feature vectors) $\mathbf{x}_i \in X$, represented in a proper feature space X, has long been one of the most relevant and challenging tasks in machine learning and statistical pattern recognition (Jain et al., 2000). *Supervised learning* and *unsupervised learning* are the two major directions of traditional machine learning.

In the supervised framework, any given generic observation (or, pattern) $\mathbf{x} \in \mathcal{T}$ is uniquely associated with a corresponding target label $y \in Y$. It is assumed that X is a real-valued vector space (i.e., $X \subseteq \mathbb{R}^d$), and that $Y = \{y_1, \ldots, y_L\}$ is the set of L (different) class labels reflecting the ground truth of the classification problem at hand. Intervention from human experts is needed in order to label the training set correctly. During an initial phase of data collection and data annotation, a supervised training set

$$\mathcal{S} = \{(\mathbf{x}_i, y_i) \,|\, \mathbf{x}_i \in \mathbb{R}^d, y_i \in Y,\ i = 1, \ldots, m\}$$

is thus prepared. It is assumed that the data in $\mathcal{S}$ are independently drawn from some (unknown, yet identical) probability distribution defined on $\mathbb{R}^d \times Y$ (i.i.d. assumption) (Bishop, 2006). Subsequently, $\mathcal{S}$ is fed into a pre-selected supervised learning algorithm aimed at training a classifier C, that is a mapping $C : \mathbb{R}^d \to Y$. This algorithm is expected to exploit the information encapsulated within both the feature vectors and the corresponding class labels. Besides the training algorithm, a hypothesis space has to be fixed, as well (e.g., the space of multivariate polynomials of maximal degree p, or the set of two-layer artificial neural networks with p logistic hidden neurons). The hypothesis space consists of all the potential candidate classifiers C which may be the eventual outcome of the computation of the learning algorithm on the training set (Alpaydin, 2010).

As we say, data annotation is an additional, expensive, and error-prone preparation process. Individual data have to be carefully inspected (by one, or even more domain experts) in order to pinpoint somewhat reliable class labels for the training patterns. Instances of the difficulties involved in the process are found in areas such as bioinformatics, speech processing, or affective computing, where the exact class labels may not even be explicitly observable. Although annotating data might be extremely difficult and time consuming (or, sometimes, even impossible), supervised learning is still far the most prominent branch of machine learning and pattern recognition.

In the unsupervised learning framework a variety of methods and algorithms can be found in the literature. Major instances are represented by data clustering, density estimation, and dimensionality reduction (just to mention a few). The goal of the learning process is usually defined through an objective function, where the learning schemes use the observations without prior knowledge of

* Corresponding author. Tel.: +49 731 502 4159; fax: +49 731 502 4156.
E-mail address: friedhelm.schwenker@uni-ulm.de (F. Schwenker).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.10.017

the class labels $y \in Y$. In a typical unsupervised learning scenario the training set is defined as

$$\mathcal{U} = \{\mathbf{u}_i \,|\, \mathbf{u}_i \in \mathbb{R}^d, i = 1, \ldots, M\}$$

where the data $\mathbf{u}_i$ are independently drawn from an identical probability distribution over $\mathbb{R}^d$. Clearly, the lack of any prior expert knowledge renders unsupervised learning a particularly complex machine learning/pattern recognition task (Jain, 2010). In particular, the absence of target class labels during the training phase prevents the machine from resulting in a (more or less reliable) classifier. Indeed, from a general standpoint the data set $\mathcal{U}$ could not even involve any classification task at all. All the learning algorithm can do is analyzing the data, in an attempt to capture either probabilistic (e.g., the probability density function) or geometric/topological (e.g., some distance/similarity measure, or a partitioning of the data into homogeneous clusters) information describing the nature of the data distribution.

Moving a step froward from the traditional learning frameworks, it is easy to see that a somewhat intermediate scenario occurs under all practical circumstances where a classification problem is faced relying on a data set $\mathcal{T}$ whose data are only partially labeled, such that $\mathcal{T} = \mathcal{S} \cup \mathcal{U}$ for a proper, labeled subset $\mathcal{S}$ and its unlabeled counterpart $\mathcal{U}$. While classic unsupervised techniques do not lead to any classifier C in this setup, practitioners can still rely on regular supervised classifiers trained over $\mathcal{S}$. Unfortunately, in so doing all the data in $\mathcal{U}$ would not be exploited, resulting in a waste of potentially useful additional information which could strengthen the very classifier. As a consequence, the framework of *partially supervised learning* (PSL) was introduced, having the form of a family of machine learning algorithms lying between supervised and unsupervised learning. Moreover, PSL can be seen as machine learning under weak supervision, for instance learning with a fuzzy teacher (or, with fuzzy rewards).

1.1. Prominent directions of PSL research

In practical PSL applications, after collecting the raw data, several questions arise concerning the following data processing steps:

1. How many data shall be labeled, and how do we select the (possibly small) subset of informative patterns that will be labeled?
2. How do we combine and exploit both labeled and unlabeled data within a unifying, effective training scheme?
3. How many human experts should be involved in the (robust) labeling process, and how will labels be represented in case some of the experts mutually disagree?
4. How can the machine deal with soft/fuzzy labels or multiple labels in a PSL scenario?

In an attempt to put forward plausible answers to these and further questions, several prominent directions of research have been developed so far by the community in the PSL area, including: *active learning*, general *semi-supervised learning* (SSL) (further classified into *semi-supervised classification* and *semi-supervised clustering*), SSL with graphs, PSL in ensembles and multiple classifier systems. Furthermore, a variety of PSL approaches have been investigated in the broad realms of artificial neural networks, deep learning architectures and support vector machines.

In active learning, also known as *selective sampling* or *instance selection*, it is assumed that the learning algorithm can select the most informative input training data from the pool of unlabeled examples, and a human expert is asked to add label information to the selected examples (Settles, 2009). Popular algorithms are *uncertainty sampling* and *query by committee* sampling. The former trains a single classifier and then query the unlabeled example on which the classifier is least confident (Lewis and Catlett, 1994); the latter constructs multiple classifiers and then queries the unlabeled example on which the classifiers disagree the most (Freund et al., 1997).

In semi-supervised learning the basic idea is to take advantage of unlabeled data during a supervised learning procedure (known as *semi-supervised classification*), or to incorporate some type of prior information of data points such as class labels, or constraints on pairs of patterns as "must-link" or "cannot-link" (know as *semi-supervised clustering*). In contrast to active learning, an annotator is not involved in the processing cycle. *Transductive learning* is a special case of semi-supervised classification introduced in Vapnik (1995), where the test data set is known in advance, and the goal is to optimize the classification performance on the test set itself. Recent research on SSL concentrates, in addition to semi-supervised classification (Blum and Mitchell, 1998; Nigam et al., 2000; Zhou and Li, 2005; Li and Zhou, 2007; Peng et al., 2009) and semi-supervised clustering (such as constrained and seeded k-means clustering) (Wagstaff et al., 2001; Basu et al., 2002, 2004; Chu et al., 2009; Soleymani Baghshah and Bagheri Shouraki, 2010), on semi-supervised dimensionality reduction (Zhou et al., 2007; Kalakech et al., 2011), semi-supervised non-negative matrix factorization (Lee et al., 2010), semi-supervised manifold regularization (Belkin et al., 2006), or semi-supervised regression (Zhou and Li, 2005).

Other relevant branches of PSL encompass investigations of SSL with generative models (Nigam et al., 2000; Nigam, 2001), SSL with graphs (Blum and Chawla, 2001; Zhou et al., 2004; Zhu et al., 2003; Kulis et al., 2009), multi-view learning (including co-training) (Blum and Mitchell, 1998), PSL in ensembles/multiple classifiers (including learning by disagreement) (Zhou and Li, 2010), and PSL in neural networks and kernel machines. All these research directions are surveyed in the following sections.

1.2. Organization of the paper

We made every effort in trying and categorizing the different approaches to PSL in a suitable taxonomy. This resulted in the following organization of the paper. Sections 2 reviews active learning, including uncertainty sampling and query by committee. Next, learning from a fuzzy teacher is introduced in Section 3, embracing (amongst others) fuzzy nearest prototype, fuzzy learning vector quantization, and fuzzy-input fuzzy-output support vector machines. Both active learning and fuzzy learning paradigms are basically supervised learning schemes. SSL for classification is then discussed in Section 4, according to the following sub-topics: self-training, SSL with generative models, semi-supervised support vector machines and transductive learning, and SSL with graphs. In Section 5, SSL is surveyed in the context of unsupervised cluster analysis (including must-link/cannot-link strategies). PSL in multiple classifier systems/ensembles is reviewed in Section 6 (covering, amongst others: query by committee, learning by disagreement, multi-view learning and co-training, democratic co-learning, tri-training, etc.), while Section 7 covers a number of partially supervised approaches to artificial neural networks (multilayer perceptrons, deep architectures, radial basis function networks, self-organizing maps, and ad hoc architectures). Finally, conclusions are drawn in Section 8.

2. Active learning

The key idea behind active learning is that the learning algorithm is allowed to build a labeled training set $\mathcal{S}^* \subset \mathcal{S}$ autonomously. Starting from a small subset of labeled data, say $\mathcal{S}^0 \subset \mathcal{S}$,

a classifier is first constructed. This classifier is then applied to a pool of unlabeled data, in order to come up with a hypothesis of their labels. Then, these pre-classified samples can be arranged according to a measure of informativeness. Basically, the samples with high informativeness values are given to a human expert (or, even a group of experts of the application domain) who is asked to provide the selected data with proper labels. The newly labeled data are then added to the training set and the classifier, in turn, is re-trained on this augmented training data set, and so on. For an excellent review on active learning the reader is referred to Settles' literature survey (Settles, 2009).

Active learning is roughly divided in sequential active learning and pool-based active learning. Sequential active learning, also known as stream-based selective sampling, refers to methods where a single unlabeled example is sampled from the prior distribution and a pre-trained classifier decides whether the example is informative and should be added to the training set or not. Pool-based active learning is applied in such applications in which the unlabeled data have been collected before the training is going to start, so usually a set of labeled training data $\mathcal{S}$ is available together with an unlabeled data set $\mathcal{U}$. Both approaches have been studied in many applications, for example in text classification (Tong and Koller, 2002), image classification (Tong and Chang, 2001), speech recognition (Hakkani-Tür et al., 2006), affective computing (Meudt and Schwenker, 2012; Zhang, 2013) and medial diagnosis (Haque et al., 2013). Remote sensing image classification is possibly the most popular pattern recognition application of active learning (Thiel et al., 2008; Tuia et al., 2011). Here, high-resolution sensors generate a huge amount of data in a large number of spectral bands with the goal to detect and classify objects on the Earth. Finding the ground truth of the data by visual inspection of the full scene is impossible, so usually a rather limited number of samples can be obtained, and the reduction of labeling costs is the major goal here.

The evaluation of the unlabeled data and finding the most informative sample are the most important steps in an active learning procedure. In Settles (2009) a distinction is made between *uncertainty sampling* and *query by committee*. In case of probabilistic binary classifiers, the most informative pattern $\mathbf{u} \in \mathcal{U}$ is defined to be the one whose output (as computed by the classifier) is the closest to $1/2$ (among all outputs yielded by the very classifier over each and every pattern in $\mathcal{U}$). In multi-class classification some more general measures must be applied, e.g. the least confident example might be selected through minimizing the classifier output $P(\hat{y}_1|\mathbf{u})$ over all $\mathbf{u} \in \mathcal{U}$. Here $\hat{y}_1$ is the class label with the highest classifier output. Let $\hat{y}_2$ be the runner up, i.e. the class label such that $P(\hat{y}_2|\mathbf{u})$ is the second highest output. Knowledge of the label for $\mathbf{u} \in \mathcal{U}$ which minimizes the difference $P(\hat{y}_1|\mathbf{u}) - P(\hat{y}_2|\mathbf{u})$ might be useful to build a better classifier. This variant of multi-class uncertainty sampling is called *margin sampling* (Scheffer et al., 2001). Even more general uncertainty coefficients are viable, such as *entropy index*, i.e. $-\sum_y P(y|\mathbf{u}) \log P(y|\mathbf{u})$ or *Gini-index*, i.e. $1 - \sum_y P(y|\mathbf{u})^2$. Empirical evaluations of these different measures show that the best criterion highly depends on the application at hand (Settles and Craven, 2008; Settles, 2009). In the crisp classifier domain active learning algorithms have been applied to a wide range of base classifier schemes, e.g. tree classifiers (Lewis and Catlett, 1994), nearest neighbors (Lindenbaum et al., 2004) and also support vector machines (Tong and Chang, 2001). In the decision tree classifier any leaf node represents a subset of the training set $\mathcal{S}$, consisting of diverse fractions of samples from the different classes. From these fractions, a probabilistic label can be derived. A similar idea can be applied to the nearest neighbor setting in which a soft label can be computed using the labels of the k nearest neighbors. In the support vector setting the distance of an instance to the decision hyperplane can computed through logistic regression (Platt et al., 1999; Thiel et al., 2007).

3. Learning with fuzzy labels

Under several real-world circumstances (e.g., medical diagnosis, biological phenomena, stress or emotion recognition), the ground truth (i.e., the exact notion of the specific classes involved, and the individual class any given training pattern actually belongs to) may not be clearly defined, even for human experts. Labeling in such scenarios is expensive and time consuming. Additionally, it is error-prone, and individual experts might disagree on the class label for a given input example. Therefore, the results of multi-expert annotation procedures are usually expressed in terms of "soft" or fuzzy labels for the training data set:

$$\mathcal{S} = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in \mathbb{R}^d,\ \mathbf{y}_i \in \Delta^L,\ i = 1, \ldots, m \right\}$$

where $\Delta^L = \{\mathbf{y} \in [0,1]^L \mid \sum_{j=1}^{L} y_j = 1\}$ and L is the number of classes. Here the elements of vector $\mathbf{y}_i \in \Delta^L$ can be seen as the class memberships. One approach to deal with fuzzy memberships is to assign a crisp class label to maximum membership following the winner-takes-it-all principle. The more challenging option, which we follow here, is to incorporate the fuzzy memberships into the learning phase in order to take advantage of the graded class memberships. Class memberships range from crisp labels (which can be seen as a strong supervised learning setting) to the uniform class membership distribution $y_j = 1/L$ for all possible classes (which can be considered as a special unsupervised scenario), and therefore, we consider learning with uncertain class labels, or with weak teaching signals, as a special type of PSL.

Regression-based classification approaches are able to handle fuzzy labeled data directly. Examples of this class are multivariate polynomials, multi-layer perceptrons and radial basis function neural networks (Poggio and Girosi, 1989; Schwenker et al., 2001). The goal is then to approximate the class memberships by solving a regression problem. For other classifiers the standard training algorithm must be enhanced to obtain a fuzzy-input fuzzy-output behavior. For instance, the k-nearest neighbor classifier can easily be extended to show this fuzzy-input fuzzy-output behavior. Given a training set $\mathcal{S}$ as defined above and a data point $\mathbf{x}$ to be classified, the soft labels can be easily aggregated into a single fuzzy class membership, for example by a weighted average of the class memberships. Weighting the soft decisions is performed through incorporating the distance of the data point to the prototype vector. Subsequently a crisp label may be derived through maximum detection (Gayar et al., 2006).

Related approaches to classification are the *soft nearest prototypes* (Seo et al., 2003) and its extension (Villmann et al., 2005), where prototypes also are assigned fuzzy labels in the training process but the training data are still crisp labeled. A more general approach is the *fuzzy learning vector quantization* proposed in Thiel et al. (2008). Starting form the well-known, standard learning vector quantization (LVQ) approach and enhance it with the ability to work with soft labels, both as training data and for the prototypes some variants of fuzzy LVQ exist, for example those of Karayiannis and Bezdek (1997), that have crisp labels of the prototypes, and a soft neighborhood function.

In Thiel et al. (2007) a fuzzy-input fuzzy-output SVM (F^2SVM) is introduced where fuzzy class memberships are used during the training phase, and a fuzzy output is generated by using a logistic transfer function. For a two class classification problem, for instance, the class memberships y_i^+ and y_i^- for a data point $\mathbf{x}_i$ are incorporated in the SVM training by

$$\Theta(\mathbf{w}, \xi) := \frac{\mathbf{w}^T\mathbf{w}}{2} + C \sum_{i=1}^{m} (\xi_i^+ y_i^+ + \xi_i^- y_i^-)$$

with slack variables $\xi_i^- \geqslant 0,\ \xi_i^+ \geqslant 0,\ i=1,\ldots,m,\ C>0$ and constraints

$$\mathbf{w}^T\Phi(\mathbf{x}_i)+w_0 \geqslant 1-\xi_i^+ \quad \text{and} \quad \mathbf{w}^T\Phi(\mathbf{x}_i)+w_0 \leqslant -1+\xi_i^- \quad i=1,\ldots,m.$$

In primal form the optimization problem is given by

$$L_P(\mathbf{w},w_0,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r}) = \Theta(\mathbf{w},\boldsymbol{\xi}) - \sum_{i=1}^m r_i^+\xi_i^+ - \sum_{i=1}^m r_i^-\xi_i^- - \sum_{i=1}^m \alpha_i^+(\mathbf{w}^T\Phi(\mathbf{x}_i) + w_0 - 1 + \xi_i^+) + \sum_{i=1}^m \alpha_i^-(\mathbf{w}^T\Phi(\mathbf{x}_i)+w_0+1-\xi_i^-)$$

with non-negative Lagrange multipliers $r_i^+,\ r_i^-,\ \alpha_i^+,\ \alpha_i^-$, and in dual form

$$W(\alpha) = \sum_{i=1}^m \alpha_i^+ + \sum_{i=1}^m \alpha_i^- - \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m (\alpha_i^+-\alpha_i^-)(\alpha_j^+-\alpha_j^-)\mathbf{x}_i^T\mathbf{x}_j$$

subject to

$$\sum_{i=1}^m (\alpha_i^+-\alpha_i^-)=0, \quad \text{and} \quad 0\leqslant\alpha_i^+\leqslant Cy_i^+, \quad 0\leqslant\alpha_i^-\leqslant Cy_i^-, \quad i=1,\ldots,m.$$

The decision function is then given by

$$f(\mathbf{x}) = \operatorname{sign}\left(\underbrace{\textstyle\sum_{i=1}^m(\alpha_i^+-\alpha_i^-)\mathbf{x}^T\mathbf{x}_i+w_0}_{=:d_{\mathbf{x}}}\right)$$

and a fuzzified output can be achieved by means of a sigmoidal function

$$o_{a,b}(\mathbf{x}) = \frac{1}{1+\exp(-ad_{\mathbf{x}}+b)}.$$

Here, the parameters $a,b\in\mathbb{R}$ can be computed by optimizing error functions such as $E(a,b)=\sum_{i=1}^m(o_{a,b}(\mathbf{x}_i)-y_i)^2$ or $E(a,b)=\sum_{i=1}^m|o_{a,b}(\mathbf{x}_i)-y_i|$ on the training data $\mathcal{S}$. To keep the decision boundary fixed one could further assume $b=0$ (see Platt et al., 1999).

In Scherer et al. (2013) the F^2SVM is applied in speech processing for classification of voice quality characteristics. Voice quality refers to the timbre or coloring of a speaker's voice, this is a typical pattern recognition task in affective computing where fuzzy labels appear. It was demonstrated that F^2SVM has the potential to outperform standard classifiers.

4. Semi-supervised learning in classification

In the previous two sections we assumed that every pattern $\mathbf{x}\in X$ can be annotated by means of some crisp label $\mathbf{y}\in Y=\{y_1,\ldots,y_L\}$, or by a fuzzy membership vector $y\in\Delta^L, L$ being the number of classes. In semi-supervised classification the training set is given in two parts, a data set

$$\mathcal{S}=\{(\mathbf{x}_i,y_i)\,|\,\mathbf{x}_i\in\mathbb{R}^d, y_i\in Y, 1\leqslant i\leqslant m\}$$

annotated usually with crisp labels, and a set of unlabeled data

$$\mathcal{U}=\{\mathbf{u}_i\in\mathbb{R}^d\,|\,i=1,\ldots,M\}.$$

In principle this scenario could be considered as supervised when using the labeled data set $\mathcal{S}$ to build a classifier in the traditional supervised manner. On the other hand, unsupervised estimation of the probability density function $p(\mathbf{x})$ of the input data can take advantage from both $\mathcal{U}$ and $\mathcal{S}$. Thus, in situations where knowledge about $p(\mathbf{x})$ is useful to estimate $p(y|\mathbf{x})$, SSL as a combination of supervised and unsupervised learning might be useful in classification. In order to characterize such situations, some sort of smoothness assumptions on the data are required: (i) the "cluster assumption" (Patra and Bruzzone, 2012) says that data points in the same cluster are likely to belong to the same class; (ii) the "manifold assumption" (Song et al., 2008) says that a high-dimensional data sets can be embedded into a lower dimensional manifold (Chapelle et al., 2010). Such assumptions are far from being unusual in machine learning, for instance, the smoothness of a classifier revolves around the notion that if two input data points $\mathbf{x}_1,\ \mathbf{x}_2$ are close to each other, then the corresponding classifier outputs $y_1,\ y_2$ are mutually close, as well.

Many semi-supervised classification algorithms have been developed during the last 20 years. Following Zhu's literature survey (Zhu, 2008) SSL is structured in the following categories: (1) self-training; (2) SSL with generative models; (3) semi-supervised support vector machines (S^3VM), or transductive SVM; (4) SSL with graphs; (5) SSL with committees (or, SSL by disagreement). Except for the latter, which is reviewed in-depth in Section 6 (together with active learning approaches for ensembles), the former four topics of the list are surveyed in the following sub-sections, in the order.

4.1. Self-training

Self-training (Seeger, 2002; Nagy and Shelton, 1966) is an incremental algorithm to train a single classifier on $\mathcal{S}$ and $\mathcal{U}$ (Nigam and Ghani, 2000; Li et al., 2008; Frinken et al., 2009). The pre-trained base classifier is applied to the samples $\mathbf{u}\in\mathcal{U}$. Examples are ranked by some confidence measure and the most confident ones are added to the training set $\mathcal{T}$ in order to re-train the classifier by that augmented training data set. This process is repeated until some convergence criterion is fulfilled. Self-training is also known as self-corrective recognition (Nagy and Shelton, 1966), naïve labeling (Inoue and Ueda, 2003), or decision-directed estimation (Young and Farjo, 1972). It is a wrapper algorithm that, in principle, can be applied to any supervised learning algorithm. When self-training is applied to linear classifiers, such as support vector machines, the confident examples might not be informative because many of the confident instances have large distances from the decision boundary. Thus, in general the procedure does not select training sets comprising of informative examples.

4.2. Semi-supervised learning with generative models

Generative models have been used in statistical applications for many years (Miller and Uyar, 1997; Nigam et al., 2000; Shahshahani and Landgrebe, 1994). Here it is assumed that both labeled and unlabeled examples come from the same parametric model. Once the model parameters are computed, unlabeled examples are classified using the mixture components associated with each class. Such algorithms (see, for instance, Nigam et al., 2000; Nigam, 2001) usually treat the class labels of the unlabeled data $\mathcal{U}$ as missing values and employ the *expectation–maximization* (EM) algorithm to conduct maximum likelihood estimation of the model parameters (Dempster et al., 1977). Initially the model is trained on the labeled examples $\mathcal{S}$. Then, the current model is used to estimate the class probabilities of all the unlabeled examples, and eventually to compute a new model from all labeled examples. These approaches differ according to the generative models used to fit the data, e.g. mixture of Gaussian distributions (GMM) in image classification (Shahshahani and Landgrebe, 1994), mixture of multinomial distributions (Naive Bayes) in text categorization (Nigam et al., 2000; Nigam, 2001), Hidden Markov Models (HMM) in speech recognition (Inoue and Ueda, 2003). Generative models may be more accurate than discriminative models when the labeled training set is small. Yet, performance degradation

can be observed when the model assumption is incorrect (Cozman and Cohen, 2002; Zhu, 2008; Singh et al., 2008).

4.3. Semi-supervised support vector machines and transductive SVM

Semi-supervised SVM (S^3VM) are a special case of semi-supervised classifiers (Joachims, 1999; Grandvalet and Bengio, 2005; Chapelle et al., 2008). Basically, the idea of S^3VM learning is to exploit the unlabeled data $\mathcal{U}$ to move the decision boundary into a low density regions while keeping the labeled examples correctly classified (Joachims, 1999; Chapelle and Zien, 2005), and the following optimization problem is solved

$$\min_{w,b,\hat{y}} \Theta(w,b,\hat{y}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i)) + C^*\sum_{j=1}^{M} V(\hat{y}_j, f(\mathbf{x}_j)), \quad (1)$$

where $\hat{y}_i \in \{-1,1\}$, $i = 1,\ldots,M$ are the labels for the unlabeled data set $\mathcal{U}$ and $f(\mathbf{x}_i)$ defines the decision hyperplane of the *SVM*. Furthermore, V is some margin loss function, e.g. the *hinge loss* defined by

$$V(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))^p$$

with $p = 1$ or $p = 2$. The first two terms of Eq. (1) define standard *SVM* functional while the third sum incorporates the unlabeled data. Parameters C and C^* balance the loss between labeled and unlabeled data. A variety of different optimization techniques have been studied in the literature, e.g. local combinatorial search (Joachims, 1999), gradient descent (Chapelle and Zien, 2005), continuation techniques (Chapelle et al., 2006), convex–concave procedures (Fung and Mangasarian, 2001), semi-definite programming (Bie et al., 2006), deterministic annealing (Sindhwani et al., 2006), genetic algorithm optimization (Adankon and Cheriet, 2010) and branch-and-bound algorithms (Chapelle et al., 2006).

4.4. Semi-supervised learning with graphs

SSL with Graphs has been widely investigated so far, see for instance (Blum and Chawla, 2001; Zhu et al., 2003; Kulis et al., 2009). The basic idea is that the nodes of the graph are the examples of the data sets $\mathcal{S}$ and $\mathcal{U}$, and the edges between nodes show the similarities or dissimilarities between the nodes. For instance, Blum and Chawla (2001) constructed a graph whose nodes represent both labeled and unlabeled training examples and the edges between nodes are weighted according to the similarity between the corresponding examples. The aim is to find the minimum cut of the graph such that nodes in each connected component have the same label. Zhu et al. (2003) introduced a continuous prediction function. They modeled the distribution of the prediction function over the graph with Gaussian random fields and analytically proved that the prediction function with the lowest energy should have the harmonic property. They designed a label propagation strategy over the graph using such a harmonic property where the labels propagate from the labeled nodes to the unlabeled ones. It is worth noting that all graph-based methods assume that examples connected by strongly weighted edges tend to have the same class label and vice versa (Zhu, 2008).

5. Semi-supervised learning in clustering

Clustering is probably the most important task in the unsupervised learning scenario. Here the training data are given as a finite data set $\mathcal{U}$ without any additional prior information. The major goal of clustering is to group the data vectors in k sets, with $1 < k < M$. Data clustering relies on some kind of similarity (or, distance) measure, in combination with an overall objective function. In case of real-valued vectors, a distance between $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U}$ is given by $d(\mathbf{u}_1, \mathbf{u}_2) = \|\mathbf{u}_1 - \mathbf{u}_2\|$ where $\|\cdot\|$ is a suitable norm on $\mathbb{R}^d$. A number of different clustering techniques have been proposed throughout the last decades, such as partitioning clustering, hierarchical clustering, fuzzy clustering, or spectral clustering, to mention a few. They realize different objective functions, or heuristics. None of them requires any form of prior knowledge on the probabilistic laws underlying the data distribution. An excellent survey on cluster analysis can be found in Jain (2010).

Now, the general idea of semi-supervised clustering is to integrate some type of prior knowledge into the clustering process. For instance, a subset of labeled data may be provided in addition to the unlabeled data set. From this background knowledge further constraints on pairs of patterns in form of *must-link* or *cannot-link*, may be derived (or, explicitly defined) on the training data set (Wagstaff et al., 2001; Wagstaff, 2010).

Basu et al. (2004) introduced a probabilistic model for semi-supervised clustering algorithm based on hidden Markov random fields (HMRFs), where must-link or cannot-link constraints are integrated into an overall error function. The proposed HMRF k-means algorithm performs a prototype-based semi-supervised clustering algorithm. The authors discussed adaptive versions of the well-known cosine similarity and Kullback–Leibler divergence as distortion measures, and define the training algorithm in the EM-framework. In Kulis et al. (2009) kernel k-means is applied to clustering the data, and a constraint matrix is incorporated. with positive values for must-links and negative values for cannot-links. While this enables to cluster data in feature space, the values in the constraint matrix must be tuned for each kernel, and the resulting kernel matrix must be tweaked to be positive definite.

In Faußer and Schwenker (2012), the semi-supervised kernel clustering with sample-to-cluster weights (SKC) algorithm is introduced. In these weights, patterns whose labels match with the cluster label get a higher weight than patterns with non-matching labels. Here, the weights are less sensitive to the chosen kernel function. Basu's doctoral thesis (Basu, 2005) provides an excellent overview over various theoretical and practical aspects of semi-supervised clustering, furthermore several algorithms for semi-supervised clustering with labels, semi-supervised clustering with constraints, and active learning for constraint acquisition are considered.

6. Partially supervised learning in ensembles

Another field of research is PSL in combination with *classifier ensembles*, *multi classifier systems*, or *committees of classifiers*. In ensemble learning, a set of classifiers is trained on the labeled training data set $\mathcal{S}$. A combined decision rule is built upon the individual classifiers, expectedly exploiting their strengths and mutually overcoming their limitations. The main factor for the success of any committee-based SSL, often called SSL by disagreement, is to construct an ensemble of diverse and accurate classifiers, apply unlabeled examples to classifiers, and maintain a large disagreement (diversity) between them (Blum and Mitchell, 1998; Nigam et al., 2000; Kiritchenko and Matwin, 2001; Zhou and Goldman, 2004; Zhang and Sun, 2010; Yu et al., 2010; Hady and Schwenker, 2013), see also Zhou and Li's comprehensive survey paper (Zhou and Li, 2010).

A first instance of committee-based SSL is in a specific form of active learning. Each classifier votes on the class label of the unlabeled data. The pattern on which the ensemble disagree the most is considered to be the most informative for the overall ensemble classifier. This framework is called *query by committee (QBC)* (Seung et al., 1992; Settles, 2009; Zhou and Li, 2010). The two major issues for active learning in ensembles are (1) how to measure the level of disagreement, and (2) how to generate a diverse

ensemble of classifiers. The entropy index for ensemble classifiers can be formulated through $-\sum_y V_y/C \log V_y/C$ where C is the ensemble size and V_y is the number of votes for label y in the ensemble (Dagan and Engelson, 1995). Disagreement measures based on the *Kullback–Leibler divergence* (McCallum and Nigam, 1998) have been proposed as well. In Freund et al. (1997) it is proven that QBC can exponentially improve the sample complexity in comparison to passive supervised learning. This is mostly a theoretical result, but the diversity among the base classifiers can be achieved by other ensemble learning algorithms employing *bagging* or *boosting* (Zhou, 2012).

Multi-view learning was first introduced for SSL by Blum and Mitchell in the context of *co-training* of two views (Blum and Mitchell, 1998). They state two strong requirements for successful co-training: the two views (or, feature sets) should be conditionally independent given the class label, and either of them must be sufficient to learn the classification task. First, two classifiers are trained using the labeled training data $\mathcal{S}$, and then in each further iteration, each classifier predicts the class label of the unlabeled examples, estimates the confidence in its prediction, ranks the examples by confidence, and includes the *most confident* examples to the labeled training set. In doing so, it is expected that the *most confident* examples with respect to one classifier will be *informative* for the other. Here, an example is considered to be informative with respect to a classifier if it carries some discriminating information, i.e. if it lies close to the decision boundary (thus, adding it to the training set can improve the classifier's performance).

Nigam and Ghani (2000) showed that co-training is sensitive to the view independence requirement, and they proposed another multi-view semi-supervised algorithm, called *Co-EM*. It uses the model learned in one view to probabilistically label the unlabeled examples in the other model. Intuitively, Co-EM runs EM in each view and before each new EM iteration, inter-changes the probabilistic labels predicted in each view. Co-EM is considered as a probabilistic variant of co-training. Both algorithms are based on the same idea: they use the knowledge acquired in one view, in the form of soft class labels for the unlabeled examples, to train the other view. The major difference between the two algorithms is that Co-EM does not commit to the labels predicted in the previous iteration as it uses probabilistic labels that may change from one iteration to the other. On the other hand, co-training commits to the most confident predictions that are once added into the training set but are never revisited. Thus, it may add to the training set a large number of mislabeled examples.

In a number of recent studies (Goldman and Zhou, 2000; Zhou and Goldman, 2004; Li and Zhou, 2007; Hady and Schwenker, 2010), the applicability of co-training using a single view without feature splitting has been investigated. Goldman and Zhou (2000) first presented a single-view SSL method, called *statistical co-learning*. It uses two different supervised learning algorithms with the assumption that each of them produce a hypothesis that partition the input space into a set of equivalence classes. For example, a decision tree partitions the input space with one equivalence class per leaf. A ten-fold cross validation procedure is used to: (1) select the most confident examples to label at each iteration; and (2) combine the two hypotheses producing the final decision. Its drawbacks are: first, the assumptions concerning the used algorithms limits its applicability; second, the amount of available labeled data may be insufficient for applying cross validation (which is also time-consuming). Zhou and Goldman (2004) then presented another single view method, called *democratic co-learning* which is applied to three or more supervised learning algorithms and reduce the need for statistical tests. Therefore, it resolves the drawbacks of statistical co-learning but it still uses the time-consuming cross-validation technique to measure confidence intervals. These confidence intervals are used to select the most confident unlabeled examples and to combine the hypotheses decisions.

Zhou and Li's survey article (Zhou and Li, 2010) provides an excellent introduction to disagreement-based SSL in ensembles. Besides a theoretical foundation of SSL in ensembles, they review various relevant algorithms such as *tri-training*, *co-forest*, or *SSAIR* a disagreement-based active learning method, and provide several real-world applications in SSL in ensembles.

7. Partially supervised learning in artificial neural networks

Being one of the most prominent instances of learning machines, artificial neural networks (ANN) have been involved in the development and application of PSL approaches to a significant extent. Traditionally, popular ANN architectures and training algorithms could be found either in the supervised or unsupervised frameworks. Several reviews are available in the literature, e.g. Hertz et al. (1991). In the remainder of this section we will outline the intimate relationship between ANN training and PSL in standard architectures, then we will gradually move towards more recent, ad hoc PSL-based ANN.

7.1. PSL in multilayer perceptrons

An early PSL technique for training a multilayer perceptron (MLP) in classification tasks involving both labeled and unlabeled data was presented in Verikas et al. (2001). The approach was shown to be useful whenever the labeled subset of the training data is not representative enough of the statistical distribution of the class-conditional data (at least for some of the classes at hand). The basic idea behind this technique is pretty simple. First, the MLP is trained via regular backpropagation (BP) over the labeled portion of the training set. Once BP reaches its stopping criterion, the trained MLP is used for creating a soft-labeling of the classes whom (a part of) the unlabeled data are expected to belong to. In so doing, the unlabeled data undergo a sort of "pre-processing". The latter is further refined by iterating the whole procedure all over again, by exploiting the newly-created soft-labeling as additional target outputs for BP. More recently, a similar technique was applied to the task of context-sensitive change detection in remote sensing images (Patra et al., 2007). A variant of the approach is also outlined in Raychaudhuri and Dutta (2012), where it is applied to an image binarization task.

Supervised and unsupervised learning are combined in the training algorithm proposed in Trentin (2006), which can be extended to SSL eventually. An MLP is trained to estimate the probability density function underlying an unlabeled data set (a task related to manifold estimation), inherently an unsupervised task. This is accomplished by generating synthetic target outputs via an unbiased variant of traditional non-parametric density estimation techniques, such that regular supervised BP can then be applied to the MLP. It turns out that, due to its generalization capabilities, the MLP comes up with an estimate that outperforms the (memory-based) statistical techniques. If some of the data are labeled, they can be eventually used for a final training of the MLP (which, instead of being initialized at random, starts up with an assignment of parameter values that are good already, since they realize a sort of embedding of the whole input data set). The algorithm was successfully applied to a difficult forensic anthropology task (Trentin et al., 2011), where it is further combined with a supervised classifier according to several probabilistic, partially-supervised mixing criteria.

A much more sophisticated and intriguing approach to the problem of training a MLP in a SSL scenario is handed out in Malkin et al. (2009). A smooth, regularized optimization of the MLP

parameters W is sought which exploits both labeled data $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, \ldots, m\}$ and unlabeled data $\mathcal{U} = \{\mathbf{x}_i | i = 1, \ldots, M\}$, where $n = m + M$ is the overall number of input patterns. A properly computed graph-based manifold of the whole collection of input observations (described in their original feature space) is considered, where each pattern (either labeled or unlabeled) is a vertex in the complete graph, and where the undirected edge between the generic vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ is assumed to have weight ω_{ij} (the closer the corresponding patterns, the higher the weight). If we write $\mathbf{y}_W(\mathbf{x})$ to denote the output vector from the MLP when fed with input $\mathbf{x}$ (that is expected to be an estimate of the class-posterior probabilities for all classes at hand), then the optimization scheme sought can be obtained by replacing the standard BP algorithm (aimed at minimizing the squared error between the target outputs and the actual MLP outputs) with a stochastic gradient descent minimization of the following criterion function:

$$C(W) = \sum_{i=1}^{m} D(\hat{\mathbf{y}}_i, \mathbf{y}_W(\mathbf{x}_i)) + \alpha \sum_{i,j=1}^{n} \omega_{ij} D(\mathbf{y}_W(\mathbf{x}_i), \mathbf{y}_W(\mathbf{x}_j)) + \beta \sum_{i=1}^{n} D(\mathbf{y}_W(\mathbf{x}_i), \mathbf{u}) + \gamma \|W\| \quad (2)$$

where $D(\cdot, \cdot)$ denotes a distance measure between random vectors (the Kullback–Leibler divergence between probability distributions is recommended in Malkin et al., 2009), $\mathbf{u}$ represents the uniform distribution, $\|W\|$ is the usual ℓ_2-regularizer on the MLP parameters (e.g., the *weight-decay*), and α, β, and γ are in $\mathbb{R}^+$. The first term in the right-hand side of Eq. (2) boils basically down to the usual, supervised BP criterion (for α, β, and $\gamma = 0$). The second term is the manifold regularizer, encouraging smooth solutions over the graph (vertices that are close in the graph should yield a similar class-posterior probability). The third term, due to the minimization of the divergence w.r.t. the uniform distribution (especially in proximity of the separation surfaces between pairs of adjacent classes) acts as an entropic regularizer. Finally, as we say, the last term is the generic regularizer over the model parameters. The technique was successfully applied to a phone recognition task from individual acoustic frames drawn from speech signals (Malkin et al., 2009).

7.2. From "regular" MLPs to deep architectures

Several PSL techniques have been proposed for deep (many-hidden-layers) MLP architectures, too. The very notion of "deep learning" is related to SSL in a natural way. This fact is evident if we observe that, in their basic formulation, deep neural architectures are trained via algorithms that build upon the autoassociative ANN (a.k.a. the autoencoder) training scheme (Hertz et al., 1991). The latter is inherently a combination of supervised and unsupervised learning. An intrinsically unsupervised training set $\mathcal{T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is considered, and a two-layer MLP is trained as an autoassociative memory in order to map each input pattern $\mathbf{x}_i \in \mathcal{T}$ onto itself. The hidden layer is fixed such that the number of hidden units is (much) lower than the dimensionality of the feature space. In so doing, the MLP is expected to develop a reduced-dimensionality internal representation of the input feature vectors. Albeit revolving around unlabeled data, regular supervised BP may be used for training by using a labeled data set in the form $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{x}_1) \ldots, (\mathbf{x}_n, \mathbf{x}_n)\}$ (i.e., each input pattern is also its own target output). At the end of training, removing the hidden-to-output part of the MLP yields a feedforward ANN which realizes the feature transformation. The latter can be used as the input to another autoassociative MLP, trained in a similar manner, which develops an even lower-dimensional representation of the original signal. The process may be iterated as many times as needed in order to obtain the dimensionality reduction sought, each time mounting a new MLP on the top of the others. A deep architecture this way emerges. At the last step, on the top of this deep ANN a regular MLP is mounted, and trained via BP to solve any regression or classification task originally defined over $\mathcal{T}$ and a corresponding set of labels (i.e., targets). It is seen that in a SSL scenario all the input data in $\mathcal{T}$, either labeled or unlabeled, may as well be used for building the autoassociative modules of the deep MLP, while only the labeled data are eventually used for BP training of the topmost MLP (and, for refining the lower hidden layers by further BP steps deep down the network). A variant of this methodology is applied in Ranzato and Szummer (2008) to learn a proper representation of text documents. This variant includes a specialized first (input-to-hidden) layer, devoted to developing an internal representation of discrete input features, e.g. words from a natural language dictionary. Another variation on the theme is found in Zhou et al. (2010), where active learning is exploited (in a sentiment classification task) in order to enhance the SSL of a deep architecture built upon multiple layers of restricted Boltzmann machines.

A different, intriguing semi-supervised approach to deep learning is discussed in Weston et al. (2008). While preserving the general idea that the overall learning task is better accomplished by multi-task learning, such that any layer in the MLP may have its own unsupervised auxiliary task to be carried out, the original idea of having auxiliary encoding–decoding tasks (the aforementioned autoassociative mappings) is replaced by an encoding-only notion that involves SSL *embedding*. The latter has to be learned simultaneously with the ultimate supervised solution defined at the topmost layer of the deep ANN. Embedding relies on a symmetric matrix $\mathbf{\Theta} = [\theta_{ab}]$ of mutual "weights" between pairs of patterns, such that θ_{ab} is the proximity between $\mathbf{x}_a$ and $\mathbf{x}_b$. In the basic case, $\theta_{ab} = 1$ if $\mathbf{x}_a$ and $\mathbf{x}_b$ are close to each other (i.e., they are neighbors), otherwise $\theta_{ab} = 0$. The matrix $\mathbf{\Theta}$ must be defined in advance, relying on background knowledge (e.g., in audio or video processing any two consecutive time frames may be assumed to be neighbors), or on statistical pattern recognition algorithms (e.g., the k-nearest neighbor). Stochastic gradient descent is iteratively applied in order to optimize both the supervised criterion function (e.g., the sum of hinge-losses, say $H(\cdot)$, at the output units) and the unsupervised embedding criterion, say $L(g^{(h)}(\mathbf{x}_a), g^{(h)}(\mathbf{x}_b), \theta_{ab})$, where $g^{(h)}(\mathbf{x})$ is the embedding function of pattern $\mathbf{x}$ yielded by h-th layer of the ANN. Several instances of suitable criteria $L(g^{(h)}(\mathbf{x}_a), g^{(h)}(\mathbf{x}_b), \theta_{ab})$ are considered in Weston et al. (2008), such as multi-dimensional scaling, ISOMAP, "Siamese" ANN-like, etc. At each training iteration, the algorithm goes as follows:

1. Pick up a labeled pattern $(\mathbf{x}_i, \hat{\mathbf{y}}_i)$ at random from the training set, and apply a step of gradient descent of $H(\mathbf{y}(\mathbf{x}_i), \hat{\mathbf{y}}_i)$ to the whole ANN.
2. Loop: iterate over all layers h of the ANN:
   (a) a random pair $(\mathbf{x}_i, \mathbf{x}_j)$ of unlabeled "neighbors" is drawn from the training set, and a step of gradient descent of $L(g^{(h)}(\mathbf{x}_i), g^{(h)}(\mathbf{x}_j), 1)$ is applied;
   (b) another pattern $\mathbf{x}_k$ is chosen at random, and a gradient-descent step of $L(g^{(h)}(\mathbf{x}_i), g^{(h)}(\mathbf{x}_k), 0)$ is accomplished.

The last two steps (a) and (b) realize the embedding, i.e. neighboring input patterns (that are close to each other) are closely mapped, while distant patterns are projected onto separated regions of the transformed space. In so doing, degenerate solutions are avoided without requiring any explicit balancing constraints to be added in the optimization scheme. The approach turns out to be simple, and effective in the field. It was successfully applied to the prediction of interactions between HIV1 and human proteins (Qi et al., 2010).

7.3. PSL in radial basis function networks

Just like MLP, radial basis function (RBF) networks got involved in PSL research. From a certain viewpoint, RBF have always been lying on the edge between supervised and unsupervised learning. Implicitly or explicitly, they combine an unsupervised model (i.e., a Gaussian mixture model (GMM) of the feature vectors, realized by the RBF hidden units) and a supervised one (the linear combination of the hidden nonlinearities realized by the output layer). In fact, training is easily achieved via (i) maximum-likelihood estimation of the parameters of the GMM, followed by (ii) any supervised estimation of a linear regression model between the GMM outputs and the ANN target outputs. Once again, SSL may thus be accounted for by carrying out step (i) over the whole set of labeled and unlabeled data, and step (ii) over the labeled data only. Several ad hoc PSL specializations of this basic scheme are found in the literature. In Bouchachia (2005) regular supervised clustering techniques, e.g. the popular fuzzy k-means, are applied in order to cluster both labeled and unlabeled data in order to find more robust centroids for the mean vectors of the RBF Gaussian kernels. The outcome of the clustering procedure is also used for creating synthetic labels for (some of) the unlabeled data, to be exploited during the supervised RBF training phase. In Luo and Zhang (2008) the EM algorithm is used in an active learning framework for improving SSL in RBF applied to content-based image retrieval accounting for relevance feedback. An extremely similar approach (with the addition of explicit, suitable feature selection/dimensionality reduction preprocessing of the data) is discussed in Jiang (2009) in the realm of text classification. Along the same line, Constantinopoulos and Likas (2008) uses an ANN called probabilistic RBF (a specialization of regular RBF where the i-th output unit is interpreted as the class-conditional probability density function $p(\mathbf{x}|y_i)$ of observation $\mathbf{x}$ given the class y_i, that is a GMM estimated via maximum-likelihood) in a SSL framework with an original active learning mechanism. Finally, Hady et al. (2010) presents a novel tree-structured ensemble machine built upon RBF relying on a co-training algorithm for SSL.

7.4. Partially supervised self-organizing maps

While the architectures discussed so far (MLP—either shallow or deep—and RBF) originated in the supervised framework and were later adapted to PSL scenarios, PSL variants of Kohonen's self-organizing maps (SOM) (Hertz et al., 1991) are originally rooted in the unsupervised setup. A simple, good example of SOM-based SSL is pointed out in Herrmann and Ultsch (2007). The idea is the following. Let us assume we are faced with a L-class classification task. First, train a regular (emergent) SOM on the overall original bunch of available input data, where each neuron in the SOM is associated with a L-dimensional label vector (encapsulating the expected class-posterior probabilities). A proximity graph is then devised by creating a vertex in the graph for each neuron in the SOM, adding edges between any given pair of vertices, and setting the edge weight to equal the pairwise distance (as learned by the SOM) between the corresponding neurons. Zhu's label propagation algorithm is eventually applied relying on the resulting proximity graph, therefore realizing the building block for SSL.

Hasegawa et al. investigated variants of the SOM where the topology of the ANN can grow and adapt incrementally as training proceeds, such that there is no need to fix the number of neurons/vertices of the graph in advance (Furao et al., 2007; Kamiya et al., 2007). This paradigm is known as the self-organizing incremental neural network (SOINN). One- or multi-layer versions are available. SOINN were applied to incremental online semi-supervised active learning (Furao et al., 2007; Shen et al., 2011), as well as to online semi-supervised clustering (Kamiya et al., 2007).

7.5. Other partially supervised connectionist architectures

Other ANN-based approaches to PSL revolved around less popular, or even ad hoc connectionist architectures. An evaluation of retraining rules for SSL in recurrent ANN is carried out in Frinken and Bunke (2009), in the domain of cursive word recognition. In self-training for SSL, a machine (previously trained from labeled examples) has to select some unlabeled data, and to self-label them for its own re-training. The selection strategy affects the overall SSL process to a significant extent. Strategies differ insofar as they may be either deterministic or probabilistic, usually relying on a certain confidence threshold (e.g., select the unlabeled data whose highest estimated class-posterior exceeds the threshold). The threshold is possibly tuned (i.e., learned) at training time, as well. The study reported in Frinken and Bunke (2009) investigates empirically the consequences of different choices for such selection strategies.

Learning a semi-supervised node labeling from unbalanced data in a specific type of recurrent ANN, namely the Hopfield net, is outlined (for instance) in Frasca et al. (2013). Note that, due to their very nature, Hopfield nets are intrinsically suitable for "completion" tasks, such as the labeling of the unlabeled nodes (i.e., the corresponding neurons) in a proximity graph (i.e., the network itself) which, initially, is only partially labeled. Furthermore, in Bertoni et al. (2011) the authors investigate a cost sensitive Hopfield network (COSNet) for SSL of vertex-labels in graphs, as an alternative to Zhu's label propagation as well as to standard Hopfield nets, capable of preserving the prior knowledge and of accounting for unbalanced data sets (i.e., uneven fractions of positive vs. negative examples). Basically, the latter goals are achieved by partitioning the COSNet topology into two, distinct ANN: a "labeled", traditional Hopfield sub-net; and a (somewhat complementary) unlabeled Hopfield sub-net, where a specific variant of label propagation is accomplished.

Amongst the other ad hoc ANN for PSL, it worths mentioning the "hybrid neural network" architecture proposed in Guan et al. (2007), which combines an MLP and an adaptive resonance theory (ART-2) network (Hertz et al., 1991). Capable of processing both supervised and unsupervised data, this hybrid ANN is successfully applied to semi-supervised clustering, and turns even out to improve standard unsupervised clustering. Finally, an evolving granular neural network (basically, an adaptive fuzzy neural classifier) is proposed in Leite et al. (2010) for the SSL and classification of data generated by non-stationary streams.

8. Conclusion

During the last 20 years a variety of different methods and algorithms have been proposed to train models in pattern recognition applications (Artstein and Poesio, 2008; Guillaumin et al., 2010; Scherer et al., 2012) using partially or weakly labeled training data sets. In this survey paper we have reviewed several current concepts of machine learning for pattern recognition under partial supervision: active learning, learning with fuzzy labels, semi-supervised classification, semi-supervised clustering, partially supervised learning in ensembles, and partially supervised learning in neural networks. All learning schemes face the issue that labeled data are hard to obtain, and reliably labeled data are rare. Particularly, SSL and active learning tackle the same problem but from different directions. Whereas active learning is a particular case of supervised learning (actually, all the labels are given by a human expert), semi-supervised classification aims to solve a

supervised learning problem using labels generated by the learning machine itself. Self-labeled data (even those patterns that are labeled with very high confidence) may be assigned false labels, and accumulating such incorrectly labeled data in the training set might degrade the classifier performance. Therefore, combinations of active learning and SSL are seriously useful in practical applications, where the set of informative data—the data close to the decision boundary—is labeled by the expert and the set of non-informative (or, confident) data is automatically labeled by the learner itself. In order to avoid the accumulation of mislabeling noise, one could allow fuzzy labels, or labels together with some confidence value during the self-labeling phase, and thus SSL algorithms capable to use soft-labeled data might be a promising direction of future research.

Other directions (and, challenges) for further development of the field, aimed at overcoming several open issues, are the following. First, closing the remaining gap between PSL research and the vast knowledge the community has accumulated throughout the decades in the related area of statistical pattern recognition from missing/incomplete data. Then, fully exploiting the notion that the unlabeled subset of the data can be regarded as a highly informative collection of implicit constraints posed on the labeled data subset. These constraints are the consequence of the topological and/or statistical properties of the unlabeled data. Hence, any classifier trained from the supervised portion of the data shall comply with these constraints. In so doing, if the classifiers $\mathcal{C}_a$ and $\mathcal{C}_b$ yielded similar values of the supervised loss function on the labeled training set, still $\mathcal{C}_a$ would outperform $\mathcal{C}_b$ if the former did not implicitly violate the topological/statistical properties of the unlabeled data subset (while $\mathcal{C}_b$ did). In other words, not only constraints must be satisfied, they should also improve learning effectively. Finally, neural networks and other machine learning paradigms which are not explicitly statistical in nature could benefit from a probabilistic interpretation of the underlying laws they learn to encapsulate, in such a way that improved laws are learned by complying with the probability density function of the whole (labeled and unlabeled) data.

References

Adankon, M., Cheriet, M., 2010. Genetic algorithm based training for semi-supervised SVM. Neural Computing and Applications 19, 1197–1206.

Alpaydin, E., 2010. Introduction to Machine Learning, second ed. MIT Press.

Artstein, R., Poesio, M., 2008. Inter-coder agreement for computational linguistics. Computational Linguistics 34 (4), 555–596.

Basu, S., 2005. Semi-supervised clustering: probabilistic models, algorithms and experiments. Ph.D. Thesis, University of Texas at Austin. <http://www.cs.utexas.edu/users/ai-lab/?basu:thesis05>.

Basu, S., Banerjee, A., Mooney, R., 2002. Semi-supervised clustering by seeding. In: Proc. of the 19th International Conference on, Machine Learning (ICML'02), pp. 19–26.

Basu, S., Bilenko, M., Mooney, R., 2004. A probabilistic framework for semi-supervised clustering. In: Proc. of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'04), pp. 59–68.

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

Bertoni, A., Frasca, M., Valentini, G., 2011. COSNet: a cost sensitive neural network for semi-supervised learning in graphs. In: Proc. of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'11). Springer, Berlin, Heidelberg, pp. 219–234.

Bie, T.D., Cristianini, N., 2006. Semi-supervised learning using semi-definite programming. In: O., Chapelle, Schölkopf, B., Zien, A. (Eds.), Semi-Supervised Learning. MIT Press, pp. 119–135.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Blum, A., Chawla, S., 2001. Learning from labeled and unlabeled data using graph mincuts. In: Proc. of the 18th International Conference on, Machine Learning (ICML'01), pp. 19–26.

Blum, A., Mitchell, T., 1998. Combining labeled and unlabeled data with co-training. In: Proc. of the 11th Annual Conference on Computational Learning Theory (COLT 1998), pp. 92–100.

Bouchachia, A., 2005. RBF networks for learning from partially labeled data. In: Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data, pp. 10–19.

Chapelle, O., Zien, A., 2005. Semi-supervised learning by low density separation. In: Proc. of the 10th International Workshop on Artificial Intelligence and, Statistics, pp. 57–64.

Chapelle, O., Chi, M., Zien, A., A continuation method for semi-supervised SVMs. In: International Conference on, Machine Learning (ICML'06), pp. 185–192.

Chapelle, O., Sindhwani, V., Keerthi, S., Branch and bound for semi-supervised support vector machines. In: Advances in Neural Information Processing Systems (NIPS'06), pp. 217–224.

Chapelle, O., Sindhwani, V., Keerthi, S.S., 2008. Optimization techniques for semi-supervised support vector machines. The Journal of Machine Learning Research 9, 203–233.

Chapelle, O., Schölkopf, B., Zien, A., 2010. Semi-Supervised Learning. MIT Press.

Chu, S.M., Tang, H., Huang, T.S., 2009. Fisher-voice and semi-supervised speaker clustering. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'F09). IEEE, pp. 4089–4092.

Constantinopoulos, C., Likas, A., 2008. Semi-supervised and active learning with the probabilistic RBF classifier. Neurocomputing 71 (13–15), 2489–2498.

Cozman, F.G., Cohen, I., 2002. Unlabeled data can degrade classification performance of generative classifiers. In: Proc. of the 15th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS'02), pp. 327–331.

Dagan, I., Engelson, S.P., 1995. Committee-based sampling for training probabilistic classifiers. In: Proc. of the 12th International Conference on Machine Learning (ICML'95). Morgan Kaufmann, pp. 150–157.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B Methodological 39 (1), 1–38.

Faußer, S., Schwenker, F., 2012. Semi-supervised kernel clustering with sample-to-cluster weights. In: Schwenker, F., Trentin, E. (Eds.), Partially Supervised Learning (PSL'11), LNAI, vol. 7081. Springer, pp. 72–81.

Frasca, M., Bertoni, A., Re, M., Valentini, G., 2013. A neural network algorithm for semi-supervised node label learning from unbalanced data. Neural Networks 43, 84–98.

Freund, Y., Seung, H., Shamir, E., Tishby, N., 1997. Selective sampling using the query by committee algorithm. Machine Learning 28 (2–3), 133–168.

Frinken, V., Bunke, H., 2009. Evaluating retraining rules for semi-supervised learning in neural network based cursive word recognition. In: Proc. of the 10th International Conference on Document Analysis and Recognition (ICDAR'09). IEEE Computer Society, Washington, DC, USA, pp. 31–35.

Frinken, V., Bunke, H., 2009. Self-training strategies for handwriting word recognition. In: Advances in Data Mining. Applications and Theoretical Aspects. Springer, pp. 291–300.

Fung, G., Mangasarian, O., 2001. Semi-supervised support vector machines for unlabeled data classification. Optimization Methods and Software 15, 29–44.

Furao, S., Sakurai, K., Kamiya, Y., Hasegawa, O., 2007. An online semi-supervised active learning algorithm with self-organizing incremental neural network. In: International Joint Conference on Neural Networks (IJCNN 2007), pp. 1139–1144.

Gayar, N.E., Schwenker, F., Palm, G., 2006. A study of the robustness of KNN classifiers trained using soft labels. In: Schwenker, F., Marinai, S. (Eds.), Artificial Neural Networks in Pattern Recognition (ANNPR'06), LNAI, vol. 4087. Springer, pp. 67–80.

Goldman, S., Zhou, Y., 2000. Enhancing supervised learning with unlabeled data. In: Proc. of the 17th International Conference on, Machine Learning (ICML'00), pp. 327–334.

Grandvalet, Y., Bengio, Y., 2005. Semi-supervised learning by entropy minimization. Advances in Neural Information Processing Systems (NIPS'05) 17, 529–536.

Guan, D., Gavrilov, A., Yuan, W., Lee, Y.-K., Lee, S., 2007. A novel hybrid neural network for data clustering. In: Arabnia, H.R., Dehmer, M., Emmert-Streib, F., Yang, M.Q. (Eds.), MLMTA. CSREA Press, pp. 284–288.

Guillaumin, M., Verbeek, J., Schmid, C., 2010. Multimodal semi-supervised learning for image classification. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 902–909.

Hady, M. Abdel, Schwenker, F., 2010. Combining committee-based semi-supervised learning and active learning. Journal of Computer Science and Technology (JCST): Special Issue on Advances in Machine Learning and Applications 25 (4), 681–698.

Hady, M.F.A., Schwenker, F., 2013. Semi-supervised learning. In: Handbook on Neural Information Processing. Springer, pp. 215–239.

Hady, M.F.A., Schwenker, F., Palm, G., 2010. Semi-supervised learning for tree-structured ensembles of RBF networks with co-training. Neural Networks 23 (4), 497–509.

Hakkani-Tür, D., Riccardi, G., Tur, G., 2006. An active approach to spoken language processing. ACM Transaction on Speech Language Processing 3 (3), 1–31.

Haque, M., Holder, L., Skinner, M., Cook, D., 2013. Generalized query based active learning to identify differentially methylated regions in DNA. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1.

Herrmann, L., Ultsch, A., 2007. Label propagation for semi-supervised learning in self-organizing maps. In: Proc. of the 6th International Workshop on Self-Organizing Maps (WSOM'07), Bielefeld Germany.

Hertz, J., Krogh, A., Palmer, R., 1991. Introduction to the Theory of Neural Computation. Addison Wesley.

Inoue, M., Ueda, N., 2003. Exploitation of unlabeled sequences in hidden Markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12), 1570–1581.

Jain, A.K., 2010. Data clustering: 50 years beyond *k*-means. Pattern Recognition Letters 31 (8), 651–666.

Jain, A.K., Duin, R.P.W., Mao, J., 2000. Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1), 4–37.

Jiang, E.P., 2009. Semi-supervised text classification using RBF networks. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (Eds.), IDA, vol. 5772. Springer, pp. 95–106.

Joachims, T., 1999. Transductive inference for text classification using support vector machines. In: Proc.of the 16th International Conference on, Machine Learning (ICML'99), pp. 200–209.

Kalakech, M., Biela, P., Macaire, L., Hamad, D., 2011. Constraint scores for semi-supervised feature selection: a comparative study. Pattern Recognition Letters 32 (5), 656–665.

Kamiya, Y., Ishii, T., Furao, S., Hasegawa, O., 2007. An online semi-supervised clustering algorithm based on a self-organizing incremental neural network, In: International Joint Conference on, Neural Networks (IJCNN'07), pp. 1061–1066.

Karayiannis, N., Bezdek, J., 1997. An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering. IEEE Transactions on Fuzzy Systems 5 (4), 622–628.

Kiritchenko, S., Matwin, S., 2001. E-mail classification with co-training. In: Proc. of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON'01), pp. 8–19.

Kulis, B., Basu, S., Dhillon, I., Mooney, R., 2009. Semi-supervised graph clustering: a kernel approach. Machine Learning 74 (1), 1–22.

Lee, H., Yoo, J., Choi, S., 2010. Semi-supervised nonnegative matrix factorization. IEEE Signal Processing Letters 17 (1), 4–7.

Leite, D., Costa, P., Gomide, F., 2010. Evolving granular neural network for semi-supervised data stream classification. In: International Joint Conference on Neural Networks (IJCNN), pp. 1877–1884.

Lewis, D., Catlett, J., 1994. Heterogeneous uncertainty sampling for supervised learning. In: Proc. of the 11th International Conference on, Machine Learning (ICML'94), pp. 148–156.

Li, M., Zhou, Z.-H., 2007. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans 37 (6), 1088–1098.

Li, Y., Guan, C., Li, H., Chin, Z., 2008. A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. Pattern Recognition Letters 29 (9), 1285–1294.

Lindenbaum, M., Markovitch, S., Rusakov, D., 2004. Selective sampling for nearest neighbor classifiers. Machine Learning 54 (2), 125–152.

Luo, Z.-P., Zhang, X.-M., 2008. A semi-supervised learning based relevance feedback algorithm in content-based image retrieval. In: Chinese Conference on Pattern Recognition (CCPR '08), pp. 1–4.

Malkin, J., Subramanya, A., Bilmes, J., 2009. On the semi-supervised learning of multi-layered perceptrons. In: INTERSPEECH. ISCA, pp. 660–663.

McCallum, A., Nigam, K., 1998. Employing em and pool-based active learning for text classification. In: Proc. of the 15th International Conference on, Machine Learning (ICML'98), pp. 350–358.

Meudt, S., Schwenker, F., 2012. On instance selection in audio based emotion recognition. In: Mana, N., Schwenker, F., Trentin, E. (Eds.), Artificial Neural Networks in Pattern Recognition (ANNPR'12), LNAI, vol. 7477. Springer, pp. 186–192.

Miller, D.J., Uyar, H.S., 1997. A mixture of experts classifier with learning based on both labelled and unlabelled data. Advances in Neural Information Processing Systems 9, 571–577.

Nagy, G., Shelton, G., 1966. Self-corrective character recognition system. IEEE Transactions on Information Theory 12 (2), 215–222.

Nigam, K., 2001. Using unlabeled data to improve text classification. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

Nigam, K., Ghani, R., 2000. Analyzing the effectiveness and applicability of co-training. In: Proc. of the Ninth International Conference on Information and, Knowledge Management, pp. 86–93.

Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T., 2000. Text classification from labeled and unlabeled documents using EM. Machine Learning 39 (2–3), 103–134.

Patra, S., Bruzzone, L., 2012. A cluster-assumption based batch mode active learning technique. Pattern Recognition Letters 33 (9), 1042–1048.

Patra, S., Ghosh, S., Ghosh, A., 2007. Semi-supervised learning with multilayer perceptron for detecting changes of remote sensing images. In: Ghosh, A., De, R., Pal, S. (Eds.), Pattern Recognition and Machine Intelligence, vol. 4815. Springer, Berlin, Heidelberg, pp. 161–168.

Peng, B., Qian, G., Ma, Y., 2009. Recognizing body poses using multilinear analysis and semi-supervised learning. Pattern Recognition Letters 30 (14), 1289–1294.

Platt, J.C., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. MIT Press, pp. 61–74.

Poggio, T., Girosi, F., 1989. A theory of networks for approximation and learning. Laboratory, Massachusetts Institute of Technology, Technical Report 1140.

Qi, Y., Tastan, O., Carbonell, J.G., Klein-Seetharaman, J., Weston, J., 2010. Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. Bioinformatics 26 (18), i645–i652.

Ranzato, M.A., Szummer, M., 2008. Semi-supervised learning of compact document representations with deep networks. In: Proc. of the 25th International conference on Machine Learning (ICML'08). ACM, New York, NY, USA, pp. 792–799.

Raychaudhuri, A., Dutta, J., 2012. Image binarization using multi-layer perceptron: a semi-supervised approach. International Journal of Engineering Innovations and Research 1 (2), 134–139.

Scheffer, T., Decomain, C., Wrobel, S., 2001. Active hidden Markov models for information extraction. In: Hoffmann, F., Hand, D.J., Adams, N.M., Fisher, D.H., Guimarães, G. (Eds.), Advances in Intelligent Data Analysis (IDA'01), LNCS, vol. 2189. Springer, pp. 309–318.

Scherer, S., Glodek, M., Layher, G., Schels, M., Schmidt, M., Brosch, T., Tschechne, S., Schwenker, F., Neumann, H., Palm, G., 2012. A generic framework for the inference of user states in human computer interaction: how patterns of low level communicational cues support complex affective states. Journal on Multimodal User Interfaces 6 (3), 117–141.

Scherer, S., Kane, J., Gobl, C., Schwenker, F., 2013. Investigating fuzzy-input fuzzy-output support vector machines for robust voice quality classification. Computer Speech & Language 27 (1), 263–287.

Schwenker, F., Kestler, H.A., Palm, G., 2001. Three learning phases for radial-basis-function networks. Neural Networks 14 (4–5), 439–458.

Seeger, M., 2002. Learning with labeled and unlabeled data. Technical Report, University of Edinburgh, Institute for Adaptive and Neural Computation.

Seo, S., Bode, M., Obermayer, K., 2003. Soft nearest prototype classification. IEEE Transactions on Neural Networks 14 (2), 390–398.

Settles, B., 2009. Active learning literature survey. Tech. rep., Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI.

Settles, B., Craven, M., 2008. An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08). Association for Computational Linguistics, pp. 1070–1079.

Seung, H.S., Opper, M., Sompolinsky, H., 1992. Query by committee. In: Proc. of the Fifth Annual Workshop on Computational Learning Theory (COLT '92). ACM, pp. 287–294.

Shahshahani, B., Landgrebe, D., 1994. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. IEEE Transactions on Geoscience and Remote Sensing 32 (5), 1087–1095.

Shen, F., Yu, H., Sakurai, K., Hasegawa, O., 2011. An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network. Neural Computing and Applications 20 (7), 1061–1074.

Sindhwani, V., Keerthi, S., Chapelle, O., 2006. Deterministic annealing for semi-supervised kernel machines. In: International Conference on, Machine Learning (ICML'06), pp. 841–848.

Singh, A., Nowak, R., Zhu, X., 2008. Unlabeled data: now it helps, now it doesn't. Advances in Neural Information Processing Systems (NIPS'08) 21, 1513–1520.

Soleymani Baghshah, M., Bagheri Shouraki, S., 2010. Kernel-based metric learning for semi-supervised clustering. Neurocomputing 73 (7), 1352–1361.

Song, Y., Nie, F., Zhang, C., 2008. Semi-supervised sub-manifold discriminant analysis. Pattern Recognition Letters 29 (13), 1806–1813.

Thiel, C., Giacco, F., Schwenker, F., Palm, G., 2008. Comparison of neural classification algorithms applied to land cover mapping. In: Apolloni, B., Bassis, S., Marinaro, M. (Eds.), New Directions in Neural Networks (WIRN 2008), Frontiers in Artificial Intelligence and Applications, vol. 193. IOS Press, pp. 254–263.

Thiel, C., Scherer, S., Schwenker, F., 2007. Fuzzy-input fuzzy-output one-against-all support vector machines. In: Apolloni, B., Howlett, R.J., Jain, L.C. (Eds.), Knowledge-Based Intelligent Information and Engineering Systems (KES'07), LNCS, vol. 4694. Springer, pp. 156–165.

Thiel, C., Sonntag, B., Schwenker, F., 2008. Experiments with supervised fuzzy LVQ. In: Prevost, L., Marinai, S., Schwenker, F. (Eds.), Artificial Neural Networks in Pattern Recognition (ANNPR'08), LNAI, vol. 5064. Springer, pp. 125–132.

Tong, S., Chang, E., 2001. Support vector machine active learning for image retrieval. In: Proc.of the 9th ACM International Conference on Multimedia (MULTIMEDIA'01). ACM, pp. 107–118.

Tong, S., Koller, D., 2002. Support vector machine active learning with applications to text classification. Journal Machine Learning Research 2, 45–66.

Trentin, E., 2006. Simple and effective connectionist nonparametric estimation of probability density functions. In: Proc. of the IAPR Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR'06). Springer, pp. 1–10.

Trentin, E., Lusnig, L., Cavalli, F., 2011. Comparison of combined probabilistic connectionist models in a forensic application. In: Proc. of the First IAPR Workshop on Partially Supervised Learning (PSL'11). Springer, pp. 128–137.

Tuia, D., Volpi, M., Copa, L., Kanevski, M., Munoz-Mari, J., 2011. A survey of active learning algorithms for supervised remote sensing image classification. IEEE Journal of Selected Topics in Signal Processing 5 (3), 606–617.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag.

Verikas, A., Gelzinis, A., Malmqvist, K., 2001. Using unlabelled data to train a multilayer perceptron. Neural Processing Letters 14 (3), 179–201.

Villmann, T., Schleif, F.-M., Hammer, B., 2005. Fuzzy labeled soft nearest neighbor classification with relevance learning. In: Proc.of the Fourth International Conference on Machine Learning and Applications (ICMLA'05), pp. 11–15.

Wagstaff, K.L., 2010. Constrained clustering. In: Encyclopedia of Machine Learning. Springer, pp. 220–221.

Wagstaff, K., Cardie, C., Schroedl, S., 2001. Constrained k-means clustering with background knowledge. In: Proc. of the 18th International Conference on, Machine Learning (ICML'01), pp. 577–584.

Weston, J., Ratle, F., Collobert, R., 2008. Deep learning via semi-supervised embedding. In: Proc. of the 25th international Conference on Machine Learning (ICML'08). ACM, New York, NY, USA, pp. 1168–1175.

Young, T., Farjo, A., 1972. On decision directed estimation and stochastic approximation. IEEE Transactions on Information Theory 18 (5), 671–673.
Yu, Z., Su, L., Li, L., Zhao, Q., Mao, C., Guo, J., 2010. Question classification based on co-training style semi-supervised learning. Pattern Recognition Letters 31 (13), 1975–1980.
Zhang, L., 2013. Contextual and active learning-based affect-sensing from virtual drama improvisation. ACM Transaction on Speech Language Processing 9 (4), 8:1–8:25.
Zhang, Q., Sun, S., 2010. Multiple-view multiple-learner active learning. Pattern Recognition 43 (9), 3113–3119.
Zhou, Z.-H., 2012. Ensemble Methods: Foundations and Algorithms. Chapman Hall/CRC.
Zhou, Y., Goldman, S., 2004. Democratic co-learning. In: Proc. of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04). IEEE Computer Society, pp. 594–602.
Zhou, Z.-H., Li, M., 2005. Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17 (11), 1529–1541.
Zhou, Z.-H., Li, M., 2005. Semi-supervised regression with co-training. In: Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05), pp. 908–913.
Zhou, Z.-H., Li, M., 2010. Semi-supervised learning by disagreement. Knowledge and Information Systems 24 (3), 415–439.
Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B., 2004. Learning with local and global consistency. Advances in Neural Information Processing Systems 16, 753–760.
Zhou, Z.-H., Zhang, D., Chen, S., 2007. Semi-supervised dimensionality reduction. In: Proc. of the Seventh SIAM International Conference on Data Mining (SDM'07), pp. 629–634.
Zhou, S., Chen, Q., Wang, X., 2010. Active deep networks for semi-supervised sentiment classification. In: Proc.of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1515–1523.
Zhu, X., 2008. Semi-supervised learning literature survey. Technical Report 1530.
Zhu, X., Ghahramani, Z., Lafferty, J., 2003. Semi-supervised learning using gaussian fields and harmonic functions. In: Proc. of the 20th International Conference on, Machine Learning (ICML'03), pp. 912–919.