ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Machine Learning with Labeled and Unlabeled Data

Tijl De Bie[3], Thiago Turchetti Maia[2] and Antonio Padua Braga[1]

1- Federal University of Minas Gerais, Brazil
2 - Vetta Innovation; Federal University of Minas Gerais, Brazil
3 - University of Bristol, UK

Abstract. The field of semi-supervised learning has been expanding rapidly in the past few years, with a sheer increase in the number of related publications. In this paper we present the SSL problem in contrast with supervised and unsupervised learning. In addition, we propose a taxonomy with which we categorize many existing approaches described in the literature based on their underlying framework, data representation, and algorithmic class.

1 Introduction

Machine learning has traditionally been divided in Unsupervised Learning (UL) on the one hand, and Supervised Learning (SL) on the other hand. In UL, which is sometimes referred to as exploratory data analysis, a set of data points is given, and the task is to discern any structure present in the data set. No label information about the data points is given to assist in or *supervise* this task. In SL, in contrast, for each data point in a training set a label is given, and *inductive inference* is used to infer a predictive relation between data points and the associated labels.

In this paper we provide a brief overview of the many existing approaches to the new field of Semi-Supervised learning (SSL). We start by comparing SSL to both SL and UL. We then propose a taxonomy to categorize different approaches based on three fundamental characteristics: underlying framework, data representation, and algorithmic class.

In this paper we also map the contributions to the ESANN 2009 special session on SSL to the proposed taxonomy. All SSL references in this paper are summarized in Table 1, along with their respective categorization according to each taxonomic characteristic.

2 Semi-Supervised Learning

Many SSL approaches have evolved from SL, UL, or both. The next two sections describe the SSL problem in contrast with its SL and UL ancestors.

2.1 Semi-Supervised Learning as a variant of Supervised Learning

In general, an inductive model can be described by the elements presented in Figure 1. The data generator (DG) selects random samples $\mathbf{x} \in \mathcal{X}$, drawn independently from an unknown but fixed probability distribution function $F(\mathbf{x})$. An oracle (O) returns a label $y \in \mathcal{Y}$ to every input sample $\mathbf{x}$ with respect to an unknown but fixed conditional distribution function $F(y|\mathbf{x})$. For inductive inference to be applied, a so-called training set Γ_l is assumed to be available:

$$\Gamma_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}\,, \tag{1}$$

with all pairs $(\mathbf{x}_i, y_i)$ drawn $i.i.d.$ by DG from the joint distribution $F(\mathbf{x}, y) = F(\mathbf{x})F(y|\mathbf{x})$.

A third component is the evaluator (E), which is capable of implementing any function f_α from the set $\mathcal{F} = \{f_\alpha : \mathcal{X} \to \mathcal{Y} | \alpha \in \Lambda\}$, where Λ is an arbitrary set of parameters that governs the behaviour of the function. Finally, the learning machine (LM) uses the training sample Γ_l to select one particular function $f_{\alpha_0} \in \mathcal{F}$, e.g. using ideas from empirical risk minimization or related approaches.

Once the machine is trained, the value $\alpha_0 \in \Lambda$ is determined and the function f_{α_0} can be implemented by E. For well-designed function classes $\mathcal{F}$, LM's, and loss functions, the risk is small with high probability [2], where the risk is defined as the expected loss incurred by the selected function f_{α_0} in predicting the label of an input point $\mathbf{x}$ sampled from $F(\mathbf{x})$. Then f_{α_0} may be used to estimate unknown values $\hat{y}^*$ at arbitrary points $\mathbf{x}^*$ of an unlabeled test set Γ_u.

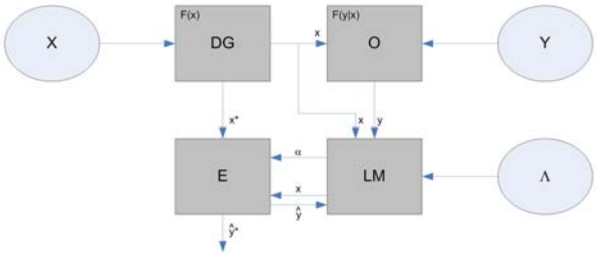

Fig. 1: Learning model based on inductive inference, where a function that best approximates the unknown functional dependence is selected from the set of all possible functions based on the given training data.

SL relies on the assumption that the data point/label pairs in the labeled data set Γ_l and in the unlabeled data set Γ_u are sampled i.i.d. from the joint distribution [3], so that a single set of parameters α_0, induced from Γ_l, is likely to also fit the test set well. From the function approximation point of view, the generator function is assumed to be stationary and Γ_l representative. This allows the estimation of joint densities $F(\mathbf{x}, y)$ and the construction of discriminative and regressive models represented by f_{α_0}. The induction of a general (rule) hyper-surface based on the assumptions of stationarity and representativeness are the basic principles of supervised inductive learning.

The focus has long been on problems where the training set Γ_l is sufficiently large for inductive inference to yield suitably accurate results. However, in the last decade, problems arising from areas such as Bioinformatics and the Internet, where the unlabeled data Γ_u is abundant and the labeled data Γ_l relatively expensive and therefore limited, called for different approaches to be developed.

Indeed, in practice there is often a relatively small labeled training set $\Gamma_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$ and a relatively much larger unlabeled data set $\Gamma_u = \{\mathbf{x}_i\}_{i=1}^{U}$. Certainly, the induction principle still holds under its standard assumptions, and standard inductive inference could be applied to Γ_l. However, this would imply that any information contained in Γ_u is neglected. For example, Γ_u may reveal structural properties of the data distribution $F(\mathbf{x})$.

In some cases, the standard i.i.d. assumption underlying inductive inference may also be invalid. For example, the data points in Γ_l may not be representative of the marginal distribution $F(\mathbf{x})$ in many cases, including where labels are more easily obtained for specific subsets of the data space $\mathcal{X}$.

Also more radical departures from the inductive learning assumptions have become more common in practice. For example, it may be that the data points in Γ_l and Γ_u are not randomly sampled, but are simply deterministic sets (e.g. the set of all genes in the genome). This setting has been studied from a statistical learning theory perspective in [4]. Then the problem of predicting the labels for the input data in the unlabeled data set Γ_u is generally known as the problem of *transduction* [2, 5, 6, 7, 8, 9, 1].

2.2 Semi-Supervised Learning as a variant of Unsupervised Learning

So far, we have motivated SSL from the inductive inference or SL perspective, and more specifically from the classification perspective where $\mathcal{Y}$ is a finite set. Semi-supervised learning is then regarded as an adaptation of classification, to allow it to take into account unlabeled data as well. Alternatively, semi-supervised learning may be interpreted as a variation of clustering (or of unsupervised learning in general), where labeled data Γ_l is provided in addition to an unlabeled data set Γ_u to aid the clustering process. Methods motivated in this way are sometimes referred to as *semi-supervised clustering* methods [10].

Many approaches to SSL have been developed starting from this perspective: by taking an UL algorithm and adapting it so that it becomes able to exploit labeled information. This is usually achieved by minimizing a cluster cost function in Γ_u, while imposing the labels provided in Γ_l as constraints. Here, the cluster cost function typically quantifies factors such as the clusters' incoherence, the overlap between clusters, and the proximity of input data belonging to different clusters.

3 Frameworks for SSL algorithm design

Existing SSL algorithms are based on different underlying frameworks. Some are evolutions of other SL or UL methods, but there are also algorithms based on different premises. The sections ahead discuss four common frameworks that have been used in the literature as basis for SSL algorithm design.

3.1 Metric learning

Many SSL approaches consist of two steps. In a first step, the data is embedded in a new metric space, which could be as simple as a linear projection of the original space up to a complex nonlinear embedding. In a subsequent step, an existing method may be applied. This strategy has been attempted in a variety of ways, including by means of kernel design strategies and by means of dimensionality reduction techniques. The supervision by the labels available in the training set may either guide the learning of the new metric or be used by the subsequent algorithm in the second step.

Unsupervised metric learning. Unsupervised metric learning strategies attempt to capture the information in the unlabeled part of the data by means of a form of metric learning. The idea is that it may be possible to embed the data into a new metric space in which any cluster structure in the data is magnified. To this end, ideas from spectral clustering and latent semantic indexing have been used in the literature. Then, in a second step, a standard SL approach may be applied in the new metric space into which the data is embedded. Approaches that may be associated to this category include [11] in this special session and [12, 13].

Supervised metric learning. A complementary approach adopted by some authors is to supervise the metric learning process using the labeled part of the data. This is done in such a way that same-labeled data points are embedded close to each other, and differently labeled data points are at a large distance from each other in the embedding. Then, in a second step, a standard UL approach may be applied in the new metric space into which the data is embedded. This strategy has been developed in e.g. [14, 15, 16].

3.2 Unsupervised methods

Perhaps the most direct approach to developing SSL algorithms is the adaptation of UL (clustering) methods to allow them to exploit the information in the labeled part of the data. This may be done by imposing constraints on the resulting clusters obtained by the algorithm. Such approaches could be called *semi-supervised clustering* techniques.

Existing work along these lines included the adaptation of standard clustering methods such as K-means clustering [17], as well as clustering approaches based on graph cuts and their spectral and convex relaxations [18, 6, 19, 9].

3.3 Supervised methods

The same issue may also be approached from the other end: by adapting SL methods to allow them to exploit information in the unlabeled part of the data. Typically this is done by using the information the unlabeled data reveals about the density of the data. For example, it may be desirable that a classification boundary crosses regions of low density only. These regions could be found based on the unlabeled data.

The first algorithms developed along these lines were able to directly address two-class problems only, e.g. [20, 5, 7, 21]. Many of these are adaptations of the Support Vector Machine classifier, or in terms of the s-t graph mincut problem. More recently, the spectrum of applications has expanded considerably beyond binary classification, now including single and multi-class classification, ranking, regression, and structured output prediction, see e.g. [22, 23, 24] and [25, 26, 27] in this special session.

3.4 Model-based approaches

Supervised as well as unsupervised techniques may often be regarded in terms of inference on probabilistic graphical models. Also, SSL methods can often be interpreted as or be derived directly from these terms. The labels of the unlabeled data are then typically regarded as unobserved variables. Such SSL techniques often involve Expectation Maximization or Variational Inference.

An early example of such an approach is [28], where the data is modeled by means of a Gaussian mixture model. In the SSL setting, only some of the labels are given. The SL extreme where the labels of all data points are specified corresponds to discriminant analysis, while the UL extreme with no labels given corresponds to Gaussian mixture modeling (also known as fuzzy K-means). Another early example where this idea is used for text classification is [29]. More recent approaches in this category include [30] and [31] in this special session.

4 Data representation

SSL techniques also vary depending on the representation of the data they use. We now examine two common data representations.

4.1 Nodes in a graph

Often it is most convenient to specify similarities between data points, which are typically positive real numbers. Such similarities may not be related directly to distances or proximities. When that is the case, SSL problems are usually formalized in graph theoretic terms. The data points are then regarded as vertices in a graph, and vertices are connected by edges weighted by the similarity score of the data points. Examples of such approaches are [21, 18, 6, 32, 19, 9] and [25] in this special session.

A graph representation is convenient as it allows one to rely on a large set of existing algorithms, e.g. to compute graph cuts that are optimal under various criteria (which amounts to partitioning the vertices and hence the data points).

4.2 Points in a vector space

Traditional machine learning and multivariate statistics approaches have more often than not relied on vector space representations of the data points. Additionally, a large number of methods that rely on vector space representations of the data have been rephrased in kernel-based formulations. This allows one to apply such methods to virtually any type of data, as long as a suitable kernel function can be defined.

As expected, many SSL methods are based on either vector representations of the data, or on implicit vector representations induced by a kernel function. Examples include [20, 5, 17, 29, 15, 14, 7, 16, 28, 22, 8, 23, 33] and [27, 34, 26, 11, 31, 30] in this special session.

5 Algorithmic class

A crucial issue in the design of SSL algorithms is their computational efficiency. Many formulations of SSL amount to combinatorial problems that cannot be solved in polynomial time.

5.1 Exact algorithms

Exponential-time. In some of the pioneer approaches such as [20], authors have attempted to find exact solutions of NP-hard formulations of the SSL problem. This inevitably limits the applicability of these methods to small-scale problems. Nevertheless, these initial attempts have been crucial in paving the way for more efficient algorithms and approximations.
Polynomial-time. Some formulations can be solved by means of polynomial-time algorithms, such as [21] and [25] in this special session. While their fast running times is a crucial advantage, it has been argued by some authors that some of these approaches are less robust to certain situations that may occur in the SSL setting (see e.g. [6]). Nevertheless, in certain circumstances their performance is excellent, and the understanding of the applicability of these efficient strategies remains an interesting open problem.
Metric learning based on spectral methods. SSL approaches relying on an initial metric learning step can often be solved exactly using an eigenvalue decomposition (i.e. a spectral method), e.g. [12, 16, 15, 13] and [11] in this special session. While this is a strong advantage of approaches based on metric learning, it should be stressed that the metric learning step is typically only the first of two steps, as explained above. It is unclear to what extent the decomposition of the SSL problem into two steps affects the performance of SSL.
Metric learning based on convex optimization. Other approaches for metric learning rely on convex optimization (see e.g. [35]), and on Semidefinite Programming (SDP) in particular. An early example is [14].

5.2 Relaxations

Certain combinatorial problems can be relaxed to problems that are much easier to solve. Often this is done by allowing the labels to have continuous values rather than discrete class values. Then, after solving the relaxed problem formulation, the continuous approximations of the labels need to be thresholded in order to obtain the label predictions.

Spectral relaxations. Probably the most well-known relaxation techniques transform combinatorial optimization problems in eigenvalue (i.e. spectral) problems (see e.g. [36]). Spectral relaxations have been used in SSL in a number of works, including [18, 6, 19]. However, spectral relaxations, while fast, may be too loose and hence too inaccurate in some cases.
Convex relaxations. Often tighter convex relaxations can be obtained, which can also be solved in polynomial time. SDP relaxations are the most common ones [35, 37, 38]. While relaxation techniques have been around for a while (see e.g. [37] and the discussion of the max-cut problem), in SSL the SDP relaxation approach was first adopted in [7] and later in [22, 8, 9], among others. SDP relaxations seem particularly promising, as they approximate the unrelaxed combinatorial problem remarkably well, while being solvable by off-the-shelf polynomial-complexity interior point solvers. In practice, however, the polynomial complexity still severely rises with the size of the problem, which makes these approaches inadequate for larger problems. It remains an open question whether further developments in convex optimization will eventually make SDP-based methods for SSL amenable to large-scale problems.

5.3 Heuristic approaches

Lastly, well-designed heuristic approaches have been used in a variety of methods. We refer to them as heuristics since they are typically not guaranteed to find the globally optimal solution. Nevertheless, they typically find a solution that is sufficiently close to the optimum, while being more scalable than, for instance, exact methods or methods based on SDP relaxations. Some examples of such methods are based on probabilistic graphical models and also on the EM algorithm or on variational inference, e.g. [17, 23] and [30] in this special session. Other methods are based on techniques such as the Concave-Convex Procedure [33], or on specially designed heuristics, e.g. [5] and [39] in this special session.

6 Conclusions

In recent years, the distinction between SL and UL has gradually faded. This desirable trend contributes to the development of more accurate, more robust methods and improves their applicability to practical scenarios where neither SL nor UL extremes fit well.

In this paper we have attempted to give an overview and propose a taxonomy for some of the existing approaches to SSL. While our list of references is by no means exhaustive and the taxonomy proposed is by no means complete, we hope this paper will help researchers new to this domain get a general understanding of the diversity of approaches as well as the existing challenges in the field of SSL.

Table 1: In this Table we attempt to categorize existing approaches according to the frameworks they rely on (Sec. 3), the representations of data they use (Sec. 4), and their algorithmic class (Sec. 5). The different works are shown grouped by framework.

Reference	Framework	Representation	Algorithm Class
[20]	Supervised	Vector	Exact (exponential-time)
[27]	Supervised	Vector	Convex relaxation
[7]	Supervised	Vector	Convex relaxation
[8]	Supervised	Vector	Convex relaxation
[22]	Supervised	Vector	Convex relaxation
[34]	Supervised	Vector	Heuristic
[26]	Supervised	Vector	Heuristic
[23]	Supervised	Vector	Heuristic
[33]	Supervised	Vector	Heuristic
[5]	Supervised	Vector	Heuristic
[25]	Supervised	Graph	Exact (polynomial-time)
[21]	Supervised	Graph	Exact (polynomial-time)
[17]	Unsupervised	Vector	Heuristic
[6]	Unsupervised	Graph	Spectral relaxation
[19]	Unsupervised	Graph	Spectral relaxation
[18]	Unsupervised	Graph	Spectral relaxation
[9]	Unsupervised	Graph	Convex/spectral relaxation
[32]	Unsupervised	Graph	Convex relaxation
[11]	Metric learning	Vector	Exact (spectral)
[12]	Metric learning	Vector	Exact (spectral)
[16]	Metric learning	Vector	Exact (spectral)
[15]	Metric learning	Vector	Exact (spectral)
[13]	Metric learning	Vector	Exact (spectral)
[14]	Metric learning	Vector	Exact (convex)
[31]	Model-based	Vector	Heuristic
[39]	Model-based	Vector	Heuristic
[30]	Model-based	Vector	Heuristic (variational)
[29]	Model-based	Vector	Heuristic (EM)
[28]	Model-based	Vector	Heuristic (EM)

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning.* MIT Press, Cambridge, MA, 2006.

[2] V. N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, New York, 2nd edition, 1999.

[3] Ryszard S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20(2):111–161, 1983.

[4] P. Derbeko, R. El-Yaniv, and R. Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[5] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of the International Conference on Machine Learning (ICML99)*, pages 200–209, 1999.

[6] T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. of the International Conference on Machine Learning (ICML03)*, pages 290–297, 2003.

[7] T. De Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems 16 (NIPS03)*, pages 73–80, 2004.

[8] T. De Bie and N. Cristianini. Semi-supervised learning using semi-definite programming. In O. Chapelle, B. Schoëlkopf, and A. Zien, editors, *Semi-supervised learning.* MIT Press, Cambridge-Massachussets, 2006.

[9] T. De Bie and N. Cristianini. Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *Journal of Machine Learning Research*, 7:1409–1436, 2006.

[10] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *ICML*, pages 27–34, 2002.

[11] David Tomas and Claudio Giuliano. A semi-supervised approach to question classification. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[12] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems 15 (NIPS02)*, pages 585–592, 2003.

[13] J. Weston, C. Leslie, D. Zhou, A. Elisseeff, and W. Noble. Semi-supervised protein classification using cluster kernels. In *Advances in Neural Information Processing Systems 16 (NIPS03)*, pages 595–602, 2004.

[14] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15 (NIPS02)*, pages 505–512, 2003.

[15] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *Proc. of the 7th European Conference of Computer Vision (ECCV02)*, pages 776–792, May, 2002.

[16] T. De Bie, M. Momma, and N. Cristianini. Efficiently learning the metric with side-information. In *Proc. of the 14th International Conference on Algorithmic Learning Theory (ALT03)*, pages 175–189, 2003.

[17] P. Bradley, K. Bennett, and Ayhan Demiriz. Constrained K-means clustering. Technical Report MSR-TR-2000-65, Microsoft Research, 2000.

[18] Stella X. Yu and Jianbo Shi. Grouping with bias. In *Advances in Neural Information Processing Systems 14 (NIPS01)*, pages 1327–1334, 2002.

[19] T. De Bie, J. Suykens, and De Moor B. Learning from general label constraints. In *Proc. of IAPR International Workshop on Statistical Pattern Recognition (SPR04)*, pages 671–679. 2004.

[20] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11 (NIPS98)*, pages 368–374, 1999.

[21] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. of the 18th International Conf. on Machine Learning (ICML01)*, pages 19–26, 2001.

[22] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *The Twentieth National Conference on Artificial Intelligence (AAAI05)*, 2005.

[23] U. Brefeld and T. Scheffer. Semi-supervised learning for structured output variables. In *Proc. of the 23th International Conf. on Machine Learning (ICML06)*, 2006.

[24] C. Cortes and M. Mohri. On transductive regression. In *Advances in Neural Information Processing Systems (NIPS) 19*, 2006.

[25] Kristiaan Pelckmans and Johan Suykens. Transductively learning from positive examples only. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[26] Tuong Vinh Truong, Massih-Rez Amini, and Patrick Gallinari. A self-training method for learning to rank with unlabeled data. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[27] Liva Ralaivola. Semi-supervised bipartite ranking with the normalized rayleigh coefficient. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[28] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with EM using equivalence constraints. In *Advances in Neural Information Processing Systems 16 (NIPS03)*, pages 465–472, 2004.

[29] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[30] Peng Li, Yiming Ying, and Colin Campbell. A variational approach to semi-supervised clustering. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[31] Charles Bouveyron, Stephane Girard, and Madalina Olteanu. Supervised classification of categorical data with uncertain labels for dna barcoding. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[32] E.P. Xing and M.I Jordan. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical Report CSD-03-1265, Division of Computer Science, University of California, Berkeley, 2003.

[33] R. Collobert, J. Weston, and L. Bottou. Trading convexity for scalability. In *Proc. of the 23th International Conf. on Machine Learning (ICML06)*, 2006.

[34] Ruy Luiz Milidu and Julio Cesar Duarte. Improving BAS committee performance with a semi-supervised approach. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.

[35] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, U.K., 2003.

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

[37] C. Helmberg. Semidefinite programming for combinatorial optimization. Habilitationsschrift ZIB-Report ZR-00-34, TU Berlin, Konrad-Zuse-Zentrum Berlin, 2000.

[38] Tijl De Bie. Deploying SDP for machine learning. In *Proc. of the 15th European Symposium on Artificial Neural Networks (ESANN08)*, 2008.

[39] Etienne Come, Latifa Oukhellou, Patrice Aknin, and Thierry Denoeux. Partially-supervised learning in independent factor analysis. In *Proc. of the 16th European Symposium on Artificial Neural Networks (ESANN09)*, 2009.