Meta-prediction of semi-naive Bayesian network classifiers based on dataset complexity characterization

M. Julia Flores, José A. Gámez, Ana M. Martínez
Universidad de Castilla-La Mancha, Spain
{julia.flores,jose.gamez,anamaria.martinez}@uclm.es

Abstract

Ever since naive Bayes was proposed, many have been the attempts to try to alleviate its naive assumption to obtain better accuracy records, without further increasing its complexity. In this line we can find a group of Bayesian network classifiers that either do not perform structural search or it is very simple, known as the family of semi-naive Bayesian network classifiers (BNCs). Given a particular dataset, it would be desirable, based on the characteristics presented, to find out which semi-naive BNCs obtains the best possible expected prediction, since estimations based solely on expected accuracy on training can be misleading. In this paper we propose an automatic procedure to carry out this meta-prediction process, based on the values of several data complexity measures for supervised classification. We resort to multi-label classification to test this procedure, obtaining promising results.

1 Introduction

Learning the structure of a network can take a long time and effort, especially for datasets of high dimensionality. That is why it is often convenient to consider a partially or totally pre-fixed structure from which the conditional probability distributions are learnt. The most simple of these structures is the one used by naive Bayes (NB), that assumes all the attributes are independent given the class. In spite of its naive assumption, it performs surprisingly well in some domains. Many techniques improve the accuracy of NB by alleviating the attribute interdependence problem, such as Averaged One-Dependence Estimators (AODE), Tree augmented naive Bayes (TAN) or k-dependence Bayesian classifier (KDB). All these techniques, known as semi-naive BNCs, do not perform structural search or this search is very simple (Flores et al., 2012).

It is still unclear, at the moment of writing, which of these classifiers should be used for a particular dataset. There have been several studies oriented to compare classifiers of different nature. In Ho and Basu (2002), a selection of several measures for characterizing the complexity of classification problems is presented, along with an empirical study on the distribution of real world problems compared to random noise, indicating that it is possible to find learnable structures with the geometrical measures presented. These measures indicate the overlap of individual feature values; the separability of classes; and geometry, topology and density of manifolds. This group of measures encounters its natural definition in the two-class domain. Nevertheless, attempts to generalize some of these measures to the multi-class domain can be found in Mollineda et al. (2005) and more recently in Orriols-Puig et al. (2010).

Numerous studies have followed that try to obtain the domains of competence for one or more particular classifiers, by studying error rate patterns with respect to individual or combination of complexity measures (CMs), usually bivariate combinations. Some of these works are Bernadó-Mansilla and Ho (2004) for 1-nearest-neighbour (1NN), linear classification, decision trees and decision forests; Sánchez et al. (2007) for kNN; and more

recently, Luengo and Herrera (2010) and Luengo and Herrera (2012) for fuzzy rule based classification systems, or artificial neural networks and support vector machines.

Note that none of these studies focuses on the family of semi-naive BNCS. As a different and more practical approach, we propose an automatic mechanism to select the most promising semi-naive BNC for a particular dataset, based on the values of some of the CMs. An interesting work in relation to this topic is presented in Hernández-Reyes et al. (2005), where an automatic classifier selection based on data complexity measures is proposed. Their method describes problems with complexity measures and labels them with the classifier that gets the best accuracy among a set of five classifiers: kNN, NB, linear regression, RBFNetwork and J48. We argue that simply selecting the classifier that obtains the highest accuracy is inaccurate, since several classifiers can be equally good for a given problem. We present an alternative procedure based on *partial* multi-label classification, where the term partial refers to the use of the multi-label paradigm for the training phase exclusively, and empirically test this procedure.

The paper is divided as follows: Section 2 introduces the semi-naive BNC considered. In Section 3 the existing data complexity measures for supervised classification are presented. Section 4 introduces the multi-label classification problem. Section 5 outlines our proposal for automatic meta-prediction of the semi-naive BNCs. The empirical results obtained are presented in Section 6. Section 7 includes the main conclusions and future work.

2 Semi-naive Bayesian network classifiers

Apart from NB, the following semi-naive BNCs will be considered:

AODE: (Webb et al., 2005) is an ensemble of n models (where n is the number of attributes) in which every attribute depends on the class and another shared attribute (at model i, A_i), designated as superparent. AODE is an attractive alternative to other approaches that aim to improve NB maintaining its efficiency, as it provides competitive error rates with an efficient profile (Zheng and Webb, 2005).

HODE (Flores et al., 2009) estimates a new hidden variable using the EM algorithm (Dempster et al., 1977), whose main objective is to model the meaningful dependences between each attribute and the rest of the attributes that AODE takes into account. HODE can be considered an attractive alternative to AODE, especially in datasets where the number of attributes or number of values per attributes is very large.

TAN: (Friedman et al., 1997) learns a maximum weighted spanning tree based on the conditional mutual information between two attributes given the class label. Then, the arcs in the tree are oriented by choosing a root and the model is completed by adding a link from the class to each attribute.

KDB: Sahami (1996) introduced the notion of k-dependence estimators, from which the probability of each attribute value is conditioned by the class and, at most, k other attributes. Throughout the KDB algorithm it is possible to construct classifiers across the whole spectrum, from the NB structure to the full BN structure, by varying the value of k, i.e. the maximum number of parents that every attribute can have.

3 Data complexity measures

The complexity measures originally proposed in Ho and Basu (2002) have already shown their power for characterizing classifiers of different nature, although mainly on continuous datasets. Some of the CMs have been originally defined for numeric values. Since the natural domain of the Bayesian network classifiers is with discrete variables, nominal attributes will be mapped into integer numbers for the calculations of these measures, assuming though a non-existent order between the labels.

These complexity measures are summarized in Table 1, where as a novelty, a new column is added to indicate the tendency that a particular measure may follow according to its definition,

Type	Identifier	Name/Description	Simpler if...
Overlaps in the feature values from different classes	F1	Maximum Fisher's discriminant ratio	+
	F1v	Directional-vector maximum Fisher's discriminant ratio	+
	F2	Overlap of the per-class bounding boxes	−
	F3	Maximum (individual) feature efficiency	+
	F4	Collective feature efficiency	+
Measures of class separability	L1	Minimized sum of the error distance of a linear classifier	−
	L2	Training error of a linear classifier	−
	N1	Fraction of points on the class boundary	−
	N2	Ratio of average intra/inter class nearest neighbor distance	−
	N3	Leave-one-out error rate of the one-nearest neighbor classifier	−
Measures of geometry, topology, and density of manifolds	L3	Nonlinearity of a linear classifier	−
	N4	Nonlinearity of the one-nearest neighbor classifier	−
	T1	Fraction of maximum covering spheres	−
	T2	Average number of points per dimension	+

Table 1: Summary of CMs for supervised classification.

in order to reflect more simplicity on the problem it applies. Note that the symbol + indicates more simplicity as the value of the corresponding complexity measure increases, and − as it decreases.

4 Multi-label classification

Multi-label classification is a type of classification problem where multiple class labels can be assigned to each example[1].

A comprehensive overview on multi-label classification can be found in Tsoumakas and Katakis (2007). Numerous multi-label classification techniques are gathered, mainly divided into problem transformation and algorithm adaptation methods. The first group of methods transform the learning task into one or more single-label classification tasks, whereas the second group extend specific learning algorithms in order to handle multi-label data directly.

We are using two classifiers that belong to the category of problem transformation methods: NB using the binary relevance (NB-BR) transformation method; and the RAndom k-labELsets method (RAkEL). BR is one of the simplest and most popular transformation methods; where L single-label classifiers (L being the number of class labels) are learned from the L class-binary datasets created. RAkEL uses an ensemble of label powerset (LP) transformation methods. LP considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classifier.

5 Meta-classification of semi-naive BNCs

The main objective is that, given a particular dataset to be classified, it is possible to predict which semi-naive BNC is more likely to provide the most accurate predictions. For this purpose, it is necessary to create what we call a *training meta-dataset*: where every instance represents a single dataset, for which the predictive attributes correspond, in principle, to the 14 complexity measures in Table 1. This is in concordance with the method proposed in Hernández-Reyes et al. (2005). One of the main differences between their approach and ours is that they consider a single class label dataset, assigning to every instance the classifier with the lowest error for the dataset that represents that instance.

In our case, 5 semi-naive BNCs for discrete attributes are considered, in particular: NB, AODE, HODE, TAN and KDB3[2]. We believe that the selection of a single classifier based directly on the lowest error value can be too arbitrary. Alternatively, we propose to carry out statistical tests on the classifiers' results for each problem, in order to keep the best classifier and also those whose error rates are not significantly

[1] Also known as multi-dimensional classification (Bielza et al., 2011). Note, in any case, the difference with a multi-class problem, that simply refers to the existence of one class with more than two labels.

[2] KDB with $k = 3$ has been selected for these experiments in order to gain some variety among the classifiers considered.

Table 2: Main characteristics of the 26 numeric datasets: number of predictive variables (n), number of classes (c) and number of instances (m).

Id	Datasets	n	c	m	Id	Datasets	n	c	m
1	balance-scale	4	3	625	14	mfeat-fourier	76	10	2000
2	breast-w	9	2	699	15	mfeat-karh	64	10	2000
3	diabetes	8	2	768	16	mfeat-morph	6	10	2000
4	ecoli	7	8	336	17	mfeat-zernike	47	10	2000
5	glass	9	7	214	18	optdigits	64	9	5620
6	hayes-roth	4	4	160	19	page-blocks	10	5	5473
7	heart-statlog	13	2	270	20	pendigits	16	9	10992
8	ionosphere	34	2	351	21	segment	19	7	2310
9	iris	4	3	150	22	sonar	60	2	208
10	kdd-JapanV	14	9	9961	23	spambase	57	2	4601
11	letter	16	26	20000	24	vehicle	18	4	946
12	liver-disorders	6	2	345	25	waveform-5000	40	3	5000
13	mfeat-factors	216	10	2000	26	wine	13	3	178

worse. Given that the considered classifiers belong to the same family, it is reasonable to expect small differences.

This then requires to resort to multi-label classification in order to handle the existence of multiple class labels. 5x2cv is used for the evaluation process, and the 5x2cv F Test defined by Alpaydin (1999) has been used to select the semi-naive BNCs for each dataset (unilateral comparisons). The level of significance has been fixed at 5% ($\alpha = 0.05$).

In order to construct the meta-dataset with the complexity measure values from different datasets, we select the group of 26 numeric datasets in Table 2. Since most of the CMs are only defined to deal with datasets with two class labels, we have created several binary datasets from each dataset with more than 2 class labels, specifically, as many as the total number of class labels per dataset (by following the strategy known as one-against-all). Hence, there is a total of 157 datasets with two class labels. Additionally, we discretize them applying unsupervised equal frequency discretization with 5 bins. Motivated by the work in Flores et al. (2011) we believe that the choice of the discretization technique is not decisive in this context.

A small sample of the resulting training meta-dataset is shown in Table 3. Every example corresponds to the result of the 14 complexity measures for a specific dataset, whereas the class

Examples: 157		**Labels:**	5 (binary)
Predictive attributes: 14 (numeric)			
Distinct Labelsets: 24			
Cardinality:	2.52	**Density:**	0.50
***Percentage of examples with label:**			
1(NB):	19.74%	4(TAN):	52.23%
2(AODE):	56.59%	5(KDB3):	61.78%
3(HODE):	61.15%		
***Examples of cardinality:**			
0:	0	3:	43
1:	39	4:	19
2:	43	5:	13

Table 4: Statistics of the meta-dataset created.

labels are binary and correspond to the 5 following semi-naive BNCs: NB, AODE, HODE, TAN and KDB3: a bit equal to 1 for the j^{th} class label on the i^{th} instance indicates that the classifier on j, either obtain the best error rate for the dataset on the i^{th} position or it is not significantly worse that the classifier that does.

The statistics of the resulting meta-dataset are summarized in Table 4. The upper part of the table (above the horizontal line) displays general information: such as the number of examples, attributes or class labels; whereas the lower part includes more specific information related to the class labels. The number of distinct labelsets indicates the binary combinations out of the 2^5 possible. The value for cardinality is calculated as the ratio of positive bits (those equal to 1) in the labels over the total number of instances. The density is just the division of the cardinality between the number of labels. These values indicate that, on average, every example can be optimally classified by at least 2 semi-naive BNCs. Two figures to be highlighted here are: the number of examples of cardinality equal to 1, i.e., those that only have one "best" classifier, which is 39; and the number of trivial examples (cardinality equal to 5), i.e., those for which any classifier would be equally eligible, which is 13.

For our experiments, we select the following two approaches to carry out the meta-classification task:

- **NB-BR**: (Tsoumakas and Katakis, 2007) In our case, we transform the original dataset into 5 datasets with binomial class,

Table 3: Sample of the meta-dataset created to predict the best semi-naive BNC based on data complexity measures.

dataset	F1	F1v	F2	F3	F4	L1	L2	L3	N1	N2	N3	N4	T1	T2	NB	AODE	HODE	TAN	KDB3
a1iris.2c0	0.474	16.455	0.000	0.807	1.000	0.333	0.007	0.000	0.067	0.209	0.013	0.263	1	37.5	0	1	0	0	0
a1iris.2c1	0.458	2.374	0.250	0.433	0.573	0.658	0.333	0.500	0.200	0.264	0.093	0.250	1	37.5	0	0	1	0	0
a1iris.2c2	0.475	10.186	0.062	0.627	0.760	0.461	0.053	0.017	0.160	0.244	0.093	0.290	1	37.5	1	1	1	0	1
⋮			⋮			⋮			⋮			⋮			⋮					⋮

each one containing the corresponding column (NB,AODE,...) as class. For the classification of a new instance, the original definition of BR would output the union of the labels that are positively predicted by the 5 classifiers. In our experiments, only the most likely label is considered.

- **RA*k*EL**: (Tsoumakas and Vlahavas, 2007), is a random k-labelset method that constructs an ensemble of LP classifiers (of type C4.5 in our experiments), where each LP is trained using a different small random subset of size k of the set labels. A ranking of the labels is produced by averaging the zero-one predictions of each model per considered label. Thresholding is used to produce a bipartition as well. Only the first label in the ranking will be considered in our case.

However, all of these strategies consider a multi-label prediction phase as well. For our purposes, even though we are training a classifier based on a multi-label paradigm, we select only the best classifier to predict with. This restriction requires to redefine the way in which the evaluation is performed, so that a specific prediction, Z_e for the example (e, Y_e), is considered successful if the label predicted is among those included in Y_e, where both Z_e and Y_e are binary vectors of length L, L being the number of semi-naive BNCs considered, i.e., the number of labels. However, given that we have modified the output such that the number of positive values in Z_e is only one, we can use the original definition of **example-based precision** (Tsoumakas et al., 2010) as evaluation measure:

$$Precision = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Z_i|}, \qquad (1)$$

where the operator $|.|$ indicates the cardinality of the positive bits. Consider, for example, an output $Y_e = \{0, 1, 1, 0, 0\}$, which indicates that both AODE and HODE provide the best results for a particular dataset e, i.e., one of them has the highest absolute value and the other is not significantly worse. Then, if the meta-classifier, let say NB with BR (NB-BR), provides the output $Z_e = \{0, 1, 0, 0, 0\}$, this example would contribute to the summation as 1. If the output provided by RA***k***EL were $Z_e = \{0, 0, 0, 0, 1\}$ instead, the contribution would be equal to 0.

Note that since the average number of "valid" labels for every instance is equal to 2.5, our classification problem can be considered to be of equivalent difficulty to a binary class problem, since there is a 50% of probability to be accurate when classifying.

Additionally, it may not be necessary to use all the complexity measures as predictive attributes, as some of them can be redundant, irrelevant and maybe the "intrinsic" dimensionality may be smaller than the total number of measures considered. To that aim, we find a large amount of literature for feature selection (FS) (Guyon and Elisseeff, 2003). Note that FS is not required here for dimensionality reduction with efficiency purposes, but it could be beneficial to remove measures that are too similar in the meta-dataset, and hence, redundant.

The whole process is outlined in Figure 1. The left-hand side displays the three steps involved in the meta-dataset formation, which entails the most time consuming part of the process. The steps required when a new dataset faces a classification process, is included in the

dashed line. Given a new dataset, the values for the CMs considered in the meta-classification process will be calculated (not necessarily all of them, as shown in Section 6). From these values the meta-classifier selected will return the best semi-naive BNC.

6 Experimental methodology and results

We have resorted to a Java library for multi-label learning, called Mulan (Tsoumakas et al., 2011), in order to handle the multiple labels. The two meta-classifiers selected to work with the meta-dataset created are NB with BR and RA***k***EL, described above. The calculations of the different measures have been obtained with the data complexity library in C++ (Orriols-Puig et al., 2010).

10-fold cross-validation has been used for evaluation. The selection of these two multi-label classifiers have been motivated by the results obtained with the different algorithms provided by Mulan. Other paradigms have been tested, such as a lazy learning approach based on kNN, ML-KNN (Zhang and Zhou, 2007); and transformation methods, such as classifier chains (Read et al., 2011) with different base classifiers. Even though this study is not an exhaustive one, since it does not cover all the multi-label classifiers in the existing literature, for instance (Bielza et al., 2011), we believe that it is sufficient for our purposes.

In Table 5, different results in terms of example-based precision are shown. The alternatives tested are as follows:

- The first column, `Data`, indicates whether the data considered are directly the value of the measures for the different datasets (Original) or the data have been transformed through principal component analysis techniques (PCA). Dimensionality reduction is accomplished by choosing enough eigenvectors to account for a percentage of the variance in the original data, which has been set to 95%[3] (PC space). Furthermore, the PC space data have been transformed back to the original space eliminating some of the worst eigenvectors, with the aim of filtering attribute noise (PC space transformed back to original space).

- Feature selection through clustering techniques has also been carried out in some cases, as indicated in the second column, `Clustering+FS`. More specifically, a k-means clustering algorithm (using Euclidean distance) is performed in the transposed dataset with[4] $k = 10$. The output indicates the following clusters: $(L1, N2)$, $(L2, L3, N3)$, $(F1, N1)$ and the rest of the measures in isolation. In order to select which measures to keep from each cluster we use PCA techniques again. The procedure is as follows: data are transformed through the PC space and back to the original space. As only the best PCs are retained, by setting the variance covered equal to 95%, we will obtain a dataset in the original space but with less attribute noise as above. Hence, the ranking obtained by this method is:

$$F4, L2, L1, F1v, F1, F3, F2, N4, T2, T1, N1, L3, N3, N2$$

  We maintain then: $L1$ from cluster $(L1, N2)$, $L2$ from $(L2, L3, N3)$ and $F1$ from $(F1, N1)$, along with the rest of the measures. And we will discard N2, L3, N3 and N1, which happen to be the last four attributes given by PCA.

- Two multi-label classifiers (BR-NB and RAkEL) are directly applied or after performing FS as explained above. Indicated in the third column, `Meta-Classifier`.

The results on the last column in Table 5, show a variety of accuracy values ranging from

[3] Default value in WEKA.

[4] Note that in this case, the purpose of feature selection is mainly carried out in order to remove possible noisy features, that is why we consider appropriate (although arbitrarily) not to remove more than 4 predictive attributes.

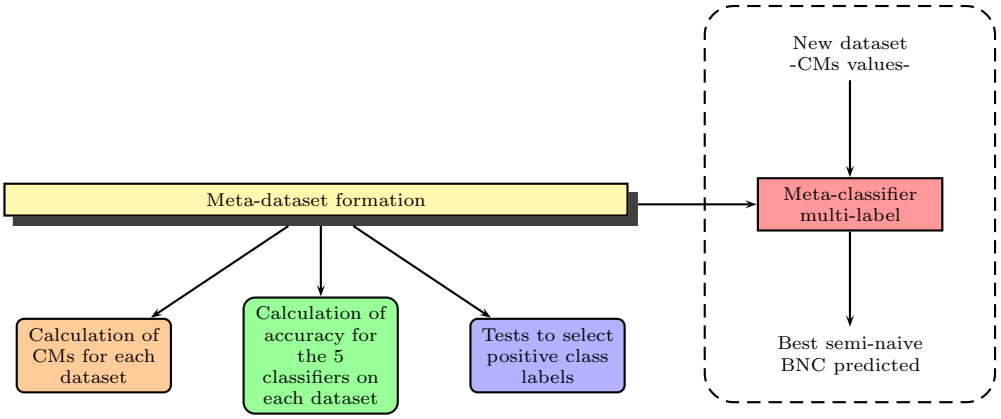

Figure 1: Schema of the meta-classification process.

Table 5: Expected example-based precision for meta-classifier selection.

Data	Clustering+FS	Meta-Classifier	Precision ± Stand. dev.
Original		BR-NB	84.79±7.40
		RAkEL	86.75±7.59
	K-means+PCA	BR-NB	86.08±5.32
	K-means+PCA	RAkEL	86.04±7.30
PC space		BR-NB	85.46±10.4
		RAkEL	77.67±8.61
PC space transformed back to original space		BR-NB	86.08±6.63
		RAkEL	86.71±5.8
	K-means+PCA	BR-NB	86.08±6.01
	K-means+PCA	RAkEL	**87.38±7.81**

77.7% to 87.4% depending on the data considered, the use or not of pre-processing techniques for FS and the multi-label classifier applied. It is obvious that the options and combinations here to test with are massive, and it is not our aim to perform an exhaustive study. The main purpose of this small comparison is to give an idea of the predictive power of the model.

All in all, the results seem to be encouraging, since in the best case, they offer a precision estimated in 87.38% of predicting correctly one of the best semi-naive BNCs, based on the complexity measures of a particular dataset with discrete attributes.

7 Conclusions

This paper presents an automatic procedure to advise on the best semi-naive BNC to use for classification. A meta-data set is created with the values of different CMs as predictive attributes. Then, a multi-label classifier is trained and afterwards used to predict a single best semi-naive BNC, given the values of a subgroup of CMs of the target dataset.

This procedure has been tested to select among 5 semi-naive BNCs. A promising estimated predictive accuracy of 87.38% has been obtained with the RAkEL multi-label classifier, and after preprocessing the meta-dataset created.

As future work, the test bed of considered datasets can be extended incorporating the datasets from the Landscape contest (Macià et al., 2010), that has been created to cover a wider range of the complexity measurement space.

Acknowledgments

This work has been financially supported by the FPU grant with reference number AP2007-02736, and it has also been partially funded by FEDER funds and the Spanish Government (MICINN) through project TIN2010-20900-C04-03.

References

E. Alpaydin. 1999. Combined 5 x 2 cv F test for comparing supervised classification learning algorithms. *Neural Computation*, 11(8):1885–1892.

E. Bernadó-Mansilla and T. K. Ho. 2004. On classifier domains of competence. In *Proc. of ICPR'04, Volume 1 - Volume 01*, pages 136–139.

C. Bielza, G. Li, and P. Larrañaga. 2011. Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, September.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38.

M. J. Flores, J. A. Gámez, A. M. Martínez, and J. M. Puerta. 2009. HODE: Hidden One-Dependence Estimator. In *Proc. of ECSQARU '09*, pages 481–492. Springer-Verlag.

M. J. Flores, J. A. Gámez, A. M. Martínez, and J. M. Puerta. 2011. Handling numeric attributes when comparing Bayesian network classifiers: does the discretization method matter? *Applied Intelligence*, 34:372–385, June.

M. J. Flores, Gámez J. A., and Martínez A. M. 2012. Intelligent data analysis for real-life applications: Theory and practice. chapter Supervised classification with Bayesian networks: A review on models and applications, pages 72–102. IGI Global.

N. Friedman, D. Geiger, and M. Goldszmidt. 1997. Bayesian network classifiers. *Machine Learning*, 29:131–163.

I. Guyon and A. Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March.

E. Hernández-Reyes, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad. 2005. Classifier selection based on data complexity measures. In *Proc. of CIARP'05*, pages 586–592. Springer-Verlag.

T. K. Ho and M. Basu. 2002. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:289–300.

J. Luengo and F. Herrera. 2010. Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method. *Fuzzy Sets and Systems*, 161:3–19.

J. Luengo and F. Herrera. 2012. Shared domains of competence of approximate learning models using measures of separability of classes. *Information Sciences*, 185(1):43–65, February.

N. Macià, T. K. Ho, A. Orriols-Puig, and E. Bernadó-Mansilla. 2010. The landscape contest at ICPR 2010. In *Proc. of ICPR'10*, pages 29–45. Springer-Verlag.

R. A. Mollineda, J. S. Sánchez, and J. M. Sotoca. 2005. Data characterization for effective prototype selection. In *Proc. of IbPRIA*, pages 27–34. Springer.

A. Orriols-Puig, N. Macià, and T. K. Ho. 2010. Documentation for the data complexity library in C++. Technical report, La Salle - Universitat Ramon Llull.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359.

M. Sahami. 1996. Learning limited dependence Bayesian classifiers. In *Proc. of KDD'96*, pages 335–338.

J. S. Sánchez, R. A. Mollineda, and J. M. Sotoca. 2007. An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Analysis and Applications*, 10:189–201, July.

G. Tsoumakas and I. Katakis. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13.

G. Tsoumakas and I. Vlahavas. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *Proc. of ECML '07*, pages 406–417. Springer-Verlag.

G. Tsoumakas, I. Katakis, and I. Vlahavas. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685.

G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. 2011. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414.

G. I. Webb, J. R. Boughton, and Z. Wang. 2005. Not so naive Bayes: Aggregating One-Dependence Estimators. *Machine Learning*, 58(1):5–24.

M.-L. Zhang and Z.-H. Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048.

F. Zheng and G. I. Webb. 2005. A comparative study of semi-naive Bayes methods in classification learning. In *Proc. of AusDM*, pages 141–156.