Multi-label Classification using Conditional Dependency Networks

Yuhong Guo and **Suicheng Gu**
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
yuhong@temple.edu

Abstract

In this paper, we tackle the challenges of multi-label classification by developing a general conditional dependency network model. The proposed model is a cyclic directed graphical model, which provides an intuitive representation for the dependencies among multiple label variables, and a well integrated framework for efficient model training using binary classifiers and label predictions using Gibbs sampling inference. Our experiments show the proposed conditional model can effectively exploit the label dependency to improve multi-label classification performance.

1 Introduction

Multi-label classification is a challenging problem in many real-world application domains, where each instance can be assigned simultaneously into multiple classes. Typical application problems include text categorization where one document can belong to multiple categories, bio-informatics where one protein may have multiple functions, etc.

Traditional two-class and multi-class problems can be viewed as special cases of multi-label classification, where each instance has only one label. A multi-label problem can also be cast as a multi-class problem by considering all possible combinations of the original classes. However, this will substantially increase the class number and increase the computational complexity of the problem. Many proposed methods tackle multi-label problems by first transforming a multi-label problem into a set of independent binary classification problems, then employing ranking or thresholding schemes for the overall multi-label classification. An obvious drawback of such methods is that they completely ignore the inter-dependencies among multiple labels. In many applications, strong co-occurrences and interdependencies exist among multiple class labels. For example, an article on the topic of *religion* is likely to talk about *culture* as well, but unlikely to talk about *football*. Capturing the dependencies among class labels during classification is thus expected to lead to improved classification performance. Many methods with this motivation have been proposed in the literature, some of which exploit graphical models to capture the label dependencies and conduct structured classification, including those using Bayesian networks [de Waal and van der Gaag, 2007; Rodriguez and Lozano, 2008; Bielza *et al.*, 2011] and conditional random fields [Ghamrawi and McCallum, 2005]. However, these approaches require much more complicated learning and prediction phases than binary classification models.

In this paper, we propose a novel multi-label classification approach based on a conditional cyclic directed graphical model, which we name as conditional dependency networks. In particular, we construct a fully connected dependency network on the class label variables, where each variable is dependent on all the other class variables and the input feature variables. By doing so, we circumvent the challenging structure learning issue associated with the Bayesian network based methods. The conditional distribution associated with each label node in our model corresponds to one binary classification model. Thus our model parameters can be learned by training k binary classifiers, where k is the number of classes. The conditional dependency network model we employ is a more natural representation of label co-occurrence dependencies than acyclic Bayesian networks, while its learning process is much simpler than those used in conditional random fields, where inferences are typically involved. Our empirical results suggest that the proposed model is effective in exploiting label dependencies to improve classification performance, and demonstrates superior performance over a few multi-label classification methods developed in the literature.

2 Dependency Networks

Graphical models have been used in many domains to represent a joint distribution over a set of random variables. They are natural ways to model the independencies/dependencies among variables. Two types of commonly used graphical models are Bayesian networks and Markov networks (Markov random fields). A Bayesian network is a directed acyclic graphical model, where each node represents one variable and the directed edges usually represent the ordered probabilistic dependencies between variables. The parameters in a Bayesian network typically encode the local conditional probability distributions on each variable given its parents. It is NP-hard to identify the optimal Bayesian network structure [Chickering *et al.*, 1994], but allows a closed-form maximum likelihood solution for the parameters given the structure. A Markov network is an undirected graphical model, where the undirected edges encode the depen-

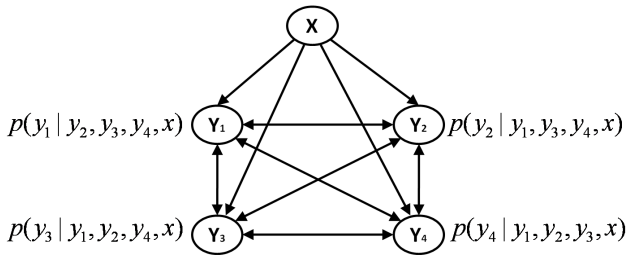

Figure 1: Conditional dependency network model.

dencies among the variables. Markov networks are more suitable to capture undirected correlations and interactions among variables. However, the learning problem associated with a Markov network is much more challenging than their counterparts in a Bayesian network: inference is usually required for parameter learning, and the general structure learning remains to be an NP-hard problem due to the difficulty of parameter estimations of the network (c.f. [Srebro, 2001]).

Dependency networks [Heckerman *et al.*, 2000] are cyclic directed graphical models, where the parents of each variable are its Markov blanket. Similar to Bayesian networks, the edges in a dependency network are directed. However, unlike Bayesian networks, the directed edges of dependency networks encode not ordered relationships but directed dependencies among variables. Actually, the independencies in a dependency network are exact to those of a Markov network with the same adjacencies. Moreover, the primary difference between dependency networks, Bayesian networks, and Markov networks is that dependency networks approximate the joint distribution over a set of random variables with a set of local conditional probability distributions that are learned independently. Thus a dependency network has the advantage of Markov networks in encoding flexible correlational interdependence relationships, while possessing the simple independent parameterization of Bayesian networks in terms of local conditional probability distributions. Dependency networks are not guaranteed to specify a consistent joint distribution, and thus exact inference techniques are not applicable. Nevertheless, Gibbs sampling inference techniques [Neal, 1993] can still be used to efficiently recover a reasonable full joint distribution.

Given a set of random variables $\mathbf{X} = \{X_1, \cdots, X_n\}$ with a joint distribution $p(\mathbf{x})$, a dependency network is a directed graph $G = (V, E)$ with a set of conditional probability distributions $\mathcal{P} = \{p(x_i|pa_i), \forall i\}$. Each variable X_i corresponds to one node $v_i \in V$. The parents of X_i, denoted as $\mathbf{Pa_i}$, are the set of nodes v_j such that $(v_j, v_i) \in E$, where E denotes the set of directed edges. Due to their natural correlational dependency representations, independent parameter estimates on each variable, and simple inference procedures, dependency networks can be applied in many tasks such as probabilistic inference, collaborative filtering, data visualization and relational learning.

In this paper, we extend dependency networks into general conditional dependency networks to tackle multi-label classification problems. In the proposed network, the discrete class label variables $\mathbf{Y}$ are interdependent on each other in a dependency network, conditioning on the observation features $\mathbf{X}$. The conditional probability distributions associated with each variable Y_i are general probabilistic prediction functions. Comparing to conditional Markov networks, the proposed model maintains the same advantages of dependency networks over Markov networks in the non-conditional case.

3 Multi-label Classification Model

Given a set of multi-label training instances $D = \{(\mathbf{x}^\ell, y_1^\ell, \cdots, y_k^\ell)\}_{\ell=1}^t$, where each y_i^ℓ is a $\{-1, +1\}$-valued class label, we aim to learn a multi-label predictor $f : \mathbf{x} \longmapsto [y_1, \cdots, y_k]$ to produce good classifications on incoming test instances $D' = \{\mathbf{x}^1, \cdots, \mathbf{x}^m\}$. Our intuition is to exploit interdependencies among label variables using natural graphical model representations. In particular, we present a conditional dependency network to model the interdependencies among multiple label variables. The proposed model allows a simple learning procedure by training k binary classifiers, where k is the number of classes, and a Gibbs sampling inference technique to predict labels for test instances.

3.1 Conditional Dependency Networks

Given observation features $\mathbf{x}$, we propose to model the conditional joint distribution over label variables $\mathbf{Y} = \{Y_1, \cdots, Y_k\}$ using dependency networks, where each Y_i is a binary variable with values from $\{-1, 1\}$. We aim to capture the label interdependency with such a model to improve classification performance. Since there are usually no particular influence directions among the label variables, we build a fully connected dependency network over the $\mathbf{Y}$ variables. That is, there is a bidirectional edge between each pair of variables, (Y_i, Y_j). Figure 1 shows an example of conditional dependency networks with four class variables. Assuming a fully connected structure, we can avoid the computational expensive step of identifying conditional optimal structures while still maintaining a simple parameter learning phase and an approximated inference procedure for non-singly connected structures. In this conditional dependency network, the strength of label interdependency and the power of prediction from the features to the labels are encoded in the model parameters, i.e., the conditional probability distributions (CPDs) associated with each variable node Y_i, given all its parents and the observation $\mathbf{X}$.

When ignoring the observations $\mathbf{X}$, the CPDs on each Y_i variable can be determined by closed-form solutions based on the sufficient statistics among the $\mathbf{Y}$ variables, same as in Bayesian networks. Since $\mathbf{Y}$ variables are discrete, the CPDs on variable Y_i in the network can be represented as a conditional probability distribution table (CPT), with entries $p(Y_i = y_i|Pa_i = \mathbf{y}_{pa_i})$. For a fully connected network, where $Pa_i = \{Y_1, \cdots, Y_{i-1}, Y_{i+1}, \cdots, Y_k\}$, the conditional distribution tables could be very large, in the size of 2^k, since each variable has all the other variables as its parents. This can easily cause overfitting and produce spurious dependence relationships among the label variables. Thus simply modeling the label dependencies in a separate complete network and then combining it with the prediction model is problematic. However, if we take the observations $\mathbf{X}$, which are either

continuous or discrete or a mixture of both, into the networks, the CPT style parameterization is not tractable anymore. Nevertheless, in the conditional scenario we can actually simplify and generalize the CPD representations by using probabilistic prediction functions. That is, we associate a binary prediction model with each variable Y_i. These prediction models are then used to define the conditional probability distributions on each label variable, and can be viewed as *generalized conditional probability tables*. These generalized conditional probability tables can substantially reduce the representation complexity and mitigate the overfitting problem.

Given the training set D introduced before, the training process for the proposed conditional dependency network model is very straightforward and simple. We directly train k binary probabilistic predictors, and each of them defines a conditional probabilistic distribution on one label variable given all the other label variables and the input features $\mathbf{x}$. For the conditional dependency network in Figure 1, it is shown one conditional distribution function is associated with each Y_i. Many existing standard binary probabilistic classifiers can be used in our model to parameterize the conditional distributions. In our experiments, we in particular used the regularized binary logistic regression classifier.

Logistic regression is a well known statistical model for probabilistic classification. For the parameter learning of our conditional dependency networks, we train k logistic regression classifiers, and each, $p(y_i = \pm 1|\mathbf{x}, \mathbf{y}_{\neg i}, \boldsymbol{\theta}_i)$, is associated with one label variable Y_i, where $\boldsymbol{\theta}_i$ denotes the model parameters. The model parameters can be trained by maximizing the regularized likelihood of the training data

$$\max_{\boldsymbol{\theta}_i} \sum_{\ell=1}^{t} \log p(y_i^\ell|\mathbf{x}^\ell, \mathbf{y}_{\neg i}^\ell, \boldsymbol{\theta}_i) - \frac{\lambda}{2}\left(\boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i\right),$$

where $\frac{\lambda}{2}\left(\boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i\right)$ is a L_2 regularization term introduced to reduce overfitting, and λ is a trade-off parameter. Logistic regression is a robust linear classifier that can be trained efficiently using convex optimization techniques. Nonlinear classifications can be achieved by simply introducing kernels.

3.2 Gibbs Sampling for Approximate Inference

After training k logistic regression models, we obtain a parameterized conditional dependency network, which has a fully connected graph structure with bidirectional edges, and has k sets of parameters $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k\}$ associated with the label variables to define generalized conditional probability distributions. Given the trained conditional dependence network, the next step is using it to predict the label vector $\mathbf{y} = \{y_1, \cdots, y_k\}$ for a test instance $\mathbf{x}$. This multi-label classification problem is equivalent to computing a type of maximum a posteriori (MAP) explanation, also known as most probable explanation (MPE):

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$$

It has been shown this problem is NP-hard even for acyclic directed graphical models, i.e. Bayesian networks [Shimony, 1999]. When there are undirected cycles in a Bayesian network, many approximate inference algorithms have been developed to address the problem [Guo and Hsu, 2002]. It is a straightforward corollary that the inference problem we have for conditional dependency networks is also NP-hard, since our model is a cyclic variant of Bayesian networks. Given the fully connected structure and the generalized conditional probability tables we have, Gibbs sampling, which samples one variable given all others fixed, is a more suitable inference technique for our model than other alternatives.

Gibbs sampling [Geman and Geman, 1984] is a Metropolis-Hastings sampling algorithm that is especially appropriate for inference in graphical models. The key to the Gibbs sampling is that one only considers univariate conditional distributions, i.e. the distribution when all of the variables but one are assigned fixed values. This property makes Gibbs sampling a perfect fit on the fully connected conditional dependency network we build above, where the univariate conditional distributions needed for Gibbs sampling are directly available from the conditional probabilistic predictors associated with each variable. The inference procedure of Gibbs sampling is very simple. We first choose a random ordering of the variables, r, and initialize each variable Y_i to a value y_i. In each sampling iteration, we visit each variable in the given order, $\{Y_{r(1)}, \cdots, Y_{r(k)}\}$, where r maps the new order index into the original variable index. The new value of each variable $Y_{r(i)}$ is resampled according to the conditional predictor associated with it, $p(y|\mathbf{x}, \mathbf{y}_{\neg r(i)}, \boldsymbol{\theta}_{r(i)})$.

The idea behind Gibbs sampling is to approximate the joint distribution from the samples obtained from the conditional distributions. The sampler is expected to converge to a stationary distribution after some burn-in iterations. One then can collect samples to recover the approximated joint distribution and determine the MPE. There are a few different ways to decide the MPE of the $\mathbf{Y}$ variables. One typical way is to compute the marginal probabilities associated with each single variable from the samples and make the prediction based on the marginals. We however compute the MPE from samples that have high values over the product of the conditional probabilities, $\prod_i p(y_i|\mathbf{x}, \mathbf{y}_{\neg i}, \boldsymbol{\theta}_i)$, since this product can be viewed as an approximation to the true conditional joint distribution $p(y_1, \cdots, y_k|\mathbf{x})$. In order to avoid the instability of picking only one such sample with the highest product value, we collect n instances that have the top product values over the conditional probabilities. The final prediction is determined by the marginals computed from these n instances. We used $n = 100$ in our experiments. The overall Gibbs sampling procedure is described in Algorithm 1. In our experiments, we set the burn-in time as 100 iterations and use another 500 iterations to collect samples.

3.3 Extension to Non-probabilistic Models

The conditional dependency network model proposed above can exploit any probabilistic binary classifier. However, non-probabilistic binary classifiers, e.g. support vector machines, often demonstrated superior classification performance than probabilistic classifiers in many scenarios. We therefore extend our model to permit discrete conditional probability distributions over each label variable, i.e. $p(y_i = 1|\mathbf{x}, \mathbf{y}_{\neg i}, \boldsymbol{\theta}_i)$ is either 1 or 0. This extension allows nonprobabilistic binary classifiers to be exploited in our model. In particular, we consider support vector machines in our experiments.

Algorithm 1 Gibbs Sampling Inference

Input: $\mathbf{x}$: observed features; k: number of classes;
n: number of instances to pick;
$\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k\}$: model parameters;
t_b: burn-in iteration number;
t_c: instance collection iteration number;
Output: sampled instance set B
Procedure:
1. initialize $\mathbf{y} = (y_1, \ldots, y_k)$, $\ell = 0$;
choose a random ordering r over variables $\mathbf{Y}$.
2.% burn-in and collection loops
for $iter = 1$ to $t_b + t_c$ **do**
for i=1 to k **do**
$q = p(y = 1|\mathbf{x}, \mathbf{y}_{\neg r(i)}, \boldsymbol{\theta}_{r(i)})$;
sample $u \sim$ uniform distribution of $(0, 1)$
if $u \leq q$ **then** $y_{r(i)} = 1$ **else** $y_{r(i)} = -1$
if $iter > t_b$ **then**
$s = \prod_j p(y_j|\mathbf{x}, \mathbf{y}_{\neg j}, \boldsymbol{\theta}_j)$
if $\ell < n$ **then**
$\ell = \ell + 1, S(\ell) = s, B(\ell, :) = \mathbf{y}$
else $j \leftarrow$ index of $v = min(S)$
if $v < s$ **then**
$S(j) = s, B(j, :) = \mathbf{y}$

The training process for the extended model is the same as before; we only need to train k binary classifiers. However, the inference procedure has to be adjusted to fit non-probabilistic models. The inference procedure is described in Algorithm 2. This procedure simply iteratively updates each variable until reaching convergence. It can be viewed as a nonprobabilistic simplification of Gibbs sampling.

4 Related Work

Multi-label classification has received increasing attention from machine learning community in recent years, due to its practical relevance, and its interesting aspects from a theoretical point of view. A large number of multi-label classification methods have been proposed, including simple methods based on binary classifiers, and advanced methods that exploit label correlations. We review a set of methods that are most related to our work in this section.

One simple way of addressing multi-label learning is to transform the multi-label classification problem into a few single-label classification problems, e.g., the most intuitive one-vs-rest learning methods [Lewis *et al.*, 2004]. Further improvements have been studied on finding proper thresholds to determine multiple labels in a related way by considering ranking scores [Schapire and Singer, 2000], SVM scores [Boutell *et al.*, 2004], etc. In [Schapire and Singer, 2000], a boosting algorithm gives rise to a multi-label ranking system. By defining a special cost function based on a ranking loss, [Elisseeff and Weston, 2001] proposed a kernel method for ranking-based multi-label classification.

Many more sophisticated methods take the label co-occurrence information directly into account to improve classification accuracy. A multi-label k-nearest neighbor (MLKNN) approach presented in [Zhang and Zhou, 2005] takes the correlations of different labels into account to extend standard k-nearest neighbors. A combination of MLKNN and logistic regression was presented in [Cheng and Hüllermeier, 2009], where the neighborhood label information was used as features for logistic regression classifiers.

[Godbole and Sarawagi, 2004] proposed an SVM method with heterogeneous feature kernels (SVM-HF). In this method, one binary SVM, S_j, was first trained for each label j, then each training instance $\mathbf{x}^\ell$ was augmented with k additional label features produced by $\{S_1, \cdots, S_k\}$. Finally, new SVMs $\{\tilde{S}_1, \cdots, \tilde{S}_k\}$ can be trained on the augmented instances. In the test process, a test instance $\mathbf{x}$ was first classified by $\{S_1, \cdots, S_k\}$ to produce the label features, and the augmented instance was then provided to $\{\tilde{S}_1, \cdots, \tilde{S}_k\}$ to obtain the final prediction result. This approach shares some similar intuition with our proposed approach, but lacks the general and principled framework we have. Moreover, its prediction procedure remains in a naive stage. In [Hariharan *et al.*, 2010], a max-margin multi-label classification approach was proposed for large scale problems. However, it requires prior label correlation information to be provided.

Graphical models have also been used for multi-label classifications. The methods developed in [de Waal and van der Gaag, 2007; Rodriguez and Lozano, 2008; Bielza *et al.*, 2011] all employed Bayesian networks to address multi-label classification problem. However, these approaches involve directed structure learning, and their models are less flexible in handling different types of input features $\mathbf{x}$. [Ghamrawi and McCallum, 2005] proposed two undirected graphical models to exploit label co-occurrence information within the framework of conditional random fields. Both their training and inference procedures are more complicated than ours on dependency network models. Without pruning, their approaches cannot handle large number of classes.

It has been theoretically shown in [Streich and Buhmann, 2009] that inference schemes ignoring co-occurrence imply a model mismatch and thus cause biased parameter estimators. It suggests both co-occurrence statistics and collective classification over multiple labels should be considered. Our proposed model nicely integrates these two aspects by training k augmented binary classifiers and employing a Gibbs sampling for the joint prediction of multiple labels. The training phase for our model is very simple and straightforward, and the complexity of the multi-label classification is mainly handled in the approximate inference phase.

5 Experimental Results

In this section we conduct experiments to investigate the empirical performance of the proposed conditional dependency network model comparing with related works. The experiments are conducted on six widely used real-world multi-label data sets: yeast, scene, enron, emotion, medical, rcv1 and genbase. These data sets come from different problem domains including text, biology, and music. All results reported in this section are averages over 10 times repeats using random training/test partitions. The train/test sizes used for each data set are listed as follows: yeast(1500/917), enron(1123/579), emotion(391/202), medi-

Algorithm 2 Discrete Inference

Input: $\mathbf{x}$: observed features; k: number of classes;
$\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k\}$: model parameters.
Output: the predicted label vector $\mathbf{y}$.
Procedure:
initialize $\mathbf{y} = (y_1, \ldots, y_k)$;
choose a random ordering r over variables $\mathbf{Y}$.
repeat
  for i=1 to k **do**
    $y_{r(i)} = 2p(y = 1|\mathbf{x}, \mathbf{y}_{\neg r(i)}, \boldsymbol{\theta}_{r(i)}) - 1$
  end for
until converge

cal(645/333), rcv1(3000/3000), and genebase(463/200).

First we used L_2 regularized probabilistic logistic regression (LR) as the binary classifier in our model and name the resulted algorithm as "conditional dependency network-logistic regression" (CDN-LR). We compared the CDN-LR algorithm with the following three multi-label classification algorithms: (1) a one-vs-rest baseline L_2 regularized logistic regression method (LR), which conducts binary classifications for each class independently; (2) the MLKNN algorithm proposed in [Zhang and Zhou, 2005]; and (3) the collective multi-label classification (CML) algorithm proposed in [Ghamrawi and McCallum, 2005], which is based on Markov random fields. We used MLKNN and CML as comparison methods since they both directly exploit the label co-occurrence information. For MLKNN, we adopted $k = 7$ as suggested in [Zhang and Zhou, 2005].

We next investigated the nonprobabilistic extension of conditional dependency networks presented in Section 3.3, using SVMs as binary classifiers. The resulted algorithm is denoted as "conditional dependency network-SVM" (CDN-SVM). We compared it with three SVM-based multi-label classification algorithms: (1) the corresponding baseline algorithm, one-vs-rest SVM, which conducts binary classifications for each class independently; (2) the SVM with heterogeneous feature kernels (SVM-HF) proposed in [Godbole and Sarawagi, 2004]; and (3) the ranking-based kernel SVM method proposed in [Elisseeff and Weston, 2001]. The hyperparameter C in SVMs were selected via cross-validation.

The results for the eight algorithms on the six data sets introduced above are reported in Table 1, 2 and 3, using the widely used *exact match ratio*, *macro-F1* measure and *micro-F1* measure respectively. The results in the three tables show that the proposed CDN-LR algorithm overperformed all the other three approaches, LR, MLKNN, and CML, almost on all the six data sets regarding all three evaluation criteria. The only exception is that MLKNN achieved a better macro-F1 than CDN-LR on enron data set. Regarding the nonprobabilistic extension, the CDN-SVM algorithm outperformed the alternative SVM, SVM-HF and RankSVM on most data sets for all three evaluation measures. It has only been slightly overperformed by SVM-HF regarding micro-F1 on genbase data set, and outperformed by RankSVM regarding macro-F1 on yeast data set. The consistent superior performance of the proposed model demonstrated in these experiments over alternative multi-label classification methods suggests the dependency model we proposed is successfully effective in exploiting the label dependency information to improve multi-label classification performance. We also noticed the difference between the performance of CDN-LR and CDN-SVM are mostly due to the difference between the performance of base LR and base SVM. This suggests the base classifier selection is important as well. Nevertheless the conditional dependency network model proposed in this paper can incorporate a wide range of classification algorithms.

6 Conclusions

In this paper, we propose a novel generalized conditional dependency network model for multi-label classification. The proposed conditional dependency network is a fully connected bidirectional graph, whose conditional distributions are defined using binary classifiers. This model allows a very simple training procedure, while its representation naturally facilitates a simple Gibbs sampling inference on the test instances. The proposed model can incorporate a wide range of simple classification algorithms, including both probabilistic classifiers and nonprobabilistic classifiers. We tested this model using two base classifiers, logistic regression and SVMs, in our experiments. Our empirical results suggest the proposed model is very effective in exploiting the dependencies of multiple labels, and has demonstrated superior performance over a few alternative multi-label classification methods that exploit the same label co-occurrence information.

References

[Bielza *et al.*, 2011] C. Bielza, G. Li, and P. Larrañaga. Multi-dimensional classification with Bayesian networks. *Inter. Journal of Appromixate Reasoning, In Press*, 2011.

[Boutell *et al.*, 2004] M. Boutell, J. Luo, X. Shen, and C. Brown. Learning multi-label scene classiffication. *Pattern Recognition*, 37(9):1757–1771, 2004.

[Cheng and Hüllermeier, 2009] W. Cheng and E. Hüllermeier. Combing instance-based learning and logistic regression for multilabel classification. *Pattern Recognition*, 76:215–225, 2009.

[Chickering *et al.*, 1994] D. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks is NP-hard. Technical report, Micro. Research, MSR-TR-94-17, 1994.

[de Waal and van der Gaag, 2007] P. de Waal and L. van der Gaag. Inference and learning in multi-dimensional Bayesian network classifiers. In *Proc. of Euro. Conf. on Symb. and Quant. Appr. to Reason. with Uncertain.*, 2007.

[Elisseeff and Weston, 2001] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Inform. Process. Systems (NIPS)*, 2001.

[Geman and Geman, 1984] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE TPAMI*, 6:721–741, 1984.

[Ghamrawi and McCallum, 2005] N. Ghamrawi and A. McCallum. Collective multi-label classification. In *Proc. of Inter. Conf. on Inform. and Know. Manage. (CIKM)*, 2005.

Table 1: Performance in terms of Exact Match Ratio

Data	Yeast	Enron	Emotion	Medical	RCV1	Genbase
LR	.149±.008	.084±.015	.191±.029	.517±.029	.434±.017	.970±.011
MLKNN	.161±.011	.074±.019	.167±.021	.492±.023	.411±.017	.940±.029
CML	.167±.009	.085±.015	.152±.022	.523±.025	.437±.017	.970±.011
CDN-LR	**.174±.008**	**.090±.014**	**.225±.025**	**.546±.023**	**.448±.016**	**.975±.010**
SVM	.158±.009	.089±.013	.233±.037	.617±.024	.458±.015	**.970±.011**
SVM-HF	.159±.010	.090±.013	.237±.037	.622±.024	.467±.015	**.970±.011**
RankSVM	.161±.011	.087±.013	.225±.043	.407±.023	.423±.016	.960±.012
CDN-SVM	**.164±.010**	**.092±.013**	**.241±.035**	**.626±.023**	**.471±.014**	**.970±.011**

Table 2: Performance in terms of Macro-F1

Data	Yeast	Enron	Emotion	Medical	RCV1	Genbase
LR	.400±009	.151±.017	.607±.025	.379±.022	.351±.021	.802±.043
MLKNN	.413±.006	**.183±.013**	.377±.025	.223±.012	.344±.019	.637±.027
CML	.405±.007	.154±.017	.568±.022	.391±.021	.355±.021	.802±.043
CDN-LR	**.438±.006**	.171±.016	**.615±.024**	**.416±.021**	**.372±.020**	**.803±.043**
SVM	.344±.010	.205±.015	.625±.028	.397±.022	.374±.020	.801±.036
SVM-HF	.353±.010	.216±.015	.629±.027	.406±.022	.379±.020	**.802±.036**
RankSVM	**.387±.010**	.201±.016	.566±.032	.366±.038	.402±.017	.757±.037
CDN-SVM	.357±.010	**.219±.015**	**.641±.026**	**.419±.022**	**.383±.020**	**.802±.036**

Table 3: Performance in terms of Micro-F1

Data	Yeast	Enron	Emotion	Medical	RCV1	Genbase
LR	.605±.008	.478±.013	.619±.020	.698±.025	.684±.019	.984±.006
MLKNN	.630±.004	.484±.010	.467±.018	.673±.021	.686±.019	.969±.017
CML	.612±.007	.481±.013	.593±.020	.703±.024	.685±.019	.984±.006
CDN-LR	**.640±.006**	**.495±.012**	**.629±.019**	**.722±.022**	**.712±.019**	**.986±.006**
SVM	.618±.006	.477±.011	.641±.027	.774±.012	.723±.018	.983±.006
SVM-HF	.635±.006	.491±.011	.646±.026	.778±.012	.732±.017	**.985±.006**
RankSVM	.587±.006	.467±.011	.651±.031	.660±.023	.695±.019	.962±.009
CDN-SVM	**.638±.006**	**.494±.011**	**.654±.025**	**.787±.011**	**.738±.017**	.983±.006

[Godbole and Sarawagi, 2004] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Proc. of Pacific-Asia Conf. on Know. Disc. and Data Mining (PAKDD)*, 2004.

[Guo and Hsu, 2002] H. Guo and W. Hsu. A survey of algorithms for real-time Bayesian network inference. In *AAAI/KDD/UAI Joint Workshop on Real-Time Decision Support and Diagnosis Systems*, 2002.

[Hariharan *et al.*, 2010] B. Hariharan, L. Zelnik-Manor, S.V.N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *Proc. of Inter. Conf. on Machine Learning (ICML)*, 2010.

[Heckerman *et al.*, 2000] D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering and data visualization. *JMLR*, 1:49–75, 2000.

[Lewis *et al.*, 2004] D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

[Neal, 1993] R. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, CRG-TR-93-1, Dept of Comput. Science, Univ. of Toronto, 1993.

[Rodriguez and Lozano, 2008] J. Rodriguez and J. Lozano. Multiple-objective learning of multi-dimensional Bayesian classifiers. In *Inter. Conf. on Hybrid Intelligent Systems*, 2008.

[Schapire and Singer, 2000] R. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning Journal*, pages 135–168, 2000.

[Shimony, 1999] S. Shimony. Finding MAPs for belief networks is NP-hard. *Arti. Intell.*, 68:399–410, 1999.

[Srebro, 2001] N. Srebro. Learning Markov networks: maximum bounded tree-width graphs. In *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[Streich and Buhmann, 2009] A. Streich and J. Buhmann. Ignoring co-occurring sources in learning from multi-labeled data leads to model mismatch. In *Inter. Workshop on Learning from Multi-Label Data*, 2009.

[Zhang and Zhou, 2005] M. Zhang and Z. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *IEEE Inter. Conf. on Granular Computing*, 2005.