ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1336

Improved Multi-Label Classification Using Inter-Dependence Structure via a Generative Mixture Model

Ramanuja Simha[1] and **Hagit Shatkay**[1]

Abstract. Single-label classification associates each instance with a single label, while multi-label classification (MLC), assigns *multiple* labels to instances. Simple MLC systems assume that labels are independent of one another, while more complex approaches capture inter-dependencies among labels. Experiments comparing performance of MLC systems demonstrate that there is much room for improvement.

Notably, when an instance is associated with multiple labels, a feature-value of the instance may depend only on a subset of these labels and thus be *conditionally independent* of the others given the label-subset. Current systems do not account for such conditional independence. Moreover, dependence of a feature-value on a label is likely to imply its dependence on other inter-dependent labels. Our hypothesis is that by explicitly modeling the dependence between feature values and specific subsets of *inter-dependent* labels, the assignment of multi-labels to instances can be done more accurately.

We present a probabilistic generative model that captures dependencies among labels as well as between features and labels, by means of a Bayesian network. We introduce the concept of *label dependency sets* as a basis for a new mixture model that represents conditional independencies between features and labels given subsets of inter-dependent labels. Experimental results show that the performance of the system we have developed based on our model for MLC significantly improves upon results obtained by current MLC systems that are based on probabilistic models.

1 Introduction

Multi-label classification (MLC) associates instances with possibly multiple labels, in contrast to *single-label classification*, where each instance is associated with a single label. Simple approaches for multi-label classification transform the task into one or more single-label classification task(s). For instance, under the Ranking by Pairwise Comparison method [22], a classifier is trained to distinguish between each possible pair of labels (one-vs-one). A computationally efficient alternative is the Binary Relevance method [22], where each classification task corresponds to distinguishing a single label from the rest (one-vs-all).

More advanced approaches for multi-label classification capture dependencies among labels. For example, Multi-Label Search [8] explores a search space of label sets to capture such dependencies while learning a mapping of instances to multi-labels. A more widely used method is a Classifier Chain [14, 15], which consists of multiple binary classifiers like those used in Binary Relevance, one classifier per label. The chain is constructed based on an input label ordering. To capture relationships among labels, the *feature-vector* used to represent an input instance given to a classifier F includes lab el assignments obtained from all classifiers preceding classifier F in the chain. Systems based on Classifier Chains include probabilistic variants [5, 6, 16], and others that explicitly learn label inter-dependencies such as a chain of Support Vector Machines [26], a chain of naïve Bayes classifiers [25], and an ensemble of Bayesian networks [1]. Other approaches that employ graphical models, however not based on Classifier Chains, include Conditional Dependency Networks [10], which use a fully-connected graphical model, and systems that utilize probabilistic generative models [13, 17, 23], typically built for classifying text. The latter class of systems have not been extensively tested against other MLC systems and on datasets other than text.

In the context of MLC, a feature-value of an instance typically depends on some subset of the instance labels and thus may be *conditionally independent* of the other labels given this subset. For example, the *grade* feature value of college students who are classified (labeled) as *Excelled in entrance tests* and *Admitted into a graduate program* is typically *High*, regardless of any other student labels. Furthermore, dependence of a feature-value on a label is likely to suggest its dependence on other inter-dependent labels. Current systems do not account for the conditional independence between a feature-value and other labels given a subset of labels. Moreover, performance of current methods leaves much room for improvement. Our hypothesis is that explicitly modeling the dependencies between feature values and *inter-dependent* labels, as part of the classifier model, can support a more accurate assignment of multi-labels to instances.

We present a probabilistic generative model that captures dependencies among labels as well as between features and labels, by means of a Bayesian network. We introduce a mixture model to represent conditional independencies between features and labels given subsets of *inter-dependent* labels, and further develop a multi-label classifier. Unlike previous approaches, our system uses an iterative process to infer values for multiple labels simultaneously. In each iteration, the Bayesian network is modified to reflect inter-dependencies among the *most recently inferred label values*; the accuracy of the updated label assignments is thus improved by capturing specific feature-label correlations and dependencies. We evaluate our system on several multi-label datasets used before for evaluating MLC systems, and demonstrate that the performance of our system improves upon that obtained by current Classifier-Chain systems.

[1] Computational Biomedicine and Machine Learning lab, Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716 USA. Email: {rsimha, shatkay}@udel.edu

In the next section we introduce relevant notations and present our probabilistic generative model. Section 3 discusses procedures used for learning structure and parameters of the model, and inference techniques applied for multi-label classification. Section 4 provides details about the multi-label datasets we use, performance evaluation measures, and experimental results, while Section 5 concludes and summarizes our findings and outlines future directions.

2 A Dependency-Based Mixture Model for Multi-Label Data

Let D be a dataset containing m instances, and $C=\{c_1,\ldots,c_q\}$ be a set of q *class*-labels. Each instance in D is associated with a subset of labels. As others have done before in the context of multi-label classification (MLC) [1, 5], we represent an instance $I \in D$ as a feature vector, $\vec{f}^I=\langle f_1^I,\ldots,f_d^I\rangle$, and I's labels as a label vector, $\vec{l}^I=\langle l_1^I,\ldots,l_q^I\rangle$. Here d is the number of features, and $l_i^I=1$ if instance I is associated with label c_i, $l_i^I=0$ otherwise. Each feature value f_j is viewed as a value taken by a feature random variable F_j, and each label-indicator value l_i is viewed as a value taken by a label random variable L_i. The task of multi-label classification thus amounts to developing a classifier that takes as input an instance represented by a feature vector, and outputs a q-dimensional label vector.

2.1 Model Framework

We use a Bayesian network framework to model inter-dependencies among labels as well as between features and labels. Each node represents either a label variable L_i ($1 \le i \le q$) or a feature variable F_j ($1 \le j \le d$), and each directed edge indicates a dependence relationship between a pair of variables.

Representing label inter-dependencies

In the context of multi-label classification, labels may be directly correlated with one another, regardless of their association with any specific instance. As a simple example, drivers that are labeled as *Speeding* are also likely to be labeled *Accident-Prone*, regardless of any specific driver characteristics (features). We represent each *unconditional dependence* between a pair of labels c_i and c_j as a directed edge from label variable L_i to variable L_j. As another example, Figure 1 shows the more complex inter-dependency structure among label variables that we learn as part of our experiments (see Section 4) in the context of the *Emotions* dataset [21]; in this example, instances are songs and labels are emotions. A directed edge, e.g. from *Amazed-Surprised* to *Sad-Lonely*, represents the assertion that knowing that an instance is associated with the label *Amazed-Surprised* influences the level of belief about the instance's association with the label *Sad-Lonely*.

A label may often depend on a small set of a few labels while being conditionally independent of other labels given this set. To continue the previous simple example, the label *Accident-prone* is conditionally independent of the label *New-driver* given the label *Speeding*. To capture such *conditional independencies*, we introduce the concept of *label dependency sets*. A *dependency set* for a label c_i is a minimal set of labels, $c_{i_1},\ldots,c_{i_m}$ such that knowing an instance's association with each label c_{i_j} in the set is sufficient to infer the likelihood of the instance to be associated with c_i. We utilize the Bayesian network framework to obtain a practical representation of a label dependency set. The network structure captures both direct dependencies between pairs of labels and conditional independencies among labels given certain subsets of them. More specifically, each label variable L_i directly depends on its parents $\mathrm{Pa}(L_i)$ while being conditionally independent of its non-descendants given $\mathrm{Pa}(L_i)$; the joint distribution of the label variables is thus given by:

$$\Pr(L_1,\ldots,L_q)=\prod_{i=1}^{q}\Pr(L_i|\mathrm{Pa}(L_i))\ .$$

Employing the above conditional independence, we refer to a label variable L_i and its parents in the network as the *label dependency set* for L_i. Thus, for each variable L_i ($1 \le i \le q$), we define a label dependency set: $LS_i=\{L_i\}\cup\mathrm{Pa}(L_i)$.

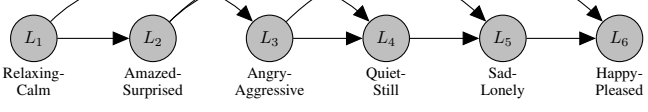

Figure 1: An example Bayesian network structure over labels that we learn using the Emotions dataset [21].

Representing dependencies between features and labels.

An instance's association with certain labels is clearly correlated with the instance feature values. Additionally, the value of a feature may be correlated with multiple labels and not just with one. For example, in the *Emotions* dataset [21], where instances are songs represented using *rhythm* and *tone* features and labels are *emotions*, the value of the tone feature is typically *Low* when a song is labeled as *Sad-Lonely* while it is typically *High* when a song is simultaneously labeled as both *Sad-Lonely* and *Amazed-Surprised*.

As explained earlier while introducing label dependency sets (LDS), the association of an instance with certain labels typically provides information about its association with other labels. For instance, a song that has a *High* tone feature value and is labeled as *Amazed-Suprised* is likely to also be labeled as *Sad-Lonely*. We represent the *dependence* of a feature, F_j, on a subset of inter-dependent labels, LS_i (to which we refer as a label dependency set) by plotting directed edges connecting each label variable in the set with the feature variable. We thus capture the *conditional dependence among labels* in the set LS_i given the feature F_j. Figure 2 extends the ex-

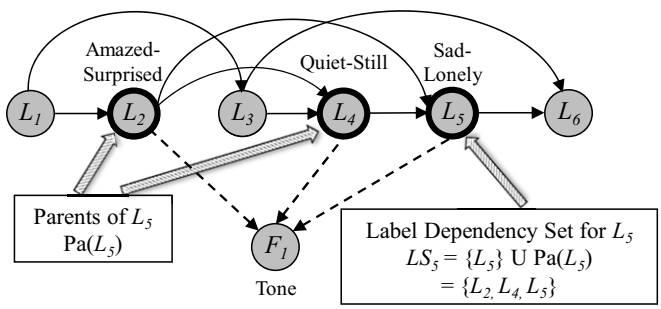

Figure 2: An extension of the network shown in Figure 1, where labels are emotions and features are rhythms/tones. The feature associated with the variable F_1 is tone. Bold-faced nodes (L_5 and its parents L_2 and L_4) form the *label dependency set*, LS_5. Solid directed edges represent dependencies among the labels while dashed edges represent dependence between the feature, F_1, and the label dependency set, LS_5.

ample Bayesian network presented in Figure 1. The dashed arrows from labels L_2, L_4, and L_5 (i.e., *Amazed-Surprised*, *Quiet-Still*, and *Sad-Lonely*) to the feature, F_1 (i.e. *Tone*) in Figure 2 capture the dependence of the feature on the subset of inter-dependent labels comprising the label L_5 and its parents, L_2 and L_4; this label subset is referred to as the *label dependency set*, $LS_5 = \{L_5\} \cup Pa(L_5)$.

Moreover, when an instance is associated with multiple labels, a feature-value of the instance may depend only *on a subset of* these labels. As an example, the *tone* feature value of a song that is labeled as *Sad-Lonely* and *Amazed-Suprised* is likely to be *High* regardless of the song's association with any other labels. That is, the *tone* is conditionally independent of all other labels, given the two labels *Sad-Lonely* and *Amazed-Suprised*. By explicitly representing *dependence* between a feature and the labels in an LDS as discussed above, our model captures *conditional independence* of the feature from all other labels given the LDS.

We next present a probabilistic generative model that captures the label inter-dependencies and dependencies between features and labels discussed above.

2.2 Model Description

Generative models have been used before for multi-label classification [13, 17, 23], typically for classifying text. While these models address dependencies among labels, they do not represent intricate dependencies between feature values and subsets of labels. In contrast, our proposed model captures conditional independencies of features from labels by directly representing the dependencies between feature values and label subsets. (In addition, our model is developed in the general context of multi-label classification — not limited to text.) We next discuss the *instance generation process*, based on a Bayesian network structure, and provide further detail about our model.

The generation process comprises two steps:

I **Labels assignment:** To generate an instance I, its class-labels are first determined, i.e., a label value l_i^I is assigned to each label variable L_i ($1 \leq i \leq q$). We view each label assignment as a *Bernoulli* event, where l_i^I=1 when I is associated with the label c_i, and l_i^I=0 otherwise. Based on the Bayesian network structure, the conditional probability of L_i to be assigned 1 given the values of its parents denoted $\mathcal{V}_{\mathrm{Pa}(L_i)}$ is denoted as: $\alpha_i = \Pr(L_i=1|\mathcal{V}_{\mathrm{Pa}(L_i)})$[2] while its probability to be assigned 0 is $1-\alpha_i$. The order in which label values are assigned is based on the topological order of label variables, $L_{t1}, \ldots, L_{tq}$ in the Bayesian network. The assigned label values form a label-vector $\vec{l}^I$ for instance I.

II **Features assignment:** Based on the label-vector $\vec{l}$, a label dependency set LS_{F_j} is selected for each feature F_j. We expect this set to constitute a small subset of labels such that F_j's value depends only on the label subset. We introduce a *Multinomial* random variable λ^{F_j} that takes on a value $k \in \{1, \ldots, q\}$ with probability: $\theta^{\vec{l}}_{j,k} = \Pr(\lambda^{F_j}=k|\vec{l})$; $\lambda^{F_j}=k$ indicates the selection of the k'th label dependency set. We denote this set as: $LS_k = \{L_k\} \cup \mathrm{Pa}(L_k)$. We refer to $\theta^{\vec{l}}_{j,k}$ as the *mixture parameter* as it models the dependence between the labels in each label dependency set, LS_k and the value of feature F_j.

The value for feature F_j is thus selected based on the values taken by the random variables in the set LS_k, denoted as $\mathcal{V}_{LS_k}$. We view each feature value selection as a *Multinomial* event, where the probability of F_j to take on a value v is: $\phi_{j,k}(v) = \Pr(F_j=v|\lambda^{F_j}=k, \mathcal{V}_{LS_k})$. Here $\phi_{j,k}(v)$ denotes the conditional probability of feature F_j to take on the value v given the label dependency set LS_k. In the model, all feature variables are assumed to take on discrete values. We thus discretize each real-valued feature in the datasets used for our experiments, as done by earlier studies [1] (see Section 4 for further details regarding discretization). The selected values for all features together form a complete feature-vector $\vec{f}^I$ representing the instance I.

[2] Throughout the paper, for any set, S, of random variables, we denote by $\mathcal{V}_S$ the values taken by the variables in this set.

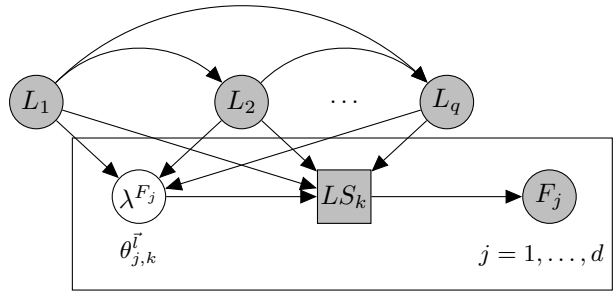

Figure 3: Bayesian network representing the generative mixture model of multi-label data.

We note that the process of assigning feature values enforces several independence relationships. Having selected LS_k as the label dependency set, we denote by $\overline{L}_k$ the set of all label variables other than $\{L_k\} \cup \mathrm{Pa}(L_k)$, i.e., $\overline{L}_k = \{L_1, \ldots, L_q\} - LS_k$. The value selection — for each feature F_j — is *conditionally independent* of all labels in $\overline{L}_k$ given LS_k. Furthermore, the selected feature-value for F_j is also *conditionally independent* of all other feature values given the label vector $\vec{l}^I$.

Formally stated, the *independence assumptions* enforced by our model are:

(i) The feature values $f_1^I, \ldots, f_d^I$ of an instance I, are *conditionally independent* of each other given the instance's label vector $\vec{l}^I$:

$$\Pr(\vec{f}^I|\vec{l}^I) = \prod_{j=1}^{d} \Pr(f_j^I|\vec{l}^I)\ . \tag{1}$$

Although this assumption over-simplifies feature inter-dependencies, it has been proven effective [1, 25].

(ii) Given the values taken by a label variable L_k and its parents $\mathrm{Pa}(L_k)$ in a *selected* label dependency set LS_k for an instance I, a value f_j^I of a feature is *conditionally independent* of all other labels of I:

$$\Pr(F_j=f_j^I|\lambda^{F_j}=k, L_1=l_1^I, \ldots, L_q=l_q^I) = \\ = \Pr(F_j=f_j^I|\lambda^{F_j}=k, L_k=l_k^I, \mathcal{V}_{\mathrm{Pa}(L_k)})\ . \tag{2}$$

Figure 3 shows a graphical representation of the generative mixture model presented above. Nodes represent random variables, and directed edges represent dependencies among variables. Label and feature random variables are denoted as circles. In contrast, the label dependency set, LS_k is denoted by a square as its value is deterministically assigned based on values taken by the label variable L_k and its parents $\mathrm{Pa}(L_k)$. Notably, the node for LS_k represents a set, and as such the directed edge from LS_k to the feature F_j is a short-hand

for multiple edges connecting each label variable in LS_k with F_j as described in Section 2.1. Variables representing labels and features are *observed*, that is, their values are provided within the training dataset. These variables are shown as shaded in the figure. The value of the variable λ^{F_j} is governed by the mixture parameter $\theta^{\vec{l}}_j$ and is not given as part of the dataset. As such, it is *latent* and shown as unshaded.

Under all the above mentioned independence assumptions and based on the structure of our generative model, the joint probability of the label vector $\vec{l}^I$ and the feature vector $\vec{f}^I$ is expressed as:

$$\Pr(\vec{l}^I,\vec{f}^I) = \Pr(\vec{l}^I)\Pr(\vec{f}^I|\vec{l}^I) = \\ = \prod_{i=1}^{q} \Pr(L_i = l_i^I | \mathcal{V}_{\mathrm{Pa}(L_i)}) \times \\ \times \prod_{j=1}^{d} \sum_{k=1}^{q} \theta^{\vec{l}}_{j,k} \Pr(F_j = f_j^I | \lambda^{F_j} = k, L_k = l_k^I, \mathcal{V}_{\mathrm{Pa}(L_k)}) \ , \tag{3}$$

where:

(a) $\prod_{i=1}^{q} \Pr(L_i = l_i^I | \mathcal{V}_{\mathrm{Pa}(L_i)})$ is the factorization of the joint probability $\Pr(\vec{l}^I) = \Pr(L_1 = l_1^I, \ldots, L_q = l_q^I)$, based on the individual q label values, given the conditional independencies encoded in the network;

(b) $\Pr(F_j = f_j^I | \lambda^{F_j} = k, L_k = l_k^I, \mathcal{V}_{\mathrm{Pa}(L_k)})$ denotes the conditional probability of a feature value f_j^I ($I \leq j \leq d$, where d is the total number of features), given the values taken by a label variable L_k and its parents $\mathrm{Pa}(L_k)$; $L_k \cup \mathrm{Pa}(L_k)$ comprises the label dependency set LS_k (under the current generative mixture model);

(c) $\theta^{\vec{l}^I}_{j,k}$ denotes the probability that the label dependency set LS_k is selected given a label-vector $\vec{l}^I$ for a feature F_j.

3 Model Learning and Inference

We next introduce a procedure for learning the structure and the parameters of our generative model, and present an inference technique for multi-label classification.

3.1 Structure and Parameter Learning

We employ an iterative procedure to learn the Bayesian network structure, specifically the structure of inter-dependencies among the label nodes shown at the top of Figure 3, and to estimate the model parameters shown on the RHS of Equation 3. This iterative procedure is summarized in the pseudocode shown in Figure 4.

In each iteration, we first learn a label inter-dependency structure using the BANJO package [20]; we then estimate the model parameters through an Expectation Maximization process; following that, we infer multi-label values for instances in the training set. The inter-dependency structure is learned in the first iteration from the training-set labels, and in subsequent iterations, from the most recently inferred label values. The model parameters are estimated throughout the learning procedure using the training-set labels.

We use this iterative process as it modifies the network structure to reflect inter-dependencies among the most recently inferred label values. We expect such a network to allow the system to capture specific feature-label correlations and conditional independencies, which in turn, may improve the accuracy of the updated label assignments. As shown in the experiments section, this assumption is indeed supported by the improved performance of our system.

```
1  Initialize Bayesian network structure using the BANJO
   package [20], based on training-set labels;
2  Initialize model parameters, α_i and φ_{j,k}(v) using maximum
   likelihood estimation, and θ^{l}_{j,k} using EM algorithm, based on
   training-set labels;
3  Set initial inferred label values (i.e. each l_i^I, i = 1, ..., q) for
   each instance I ∈ D to 0;
4  Set t to 0 and P to Hamming accuracy (or F1-score) of initial
   model over training set;
5  while True do
6      Update Bayesian network structure using BANJO, based on
       most recently inferred label values;
7      Update model parameters, α_i, φ_{j,k}(f_j^I), and θ^{l}_{j,k}, based on
       training-set labels;
8      while True do
9          Infer values taken by random variables in each label
           dependency set LS_k (see Figure 5 for details);
10         if Hamming accuracy (or F1-score) of model does not
           improve then
11             break;
12         end
13     end
14     Set P' to Hamming accuracy (or F1-score) of updated
       model over training set;
15     if P' ≤ P then
16         break;
17     end
18     P ← P'; t ← t + 1;
19 end
```

Figure 4: Summary of model learning.

At the end of each iteration we assess the classification performance of our model over the training set; the iterative procedure is terminated when there is no improvement in performance between two successive iterations. For assessing model performance, we utilize the F_1-score metric when using the dataset of multi-localized proteins, and the *Hamming accuracy* when using other multi-label datasets; these performance measures are described later in Section 4. The number of iterations needed to learn our model, which we denote by t, may vary across different datasets and also depends on the number of class-labels q; in our experiments, the number of iterations did not exceed 10.

We use maximum likelihood estimation to compute the two sets of *observed* model parameters (shown in Equation 3): (a) The conditional probability of a label l_i^I given the values taken by L_i's parents, $\alpha_i = \Pr(L_i = l_i^I | \mathcal{V}_{\mathrm{Pa}(L_i)})$ and (b) The conditional probability of a feature value f_j^I given the values taken by all variables in each label dependency set (LDS), LS_k ($1 \leq k \leq q$), $\phi_{j,k}(f_j^I) = \Pr(F_j = f_j^I | \lambda^{F_j} = k, L_k = l_k^I, \mathcal{V}_{\mathrm{Pa}(L_k)})$. To estimate the *latent* parameters, namely, the probability of each label dependency set, LS_k, $\theta^{\vec{l}}_{j,k}$, for a given label vector $\vec{l}$ and a feature F_j, we developed an Expectation Maximization algorithm [7]:

1. **Expectation step.** For each instance I, we compute the probability of each LDS LS_k, to be selected for feature F_j, that is, $\lambda^{F_j}=k$, given I's label vector $\vec{l}$ and feature-value f_j^I, as:

$$\Pr(\lambda^{F_j}=k|F_j=f_j^I,\vec{l^I}) = \frac{\theta_{j,k}^{\vec{l^I}}\Pr(F_j=f_j^I|L_k=l_k^I,\mathcal{V}_{\mathrm{Pa}(L_k)})}{\sum_{k=1}^{q}\theta_{j,k}^{\vec{l^I}}\Pr(F_j=f_j^I|L_k=l_k^I,\mathcal{V}_{\mathrm{Pa}(L_k)})}.$$

2. **Maximization step.** Using the probabilities computed in the Expectation step, we marginalize over all instances in the training set to re-estimate the mixture parameter, $\theta_{j,k}^{\vec{l}}$, for each feature F_j and label vector $\vec{l}$ as:

$$\theta_{j,k}^{\vec{l}} = \frac{\sum_{v_j}\sum_{\{I|\vec{l^I}=\vec{l},f_j^I=v_j\}}\Pr(\lambda^{F_j}=k|F_j=f_j^I,\vec{l^I})\Pr(F_j=f_j^I|\vec{l^I})}{\sum_{k=1}^{q}\sum_{v_j}\sum_{\{I|\vec{l^I}=\vec{l},f_j^I=v_j\}}\Pr(\lambda^{F_j}=k|F_j=f_j^I,\vec{l^I})\Pr(F_j=f_j^I|\vec{l^I})},$$

where v_j takes on all possible values for feature F_j.

We denote by $\vec{l^I}_{LS_k}$ the *restriction* of the label vector $\vec{l^I}$ to only those labels that are in the set LS_k. The conditional probability of a feature F_j to be assigned a value v given the values taken by the label variables in the label dependency set, LS_k, $\Pr(F_j=v|\mathcal{V}_{LS_k})$, is calculated as:

$$\Pr(F_j=v|L_k=l_k,\mathcal{V}_{\mathrm{Pa}(L_k)})=\Pr(F_j=v|\mathcal{V}_{LS_k}) = \frac{\sum_{\{I|\vec{l^I}_{LS_k}=\vec{l}_{LS_k},f_j^I=v\}}\Pr(\lambda^{F_j}=k|F_j=f_j^I,\vec{l^I})\Pr(F_j=f_j^I|\vec{l^I})}{\sum_{v_j}\sum_{\{I|\vec{l^I}_{LS_k}=\vec{l}_{LS_k},f_j^I=v_j\}}\Pr(\lambda^{F_j}=k|F_j=f_j^I,\vec{l^I})\Pr(F_j=f_j^I|\vec{l^I})}.$$

Throughout the estimation process, we apply Laplace smoothing [18] by adding fractional pseudocounts to observed counts of events to all the parameters to avoid overfitting. The process of alternating between the Expectation step and the Maximization step is carried out until convergence is reached. To determine convergence, we test that changes to the latent parameter values between iterations are smaller than 0.05.

We next present the inference procedure for assigning multiple labels to instances.

3.2 Probabilistic Multi-label Classification

Probabilistic inference in the context of multi-label classification (MLC) amounts to assigning the most probable label vector $\vec{l^I}$ to an instance I based on its feature vector $\vec{f^I}$. Inferring the conditional probability, $\Pr(\vec{l}|\vec{f})$ for each label vector $\vec{l}$ requires 2^q calculations, where q denotes the number of labels. To avoid this exponential number of calculations, some current probabilistic methods for multi-label classification assign a value to each label l_i $(1\le i\le q)$ such that the conditional probability $\Pr(L_i=l_i|\vec{f^I})$ is maximized (see e.g. [1]). Others estimate the joint probability of the labels, $\Pr(L_1=l_1,\ldots,L_q=l_q|\vec{f^I})$ and eventually infer each label value based on estimates of other labels (see e.g. [5]; [25]). These methods typically infer each label value by utilizing a fixed set of feature-label dependencies captured by their respective models.

In contrast, our system iteratively infers values for sets of multiple labels by capturing in each iteration specific feature-label dependencies based on the most recently inferred label values. We assign values to label variables in each label dependency set (LDS) LS_i (see Section 2.1 for the LDS definition), such that the conditional probability $\Pr(\mathcal{V}_{LS_i}|\vec{f^I})$ is maximized.

To ensure that our method is practically applicable, we set a limit on the maximum number of parents, p, per label variable in the network. In the experiments described here, we restrict the dependency-set size to three (i.e. we set $p=2$) because the mean number of labels per dataset is at most three; the number of inference calculations is thus $2^{p+1}q=2^3q=8q$, where q ranges between 6 and 27. To gauge the influence of changes to the values of p on classifier performance, we ran experiments by varying the maximum number of parents in the range 1-3 using *Emotions* and *Scene* datasets, which have a relatively low number of labels. While increasing the value of p leads to a notable increase in the *Subset accuracy* measure of the classifier, there is no significant improvement in the classifier's *Hamming accuracy* measure (see Section 4 for details about these measures). We anticipate that higher values of p can further improve classifier performance when running experiments on datasets with higher numbers of labels.

As our system considers *multiple* dependency structures between features and labels, we expect that setting a relatively low bound on the dependency-set size considered in each structure, as we do here, will still allow the system to capture the significant dependencies and independencies among features and label subsets, even in larger datasets. Moreover, unconditional direct dependencies are not the only ones our model captures. While each label depends on two parent-labels—thus conditionally independent of other labels, indirect inter-dependencies are still captured throughout the network structure. As demonstrated by the results in Section 4, our utilization of label subsets of even a small size still significantly improves the performance of our system compared to that of current systems.

Given a feature vector $\vec{f^I}$ of an instance I, our task is to predict its label vector $\vec{l^I}$, which involves assigning a 0/1 value to each of its labels l_i^I $(1\le i\le q)$. According to our probabilistic model, since the value of each label variable L_i depends only on values of its parent nodes $Pa(L_i)$ in a Bayesian network setting, for each L_i, we infer the values of variables in the label dependency set, $LS_i=\{L_i\}\cup Pa(L_i)$. To infer these label values, we follow an iterative process, which is summarized in the pseudocode shown in Figure 5. In each iteration, for all possible value assignments, l_i and $\mathcal{V}_{\mathrm{Pa}(L_i)}$ to the label variable L_i and its parents, respectively, we

```
1 foreach label dependency set LS_i = L_i ∪ Pa(L_i) do
2     foreach value assignment to L_i and to Pa(L_i) do
3         Calculate conditional probability:
          Pr(L_i = l_i, V_Pa(L_i) | f^I, V^I_{L_i})[3];
4     end
5     Select value assignment that maximizes the above
      conditional probability;
6     Update inferred values for labels in LS_i if classification
      performance over training set improves;
7 end
```

Figure 5: Summary of label inference.

Characteristic	Our System	EBN-M/EBN-J	ECC-NB	ECC-J48	BR-NB
Captures dependencies among labels	Using a probabilistic graphical model		Using classifier chains		No label inter-dependencies represented
Captures conditional independence between labels and features (given subsets of other labels)	Using label dependency sets and a mixture model	Do not capture such conditional independence			
Employs a generative model for data	Using Bayesian network		Using näive Bayes	No generative model used	Using näive Bayes

Table 1: Comparison of current multi-label classification systems characteristics.

calculate the conditional probability: $\Pr(L_i = l_i, \mathcal{V}_{\mathrm{Pa}(L_i)} | \vec{f}^I, \mathcal{V}^I_{\overline{L}_i})$.[3] The value assignment to L_i and to its parents, $Pa(L_i)$, that maximizes this probability is used as their current estimates. We note that label dependency sets do overlap, that is, the value of the same label variable L_i may be inferred multiple times, once for each dependency set in which it participates. As such, once the value of L_i is inferred within an iteration, it is only going to be updated during the same iteration if this improves the overall predictive performance of the model. While we currently use the standard inference techniques for Bayesian network models [18], there is much room for optimization by using methods for approximate inference that consider only the likely label combinations and fewer label sets, which we shall pursue in the future.

4 Experiments and Results

We present in this section two sets of experiments. In the first, we utilize the standard collection of multi-label datasets that were previously used to assess the performance of multi-label classification (MLC) systems. In the second, we employ a dataset used for a more concrete application in computational biology, namely predicting locations of proteins within the cell, also known as *protein multi-location prediction*. Details of these experiments are provided below.

4.1 Datasets and Performance Measures

For the first set of experiments, we use the same multi-label datasets that have been previously used in several comprehensive studies (e.g. [1, 8, 26]) to assess MLC system performance, namely: *Emotions* (72 *features*, 6 *labels*), *Scene* (294 *features*, 6 *labels*), *Yeast* (103 *features*, 14 *labels*), and *Genbase* (1186 *features*, 27 *labels*). We compare the performance of our system to that of state-of-the-art multi-label classification systems that were evaluated in a comprehensive study by Alessandro et al. [1]. The study focused primarily on MLC systems based on Classifier Chains, and included: ensemble of Bayesian networks, namely, *EBN-J / EBN-M* [1], ensemble of chain classifiers [15] using Naïve Bayes (denoted *ECC-NB*) and using J48 (denoted *ECC-J48*), and Binary Relevance using Naïve Bayes (denoted *BR-NB*) [22]. We note that the last of these four systems is not a classifier-chain but was still included in that study and is thus included here as well. As done in Alessandro's study, we discretize each real-valued feature into four bins, select features using a correlation-based feature selection technique [24], and employ the stratified 10-fold cross-validation for evaluating system performance. Table 1 summarizes the main distinguishing properties of the compared systems.

In the second set of experiments, we use a protein multi-location dataset, derived from DBMLoc [27], where each protein is represented by *30 features*, and the 9 possible subcellular locations correspond to *9 class-labels* (see [19] for detail). We compare the performance of our system to that of state-of-the-art multi-location prediction systems as reported by Briesemeister et al. [2], in their assessment of the YLoc$^+$ system [2], including Euk-mPLoc [3], WoLF PSORT [11], and KnowPred$_{\text{site}}$ [12]. According to the methods used in the previous assessment [2], we employ minimal entropy partitioning technique [9] for feature discretization, and stratified 5-fold cross-validation for training/testing the classifiers.

Under our current unoptimized implementation, wall clock time for model learning using training instances and inferring multi-labels of test instances combined is on the order of several minutes for datasets with a few labels, (lowest being $\leq$10 minutes for *Emotions*), and on the order of hours for datasets with more labels, (highest being $\sim$*20* hours for *Yeast*). We note that while the run-time of the prototypical system grows quadratically with the number of labels, it grows only linearly with the dataset size. For example, the run-time for the *protein multi-location* dataset (containing *8503* instances, with only *9* labels) is about 0.25 of the run-time for the smaller *Yeast* dataset (*2417* instances) that has *13* labels.

Throughout the experiments, we use the valuation measures described below, which are the same as those applied in the corresponding previous work. For a given instance I, let $M^I = \{c_i \mid l^I_i = 1, \text{where } 1 \leq i \leq q\}$ be the set of labels associated with I according to the dataset, and let $\hat{M}^I = \{c_i \mid \hat{l}^I_i = 1, \text{where } 1 \leq i \leq q\}$ be the set of labels assigned to I by a classifier, where each $\hat{l}^I_i$ is a 0/1 label assignment. The *Hamming* (H_{acc}) and the *Subset* (S_{acc}) accuracies used for the evaluation of multi-label prediction systems [1] are computed as:

$$H_{acc} = 1 - \frac{1}{|D|} \sum_{I \in D} \frac{1}{|C|} |M^I \Delta \hat{M}^I| \ , \text{ and}$$

$$S_{acc} = \frac{1}{|D|} \sum_{I \in D} \mathcal{I}(M^I = \hat{M}^I) \ ,$$

where Δ is the symmetric difference between M^I and $\hat{M}^I$. Additionally, the *Multi-label accuracy* (ML_{acc}) and F_1-*label* score used

[3] Recall that $\overline{L}_i$ denotes the set of all label variables *other than* L_i and $Pa(L_i)$ and that the values taken by the variables in $\overline{L}_i$ is denoted as $\mathcal{V}_{\overline{L}_i}$.

Measure	Dataset	Our system	EBN-M / EBN-J	ECC-J48	ECC-NB	BR-NB
H_{acc}	Emotions	**.793** ($\pm$ .021)	.780 ($\pm$ .022)	.780 ($\pm$ .027)	.781 ($\pm$ .026)	.776 ($\pm$ .023)
	Scene	**.898** ($\pm$ .010)	.880 ($\pm$ .010)	.883 ($\pm$ .008)	.835 ($\pm$ .007)	.826 ($\pm$ .008)
	Yeast	**.786** ($\pm$ .007)	.773 ($\pm$ .008)	.771 ($\pm$ .007)	.703 ($\pm$ .009)	.703 ($\pm$ .011)
	Genbase	**.998** ($\pm$ .001)	**.998** ($\pm$ .001)	**.998** ($\pm$ .001)	.996 ($\pm$ .001)	.996 ($\pm$ .001)
S_{acc}	Emotions	**.319** ($\pm$ .036)	.263 ($\pm$ .062)	.260 ($\pm$ .038)	.295 ($\pm$ .060)	.261 ($\pm$ .049)
	Scene	**.610** ($\pm$ .030)	.575 ($\pm$ .030)	.531 ($\pm$ .038)	.294 ($\pm$ .022)	.276 ($\pm$ .017)
	Yeast	**.158** ($\pm$ .029)	.127 ($\pm$ .018)	.132 ($\pm$ .023)	.102 ($\pm$ .023)	.091 ($\pm$ .020)
	Genbase	.956 ($\pm$ .022)	**.965** ($\pm$ .015)	.934 ($\pm$ .015)	.897 ($\pm$ .031)	.897 ($\pm$ .0031)

Table 2: Hamming and Subset accuracies, H_{acc} and S_{acc}, for multi-label prediction systems. All values except ours are taken directly from *Tables 2, 3, 4, 6* in the paper by Alessandro et al. [1]. Highest values are shown in boldface. Standard deviations are shown in parenthesis.

Measure	Our system	YLoc$^+$	Euk-mPLoc	WoLF PSORT	KnowPred$_{\text{site}}$
F_1-*label*	**0.71** ($\pm$ 0.02)	0.68	0.44	0.53	0.66
ML_{acc}	**0.68** ($\pm$ 0.01)	0.64	0.41	0.43	0.63

Table 3: F_1-*label* and ML_{acc} scores shown for protein multi-location prediction systems. All values except ours are taken directly from *Table 3* in the paper by Briesemeister et al. [2]. Standard deviations are not available there. Highest values are shown in boldface.

for evaluating multi-location prediction systems [2] are computed as:

$$ML_{acc} = \frac{1}{|D|} \sum_{I \in D} \frac{|M^I \cap \hat{M}^I|}{|M^I \cup \hat{M}^I|} \text{ , and}$$

$$F_1\text{-}label = \frac{1}{|C|} \sum_{c_i \in C} \frac{2 \times Pre_{c_i} \times Rec_{c_i}}{Pre_{c_i} + Rec_{c_i}} \text{ ,}$$

where Pre_{c_i} and Rec_{c_i} for label c_i are *adapted* measures of multi-label precision and recall given by Briesemeister et al. [2]:

$$Pre_{c_i} = \frac{1}{|\{I \in D | c_i \in \hat{M}^I\}|} \sum_{I \in D | c_i \in \hat{M}^I} \frac{|M^I \cap \hat{M}^I|}{|\hat{M}^I|} \text{ , and}$$

$$Rec_{c_i} = \frac{1}{|\{I \in D | c_i \in M^I\}|} \sum_{I \in D | c_i \in M^I} \frac{|M^I \cap \hat{M}^I|}{|M^I|} \text{ .}$$

4.2 Classification Results

Table 2 shows the *Hamming* and the *Subset* accuracies (H_{acc} and S_{acc}, respectively) of our system compared to that obtained by current MLC systems (as reported by Alessandro et al. [1], *Tables 2, 3, 4, 6* there), obtained over the same multi-label datasets and evaluation measures. The results show that our system has higher H_{acc} and S_{acc} than all other systems over all datasets except *Genbase*. The differences in the improved performance values are statistically significant ($p \ll 0.05$, according to the *2-sample t-test* [4]). Over the *Genbase* dataset, our system has the same H_{acc} as the others and a slightly lower S_{acc}, although the latter difference is not statistically significant. The reason for the lack of improvement in this case can be attributed to the fact that in the *Genbase* dataset, the mean number of labels per instance is much lower than in the other datasets. As such, there are relatively few dependencies and independencies among labels and features to be utilized by our system.

Table 3 shows the F_1-*label* score and *Multi-label* accuracy (ML_{acc}) of our system compared with those obtained by top multi-location prediction systems (as reported by Briesemeister et al. [2], *Table 3* there), obtained over the same set of multi-localized proteins and evaluation measures. The table shows that our system improves over the performance of all other systems. The differences between scores obtained by our system and those of the closest top performing system, YLoc$^+$, are highly statistically significant ($p \ll 0.001$).

Thus, the results clearly demonstrate that our system, which utilizes the intricate dependence and independence structure among features and labels, improves upon current multi-label classification methods, as shown over a variety of multi-label datasets previously used for systems-comparison.

5 Conclusions and Future Work

We presented a probabilistic generative model that captures inter-dependencies among labels as well as dependencies between features and labels. Unlike other approaches for multi-label classification (MLC), our model represents conditional independencies between feature values and labels given subsets of other labels, par-

ticularly by introducing the concept of *label dependency sets*. For example, in the *Emotions* dataset, the *tone* feature of songs depends on the class labels *Quiet-Still*, *Sad-Lonely*, *Amazed-Surprised*, and *Angry-Aggressive*. Typically, songs labeled as belonging to the first two classes have a *Low* tone while those in the last two classes have a *High* tone. Our system directly captures the conditional independence between the tone feature and the first two labels given the other two labels. Notably, current systems do not attempt to capture such subtle and informative dependencies and independencies. Our experiments over diverse datasets indeed show that directly modeling these dependence and independence relationships contributes to improved accuracy in multi-label classification, compared to previously studied systems based on Classifier Chains.

While we employ relatively small dependency sets in this study, the improved performance of our system strongly suggests that even such small sets can still help model the significant dependencies between feature and label subsets. We plan to develop approximate methods for inference by considering only the likely label combinations and fewer label sets to enable the practical use of larger label dependency sets.

Since utilizing label dependency sets has proven useful, our next aim is to directly learn label combinations that are most likely to strongly influence feature values. We will conduct experiments over larger datasets from different application domains, including more complex label combinations. We anticipate that employing such label subsets in the mixture model framework will be crucial to effectively integrate features from different sources, for example, from text and non-text data, and improve multi-label classification performance.

REFERENCES

[1] A. Alessandro, G. Corani, D. Mauá, and S. Gabaglio, 'An ensemble of Bayesian networks for multilabel classification', in *International Joint Conference on Artificial Intelligence*, pp. 1220–1225, (2013).

[2] S. Briesemeister, J. Rahnenfuhrer, and O. Kohlbacher, 'Going from where to why – interpretable prediction of protein subcellular localization', *Bioinformatics*, **26**(9), 1232–1238, (2010).

[3] K. Chou and H. Shen, 'Euk-mPLoc: A fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites', *Journal of Proteome Research*, **6**(5), 1728–1734, (2007).

[4] M. DeGroot and M. Schervish, *Probability and Statistics*, Pearson Education, New Jersey, USA, 4th edn., 2012.

[5] K. Dembczynski, W. Cheng, and E. Hüllermeier, 'Bayes optimal multilabel classification via probabilistic classifier chains', in *International Conference on Machine Learning*, pp. 279–286, (June 2010).

[6] K. Dembczynski, W. Waegeman, and E. Hüllermeier, 'An analysis of chaining in multi-label classification.', in *European Conference on Artificial Intelligence*, volume 242, pp. 294–299, (2012).

[7] A. Dempster, N. Laird, and D. Rubin, 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society, Series B*, **39**(1), 1–38, (1977).

[8] J. Doppa, J. Yu, C. Ma, A. Fern, and P. Tadepalli, 'HC-search for multi-label prediction: An empirical study.', in *AAAI Conference on Artificial Intelligence*, pp. 1795–1801, (2014).

[9] U. Fayyad and K. Irani, 'Multi-interval discretization of continuous-valued attributes for classification learning', in *International Joint Conference on Artificial Intelligence*, pp. 1022–1029, (1993).

[10] Y. Guo and S. Gu, 'Multi-label classification using conditional dependency networks.', in *International Joint Conference on Artificial Intelligence*, pp. 1300–1305, (2011).

[11] P. Horton, K. Park, T. Obayashi, N. Fujita, H. Harada, C. Adams-Collier, and K. Nakai, 'WoLF PSORT: Protein localization predictor', *Nucleic Acids Research*, **35**(Web Server issue), W585–W587, (2007).

[12] H. Lin, C. Chen, T. Sung, S. Ho, and W. Hsu, 'Protein subcellular localization prediction of eukaryotes using a knowledge-based approach', *BMC Bioinformatics*, **10**(Suppl 15), 8, (2009).

[13] A. McCallum, 'Multi-label text classification with a mixture model trained by em', in *AAAI Workshop on Text Learning*, (1999).

[14] J. Read, L. Martino, and D. Luengo, 'Efficient monte carlo optimization for multi-label classifier chains.', in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3457–3461, (2013).

[15] J. Read, B. Pfahringer, G. Holmes, and E. Frank, 'Classifier chains for multi-label classification', *Machine Learning*, **85**(3), 333–359, (2011).

[16] A. Romero and L. de Campos, 'A probabilistic methodology for multilabel classification.', *Intell. Data Anal.*, **18**(5), 911–926, (2014).

[17] T. Rubin, A. Chambers, P. Smyth, and M. Steyvers, 'Statistical topic models for multi-label document classification', *Machine Learning*, **88**(1-2), 157–208, (2012).

[18] S. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach*, Pearson Education, New Jersey, USA, 3rd edn., 2010.

[19] R. Simha, S. Briesemeister, O. Kohlbacher, and H. Shatkay, 'Protein (multi-) location prediction: utilizing interdependencies via a generative model', *Bioinformatics*, **12**, i365–i374, (2015).

[20] A. Smith, J. Yu, T. Smulders, A. Hartemink, and E. Jarvis, 'Computational inference of neural information flow networks', *PLoS Computational Biology*, **2**(11), e161, (2006).

[21] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, 'Multilabel classification of music into emotions', in *International Conference on Music Information Retrieval*, pp. 325–330, (2008).

[22] G. Tsoumakas, I. Katakis, and I. Vlahavas, 'Mining multi-label data', in *Data Mining and Knowledge Discovery Handbook*, pp. 667–685, (2010).

[23] H. Wang, M. Huang, and X. Zhu, 'A generative probabilistic model for multi-label classification', in *International Conference on Data Mining*, pp. 628–637, (2008).

[24] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques.*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edn., 2011.

[25] J. Zaragoza, L. Sucar, E. Morales, C. Bielza, and P. Larrañaga, 'Bayesian chain classifiers for multidimensional classification', in *International Joint Conference on Artificial Intelligence*, pp. 2192–2197, (2011).

[26] M. Zhang and K. Zhang, 'Multi-label learning by exploiting label dependency', in *International Conference on Knowledge Discovery and Data Mining*, pp. 999–1008, (2010).

[27] S. Zhang, X. Xia, J. Shen, Y. Zhou, and Z. Sun, 'DBMLoc: A Database of proteins with multiple subcellular localizations', *BMC Bioinformatics*, **9**, 127, (2008).