Noname manuscript No.
(will be inserted by the editor)

Viterbi Classifier Chains for Multi-Dimensional Learning

L. Martino[⋆] · **J. Read**[†] · **F. Louzada**[⋆]

Received: date / Accepted: date

Abstract Multi-dimensional classification (also known variously as multi-target, multi-objective, and multi-output classification) is the supervised learning problem where an instance is associated to qualitative discrete variables (a.k.a. *labels*), rather than with a single class, as in traditional classification problems. Since these classes are often strongly correlated, modeling the dependencies between them allows MDC methods to improve their performance – at the expense of an increased computational cost.

A popular method for multi-label classification is the classifier chains (CC), in which the predictions of individual classifiers are cascaded along a chain, thus taking into account inter-label dependencies. Different variant of CC methods have been introduced, and many of them perform very competitively across a wide range of benchmark datasets. However, scalability limitations become apparent on larger datasets when modeling a fully-cascaded chain. In this work, we present an alternative model structure among the labels, such that the Bayesian optimal inference is then computationally feasible. The inference is efficiently performed using a Viterbi-type algorithm. As an additional contribution to the literature we analyze the relative advantages and interaction of three aspects of classifier chain design with regard to predictive performance versus efficiency: finding a good chain structure vs. a random structure, carrying out complete inference vs. approximate or greedy inference, and a linear vs. non-linear base classifier. We show that our Viterbi CC can perform best on a range of real-world datasets.

Keywords classifier chains · multi-dimensional classification · multi-label classification · Viterbi algorithm

[⋆] Institute of Mathematical Sciences and Computing (ICMC-USP; at Sao Carlos). Brazil.
[†] Department of Information and Computer Science. Aalto University. Helsinki FI-00076. Finland.

1 Introduction

Multi-dimensional classification (MDC) is the supervised learning problem where an instance is associated with multiple qualitative variables, called *labels*. MDC is also known in the literature as multi-target, multi-output or multi-objective classification. The task of multi-label classification (MLC) can be viewed as a particular case of the MDC that only involves binary labels. There are a vast range of active applications of MLC, including tagging images, categorizing documents, and labelling video and other media, and learning the relationship among genes and biological functions (see [10, 1, 6, 9] for overviews). Note that any MLC dataset can be converted into an MDC dataset and vice versa, along the same logic that any number has both a binary and decimal representation: an MDC label taking up to four distinct values, can be represented equivalently as two binary labels.

In literature, there exists two main strategies for tackling a MDC problem [10]: *direct approach* where an algorithm is directly designed or adapted for handling MDC, or the *problem transformation approach* where a MDC problem is converted into a multi-class (single-label) problem. In this work, we focus on this second class.

A basic transformation approach to MDC is the *independent classifiers* (IC) method, (commonly known as *binary relevance* in the multi-label literature), which decomposes the MDC problem into a set of standard single-label classification problems (each label becomes a separate problem) and uses a classifier for each label variable (e.g., a logistic regressor or a support vector machine) separately. Unfortunately, although IC has a low computational cost, it obtains unsatisfactory performance on many data sets and performance measures, because it does not take into account the dependencies between labels [6, 12]. A main challenge in MDC is the design of efficient classification schemes that can take into account label dependencies and still deal with the scale of real-world problems.

An improvement over IC is that of *classifier chains* (CC), which improves the performance of IC by constructing a sequence of classifiers that make use of previous outputs of the chain. The original CC method [8] performs a greedy approximation, and is fast (similar to IC in terms of complexity) but is susceptible to error propagation along the chain of classifiers.

A CC-based Bayes-optimal method, probabilistic classifier chains (PCC), was proposed by [2]. However, although it improves the performance of CC, its computational cost is too large for most real-world applications. Some approaches have been proposed to reduce the computational cost of PCC at test time [12, 5, 3, 7], but the problem is still open.

In this paper we introduce a novel method that attain the performance of PCC, but remains tractable for high-dimensional data sets both at training and test times. Our approach considers a simpler dependence structure among the labels. This simplification allows a more efficient exhaustive exploration of the possible combinations of label values, using the well-known Viterbi algorithm [11]. Another advantage of the proposed algorithms is that predictive performance can be traded off for scalability depending on the application.

2 Problem statement: Multi-Dimensional Classification

Let us assume that we have a set of training data composed of N labelled examples, $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$, where

$$\mathbf{x}^{(n)} = [x_1^{(n)}, \ldots, x_D^{(n)}]^\top \in \boldsymbol{\mathcal{X}} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_D \subseteq \mathbb{R}^D$$

is the n-th feature vector (input), and

$$\mathbf{y}^{(n)} = [y_1^{(n)}, \ldots, y_L^{(n)}]^\top \in \boldsymbol{\mathcal{Y}} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_L \subset \mathbb{N}_+^L$$

is the n-th label vector (output), with $y_\ell^{(n)} \in \mathcal{Y}_\ell = \{1, \ldots, K_\ell\}$, and $K_\ell \in \mathbb{N}_+$ being the finite number of classes associated to the ℓ-th label. The goal of MDC is learning a classification function,[1]

$$\mathbf{h} = [h_1, \ldots, h_L]^\top : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}.$$

Let us assume that the unknown *true* posterior probability density function (pdf) of the data is $p(\mathbf{y}|\mathbf{x})$. From a Bayesian point of view, the optimal label assignment for a given test instance, $\mathbf{x}^*$, is provided by the maximum a posteriori (MAP) label estimate,

$$\hat{\mathbf{y}}_{\mathrm{MAP}} = \mathbf{h}_{\mathrm{MAP}}(\mathbf{x}^*) = \underset{\mathbf{y} \in \boldsymbol{\mathcal{Y}}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x}^*), \tag{1}$$

where the search must be performed over all possible test labels, $\mathbf{y} \in \boldsymbol{\mathcal{Y}}$. The MAP label estimate is the one most commonly used in the literature, although other approaches are possible, as shown in [2]. Unfortunately, the problem is further complicated by the fact that the true density, $p(\mathbf{y}|\mathbf{x})$, is usually unknown, and the classifier has to work with an approximation, $\hat{p}(\mathbf{y}|\mathbf{x})$, constructed from the training data. Hence, the (possibly sub-optimal) label prediction is finally given by

$$\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x}^*) = \underset{\mathbf{y} \in \boldsymbol{\mathcal{Y}}}{\operatorname{argmax}}\ \hat{p}(\mathbf{y}|\mathbf{x}^*). \tag{2}$$

Table 1 summarizes the main notation used this work.

Figure 1 clarifies the relationship among MDC, MLC and the ‘standard’ binary and multi-class classification tasks. For instance, in MDC w.r.t. MLC, there is a higher dimensionality (for the same value of L); MLC deals with 2^L possible values, whereas MDC deals with $\prod_{\ell=1}^{L} K_\ell$. Figure 2 depicts an

[1] We consider $\mathbf{h}$ as a vector because this fits naturally into the independent classifier and classifier chain context, but this is not universal, and $h : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$ is possible in other contexts (such as LP)

Notation	Description
$\mathbf{x} = [x_1, \ldots, x_D]^\top \in \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^D$	D-dimensional feature/instance.
$\mathbf{y} = [y_1, \ldots, y_L]^\top \in \boldsymbol{\mathcal{Y}} \subset \mathbb{N}_+^L$	L-dimensional label vector.
$\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^N$	Training data set, $n = 1, \ldots, N$.
$p(\mathbf{y}\|\mathbf{x})$	Unknown true posterior pdf.
$\hat{p}(\mathbf{y}\|\mathbf{x})$	Empirical pdf obtained by the classifier.
$\mathbf{x}^* = [x_1^*, \ldots, x_D^*]^\top \in \boldsymbol{\mathcal{X}}$	Test feature vector.
$\mathbf{h} = [h_1, \ldots, h_L]^\top : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{Y}}$	Classification function built from $\mathcal{D}$.
$\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x}) = [\hat{y}_1, \ldots, \hat{y}_L]^\top$	Generic classifier's output.

Table 1: Main notation.

example of MLC scenario. In multi-class classification, only a single label is predicted,

$$h(\mathbf{x}) : \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^D \to \mathcal{Y} \subseteq \mathbb{N}_+,$$

where $y \in \mathcal{Y} = \{1, \ldots, K\}$. If $K = 2$ the learning problem becomes binary classification (also known as filtering in the case of textual and web data). Observe that MDC can be seen as a multi-class problem with $\prod_{\ell=1}^L K_\ell$ classes. Each class is denoted as specific sequence of L integers $[y_1', \ldots, y_L']^\top \in \boldsymbol{\mathcal{Y}} \subseteq \mathbb{N}_+^L$. Clearly, this approach becomes quickly unfeasible when L or K_ℓ grows.

We have already noted that with $K_\ell = 2$, for all $\ell = 1, \ldots, L$, MDC becomes MLC. This can also be interpreted as a multi-class problem where the label takes one of 2^L possible values. For instance, in Figure 2 we can interpret that the hexagons belong jointly to a class 6 (a decimal representation of 110).

	$K = 2$	$K > 2$
$L = 1$	**binary**	**multi-class**
$L > 1$	**multi-label** (MLC)	**multi-dimensional** (MDC)

Fig. 1: Different classification paradigms: L is the number of *labels* and K is the number of *values* that each label variable can take.

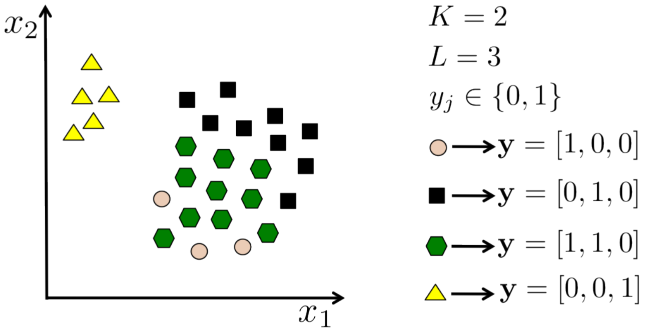


Fig. 2: Toy example of MDC with $K = 2$ (then it coincides with MLC, in this case) possible values for each label and $L = 3$ labels (thus $y_j \in \{0, 1\}$ for $j = 1, 2, 3$) and $D = 2$ features.

3 Transformation strategies for MDC problem

In literature, a major approach to tackle the MDC problem is that of *problem transformation* [10] where a MDC problem is transformed into a multi-class (single-label) problem and, as a consequence, then a single-label learning algorithm is applied (of which there already exists a plethora of exemplars in the literature).

3.1 Independent Classifiers (IC)

One simple and well-known procedure is the so-called *independent classifiers* (IC) [10,9,8,12]. As the name suggests, given a new instance $\mathbf{x}^*$ each label y_ℓ is classified using a classifier independent from the other labels. Namely, for each $\ell = 1, \dots, L$ a classifier h_ℓ is employed defined as (probabilistically speaking)

$$\hat{y}_\ell = h_\ell(\mathbf{x}^*) = \underset{y_\ell \in \mathcal{Y}_\ell}{\operatorname{argmax}} \ \hat{p}_\ell(y_\ell|\mathbf{x}^*), \tag{3}$$

so that finally $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_L]$ is $\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x}^*) = [h_1(\mathbf{x}^*), \dots, h_L(\mathbf{x}^*)]^\top$. This method is easy to build using off-the-shelf classifiers, but it does not explicitly model label dependencies, and its performance suffers as a result. In fact, it assumes complete independence, i.e., it approximates the density of the data as

$$\hat{p}_\ell(\mathbf{y}|\mathbf{x}) = \prod_{\ell=1}^{L} \hat{p}_\ell(y_\ell|\mathbf{x}). \tag{4}$$

Figure 3(a) corresponds to the IC model.

3.2 Classifier Chains (CC)

The *classifier chains* (CC) method [8] models the correlation among labels by creating a chain of labels, and using earlier labels as additional feature attributes for later feature attributes, in a cascade, as shown in Figure 3(b). Consider the chain rule from probability theory. Given a test instance, $\mathbf{x}^*$, the true label probability, is

$$p(\mathbf{y}|\mathbf{x}^*) = p_1(y_1|\mathbf{x}^*) \prod_{\ell=2}^{L} p_\ell(y_\ell|\mathbf{x}^*, y_1, \dots, y_{\ell-1}). \tag{5}$$

This probabilistic formulation w.r.t. CC was first given by [2] in what they called probabilistic classifier chains (PCC). Theoretically, label order is irrelevant in Eq. (5), as all the label orderings result in the same pdf. However, since in practice we are modelling an approximation of p (i.e., $\hat{p}$), label order can be important for attaining a good classification performance (see [7] and references therein). So, to make it clear, PCC approximates the true data density as

$$\hat{p}(\mathbf{y}|\mathbf{x}^*) = \hat{p}_1(y_1|\mathbf{x}^*) \prod_{\ell=2}^{L} \hat{p}_\ell(y_\ell|\mathbf{x}^*, y_1, \ldots, y_{\ell-1}), \tag{6}$$

where each conditional probability $\hat{p}$ is learnt by the used classifier during the training stage, thus effectively constructing a chain of classifiers: the ℓ-th classifier considers the vector $[\mathbf{x}^*, y_1, \ldots, y_L]^\top$ as its input (i.e., as instance).

PCC carries out an optimal search of Eq. (6) by trialling each possible $\mathbf{y}$ in

$$\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x}^*) = \underset{\mathbf{y}\in\boldsymbol{\mathcal{Y}}}{\text{argmax}} \ \hat{p}(\mathbf{y}|\mathbf{x}^*), \tag{7}$$

The original formulation of CC used a rough but fast greedy approximation of this where, given a new (test) instance, $\mathbf{x}^*$, CC simply predicts $\hat{y}_\ell$ from $\mathbf{x}^*$ and all the previous predictions $(\hat{y}_1, \ldots, \hat{y}_{\ell-1})$, i.e.,

$$\hat{y}_\ell = h_\ell(\mathbf{x}^*, \hat{y}_1, \ldots, \hat{y}_{\ell-1}) = \underset{y_\ell\in\mathcal{Y}_\ell}{\text{argmax}} \ \hat{p}_\ell(y_\ell|\mathbf{x}^*, \hat{y}_1, \ldots, \hat{y}_{\ell-1}). \tag{8}$$

which essentially means following a single path of labels $\mathbf{y}$ greedily down the chain of L binary classifiers. This is shown in Figure 4 through a simple example. Mathematically speaking, CC maximizes separately each conditional pdf $\hat{p}(y_\ell|\mathbf{x}^*, \hat{y}_1, \ldots, \hat{y}_{\ell-1})$ (see Eq. (8)), which is a sub-optimal strategy w.r.t. maximize the joint pdf $\hat{p}(\mathbf{y}|\mathbf{x}^*)$.

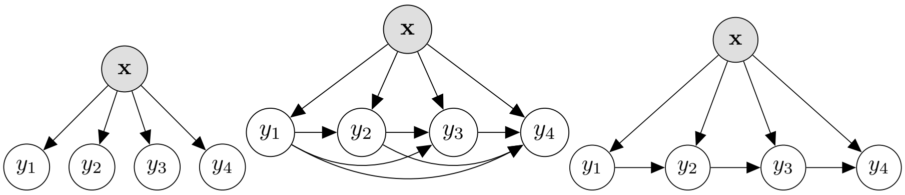
Note that CC and PCC are identical in the training phase. Both can be represented by Figure 3(b). In [2] an overall improvement of PCC over CC is reported, but at the expense of a high computational complexity: it is intractable for more than about 10 labels ($\equiv 2^{10}$ paths), which represents the majority of practical problems in the multi-label domain (clearly, the results can also depend on the chosen label order as in CC). Figure 4 depicts an example of inference by using CC and PCC: the red dashed path corresponds to the best path representing the decision of PCC (i.e., analyzing all the possible paths), whereas the green dotted path corresponds to the decision of CC.

Several 'in-between' methods have been proposed which provide a closer probabilistic approximation at inference time, while still retaining tractability. We mention these in the next section.

4 Viterbi Classifier Chain

A number of search variants have been proposed for classifier chains, including: [3]'s ϵ-approximate inference, based on performing a depth-first search in the probabilistic tree with a cutting-off list; 'beam search' [5], a heuristic search algorithm that speeds up inference considerably; and Monte Carlo search [3, 4, 7].

In this paper we consider a simplified chain structure, which still takes in account the label dependence but allows the Bayes-optimal inference in a



(a)

(b)

(c)

Fig. 3: Graphical models ($L = 3$) of **(a)** Independent Classifiers (IC), **(b)** of Classifier Chains (CC) and **(c)** of the Viterbi Classifier Chains (VCC).

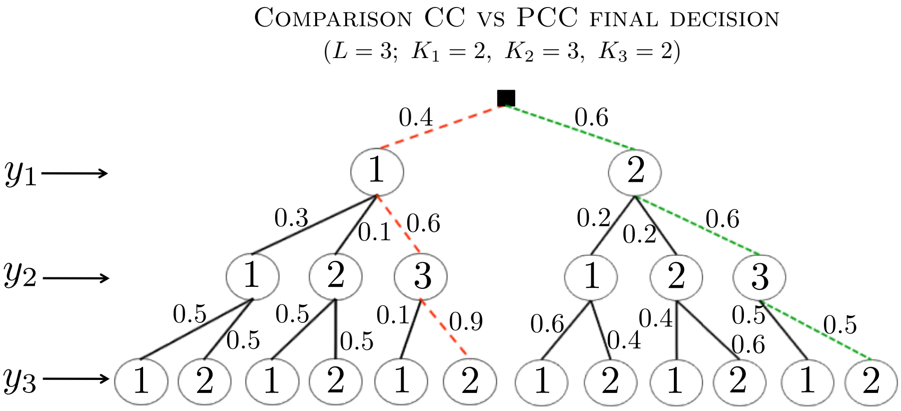


Fig. 4: Example of the $\prod_{\ell=1}^{L} K_\ell = K_1 \times K_2 \times K_3 = 2 \times 3 \times 2 = 12$ possible paths along the tree of class labels y_ℓ ($\ell = 1, \ldots, L = 3$). The best path, $\hat{\mathbf{y}}_{PCC} = [1, 3, 2]^\top$, with probability 0.2160, is shown with dashed red lines. This path represents the decision of PCC. The suboptimal path $\hat{\mathbf{y}}_{CC} = [1, 2, 3]^\top$ represents the output of CC (dotted green line).

faster and more scalable way than PCC. This simpler graphical model is shown in Figure 3(c). Namely, given a specific sequence $y_1, \ldots, y_L$, in this case we consider the simplified factorization

$$\hat{p}(\mathbf{y}|\mathbf{x}^*) = \hat{p}_1(y_1|\mathbf{x}^*) \prod_{\ell=2}^{L} \hat{p}_\ell(y_\ell|\mathbf{x}^*, y_{\ell-1}). \tag{9}$$

Note that this expression is more sophisticated than Eq. (4) (taking in account the dependence among labels) and simpler than Eq. (6) (we only consider a Markov dependence).

In this case, the tree in Fig. 4 can be transformed in a Trellis diagram as shown in Figure 5. Hence, the goal is to find the optimal label vector (as in

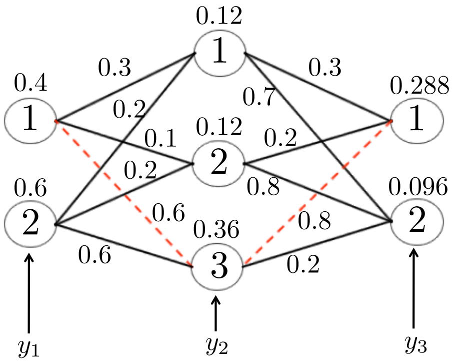


Fig. 5: Example of Trellis diagram corresponding to VCC, with 3 class labels y_ℓ ($\ell = 1, \ldots, L = 3$) and $K_1 = 2$, $K_2 = 3$, $K_2 = 2$. The best path, $\hat{\mathbf{y}}_{VCC} = [1, 3, 1]^\top$, with probability 0.2880, is shown with dashed red lines.

PCC)

$$\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x}^*) = \underset{\mathbf{y} \in \boldsymbol{\mathcal{Y}}}{\operatorname{argmax}} \ \hat{p}(\mathbf{y}|\mathbf{x}^*), \tag{10}$$

where $\hat{p}(\mathbf{y}|\mathbf{x}^*)$ is given in Eq. (9). In this case, the vector $\hat{\mathbf{y}}$ coincides with the optimal path in the Trellis diagram in Figure 5 and it is well-known that this path can be efficiently obtained using the *Viterbi algorithm* [11]. From the point of view of Hidden Markov Model (HMM), the states are represented by the different values that each label can get. The metric of each branch is defined by $\hat{p}_\ell(y_\ell|\mathbf{x}^*, y_{\ell-1})$, provided by a multi-class classifier. Observe that the transition probability $\hat{p}_\ell(y_\ell|\mathbf{x}^*, y_{\ell-1})$ depends on the index ℓ. Note that, this can consider this scenario as a special case of a generic HMM since we always "observe" the same instance $\mathbf{x}^*$, for all $\ell = 1, \ldots, L$.

The resulting Viterbi Classifier Chain (VCC) algorithm is summarized in Table 2.

As the numerical results show, VCC provide a good trade-off between the description of the label dependence and the scalability of the inference.

In the previous section we mentioned the potential importance of *order* or sequence of the chain (in which order the labels appear in the chain, chosen at training time). We do not focus specifically on this issue, but many of the same search techniques used at inference time can also be used to search the space of chain orders. In the experimental evaluation we consider that of Monte Carlo search, both for existing verities of classifier chains, and our presented Viterbi classifier chain. This task can be done separately to inference (namely, at training time).

```
1. Initialization:
      - Obtain $\hat{p}_1(y_1|\mathbf{x}^*)$ by a multi-class classifier.
      - Set $\delta_1(i) = \hat{p}_1(y_1 = i|\mathbf{x}^*)$ and $\psi_1(i) = 0$ with $i = 1, \ldots, K_1$.
2. Recursion:
  For $\ell = 2, \ldots, L$ :
    For $j = 2, \ldots, K_\ell$ :
      - Obtain $\hat{p}_\ell(y_\ell|\mathbf{x}^*, y_{\ell-1})$ by a multi-class classifier.
      - Set
          $\delta_\ell(j) = \max_{1 \leq i \leq K_\ell} \delta_{\ell-1}(i)\hat{p}_\ell(y_\ell = j|\mathbf{x}^*, y_{\ell-1} = i)$,
          $\psi_\ell(j) = \arg\max_{1 \leq i \leq K_\ell} \delta_{\ell-1}(i)\hat{p}_\ell(y_\ell = j|\mathbf{x}^*, y_{\ell-1} = i)$.
3. Output ($\hat{\mathbf{y}}_{VCC} = [\hat{y}_1, \ldots, \hat{y}_\ell, \ldots, \hat{y}_L]^\top$) :
      - $\hat{y}_L = \arg\max_{1 \leq i \leq K_\ell} \delta_L(i)$.
      - $\hat{y}_\ell = \arg\max_{1 \leq i \leq K_\ell} \phi_L(\hat{y}_{\ell-1})$.
```

Table 2: Viterbi Classifier Chain (VCC).

Table 3: Evaluation under Jaccard Index, logistic regression is the base classifier.

Dataset	IC	CC	MCC	VCC	MsCC	VsCC
Music	0.536 5	0.572 2	0.585 1	0.554 4	0.567 3	0.536 5
Scene	0.603 6	0.699 3	0.710 1	0.644 5	0.705 2	0.647 4
Yeast	0.502 6	0.526 1	0.526 1	0.512 5	0.520 3	0.514 4
Medical	0.691 4	0.699 3	0.700 2	0.691 4	0.726 1	0.691 4
Enron	0.337 4	0.354 2	0.355 1	0.337 4	0.351 3	0.334 6
avg rank	5.00	2.20	1.20	4.40	2.40	4.60

Table 4: Evaluation under Jaccard Index, random forest of decision trees as a base classifier.

Dataset	IC	CC	VCC	MsCC	VsCC
Music	0.552 5	0.588 3	0.576 4	0.589 2	0.597 1
Scene	0.665 5	0.697 3	0.694 4	0.726 2	0.727 1
Yeast	0.511 5	0.545 3	0.543 4	0.558 1	0.557 2
Medical	0.609 5	0.637 3	0.633 4	0.665 2	0.680 1
Enron	0.484 3	0.466 5	0.471 4	0.488 2	0.489 1
avg rank	4.60	3.40	4.00	1.80	1.20

5 Empirical Evaluation

We use and implement methods in the MEKA framework (`http://meka.sourceforge.com`), in a relatively standard setup on a number of commonly-used multi-label datasets. Details on datasets, evaluation metrics, and other benchmark methods, can be found in [7]. Here we only display results under the evaluation metric *Jaccard index*, $\frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{y}^{(n)} \wedge \hat{\mathbf{y}}^{(n)}|}{|\mathbf{y}^{(n)} \vee \hat{\mathbf{y}}^{(n)}|}$, but we found no difference in relative results under other metrics. The methods denoted with 's' trial 50 random chains (i.e., label orders) and elect the best one for inference. Results are displayed in Table 3 and Table 4, with two different *base classifiers* (the individual models for each label).

6 Discussion and Conclusions

Table 3 indicates that chain connectivity is more important than exact inference: the fully cascaded CC and MCC outperform VCC, even though VCC is the only one capable of exact inference (at a cost of less connectivity). Even VsCC, with chain search, cannot compete with even basic CC. Of course, the number of extra attributes in the final classifier (corresponding to the number of incoming links) is $D+L-1$ in CC, but only $D+1$ in VCC. We further note that when searching the chain space (say, by swapping two label indices at each proposal step like in [7]), VsCC only needs to rebuild the models in the sub-sequence between the first and last index, whereas MsCC must rebuild all models to the right of the left-most index; implying a much faster search of the space of chain-orders.

Nevertheless, even after providing encouraging results on time complexity, it would be difficult to argue in favor for a Viterbi classifier chain wrt Table 3. However, when we use a non-linear base learner in Table 4 (namely, random decision-tree forests), results change markedly: the performance of V(s)CC is virtually indistinguishable from M(s)CC. By using a stronger base classifier, dependence among the labels is reduced. In other words, each base model is able to make a better decision using only the input features and is thus becomes less-dependent on the other labels as features. Of course, many non-linear machine learning methods come at a more computationally intensive cost than more basic linear learners.

In summary, it is an interesting discovery that in multi-label problems, exact inference does not outperform greedy or approximate inference with greater connectivity. Nevertheless, when connectivity is limited based on scalability reasons (namely, a large numbers of labels) VCC offers some interesting options, as it scales linearly with the number of labels, and furthermore major savings can be made in finding an appropriate chain order. But in this case, the suitability of the base classifier should be carefully to avoid *introducing* dependence inadvertently. Current work has not studied this interaction between base classifier and chain connectivity and inference, although our results reveal that it is a fundamental interaction, and should be investigated further in future work.

Acknowledgements

This work was supported by the Grant 2014/23160-6 of São Paulo Research Foundation (FAPESP) and the Aalto University AEF research program (`http://energyefficiency.aalto.fi/en/`).

References

1. André C.P.L.F. Carvalho and Alex A. Freitas. A tutorial on multi-label classification techniques. In Ajith Abraham, Aboul-Ella Hassanien, and Vclav Snel, editors, *Foundations of Computational Intelligence Volume 5*, volume 205 of *Studies in Computational Intelligence*, pages 177–195. Springer, 2009.

2. Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *27th International Conference on Machine Learning (ICML)*, Haifa, Israel, June 2010.
3. Krzysztof Dembczyński, Willem Waegeman, and Eyke Hüllermeier. An analysis of chaining in multi-label classification. In *Workshop Proc. of 20th European Conf. on Artificial Intelligence (ECAI)*, pages 294–299, Montpellier, France, August 27–31 2012.
4. Krzysztof J. Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An exact algorithm for F-measure maximization. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 24*, pages 1404–1412. 2011.
5. Abhishek Kumar, Shankar Vembu, Aditya Krishna Menon, and Charles Elkan. Learning and inference in probabilistic classifier chains with beam search. In *Machine Learning and Knowledge Discovery in Databases*, volume 7523, pages 665–680, 2012.
6. Jesse Read. *Scalable Multi-label Classification.* PhD thesis, University of Waikato, 2010.
7. Jesse Read, Luca Martino, and David Luengo. Efficient Monte Carlo methods for multi-dimensional learning with classifier chains. *Pattern Recognition*, 47(3):1535–1546, 2014.
8. Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.
9. G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook.* 2nd edition, Springer, 2010.
10. Grigorios Tsoumakas and Ioannis Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.
11. A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 2(13):260–269, 1967.
12. Julio H. Zaragoza, Luis Enrique Sucar, Eduardo F. Morales, Concha Bielza, and Pedro Larrañaga. Bayesian chain classifiers for multidimensional classification. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2011.