EFFICIENT MONTE CARLO OPTIMIZATION FOR MULTI-LABEL CLASSIFIER CHAINS

Jesse Read, Luca Martino[*]

Dept. of Signal Theory and Communications
Univ. Carlos III de Madrid (Spain)

David Luengo[*]

Dept. of Circuits and Systems Engineering
Univ. Politécnica de Madrid (Spain)

ABSTRACT

Multi-label classification (MLC) is the supervised learning problem where an instance may be associated with multiple labels. Modeling dependencies between labels allows MLC methods to improve their performance at the expense of an increased computational cost. In this paper we focus on the classifier chains (CC) approach for modeling dependencies. On the one hand, the original CC algorithm makes a greedy approximation, and is fast but tends to propagate errors down the chain. On the other hand, a recent Bayes-optimal method improves the performance, but is computationally intractable in practice. Here we present a novel double-Monte Carlo scheme (M2CC), both for finding a good chain sequence and performing efficient inference. The M2CC algorithm remains tractable for high-dimensional data sets and obtains the best overall accuracy, as shown on several real data sets with input dimension as high as 1449 and up to 103 labels.

Index Terms— multi-label classification; Monte Carlo methods; classifier chains

1. INTRODUCTION

Multi-label classification (MLC) is the supervised learning problem where an instance may be associated with multiple labels, rather than with a single label as in traditional binary or multi-class single-label classification (SLC) problems. The MLC learning context is receiving increased attention in the literature, since it arises naturally in a wide variety of domains: text, audio, still images and video, bioinformatics, etc. [1, 2]. The main challenge in this area is modeling label dependencies without incurring in an intractable complexity.

A basic approach to MLC is provided by the so-called *binary relevance* (BR) method, which decomposes the MLC problem into a set of SLC problems (one per label) and uses a separate classifier for each label. In this way, the multi-label problem is turned into a series of standard binary classification problems that can be solved with any off-the-shelf binary classifier (e.g., a logistic regressor or a support vector machine). Unfortunately, although BR has a low computational cost, it cannot provide high performance, because it does not model dependencies between labels [2, 3, 4, 5, 6].

In order to model dependencies explicitly, several alternative schemes have been proposed, such as the so-called *label powerset* (LP) method [7]. LP considers each potential combination of labels in the MLC problem as a single label. In this way, the multi-label problem is turned into a traditional multi-class problem that can be solved using standard methods. Unfortunately, given the huge number of class values produced by this transformation, this method is usually unfeasible for practical application, and suffers from issues like overfitting. This was recognised by [3, 8], which provide approximations to the LP scheme that reduce these problems, although such methods have been superseded in recent years.

A more recent idea is using classifier chains (CC), which improves the performance of BR and LP by constructing a sequence of classifiers that make use of previous outputs of the chain. The original CC method, introduced in [4] and extended in [5, 9], makes a greedy approximation, and is fast but tends to propagate errors down the chain. Nevertheless, a very recent extensive experimental comparison reaffirmed that CC is among the highest-performing methods for MLC, and recommended it as a benchmark algorithm [10]. A CC-based Bayes-optimal method, probabilistic classifier chains (PCC), has also been recently proposed [5]. However, although it improves the performance of CC, its computational cost is too large for most real-world applications.

In this paper we introduce a novel method that attains the performance of PCC, but remains tractable for high-dimensional data sets. Our approach (M2CC) is based on a double Monte Carlo optimization technique and, unlike all other chain-based methods in the literature, it explicitly searches the space of possible chain-sequences during the training stage. Hence, predictive performance can be traded off for scalability depending on the application.

The paper is organized as follows. In Section 2 we review multi-label classification and the important developments leading up to this paper. In Section 3 we detail our proposed novel methods. In Section 4 we carry out empirical evaluations. Finally, in Section 5 we draw some conclusions and mention possible future work.

2. MULTI-LABEL CLASSIFICATION (MLC)

Let us assume that we have a set of training data composed of N labelled examples, $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$, where $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_D^{(i)}]^{\top}$ is the i-th D-dimensional instance (input), with $x_d^{(i)} \in \mathcal{X}_d$ for $1 \leq d \leq D$, and $\mathbf{y}^{(i)} = [y_1^{(i)}, \ldots, y_L^{(i)}]^{\top}$ is the i-th example's $L \times 1$ label relevance vector (output), with $y_j^{(i)} \in \{0, 1\}$ being its j-th label assignment (1 iff the label is relevant to $\mathbf{x}^{(i)}$, 0 otherwise).

In MLC we seek to learn a function, $\mathbf{y} = \mathbf{h}(\mathbf{x})$, that assigns a vector of labels, $\mathbf{y} \in \{0, 1\}^L$, to each instance, $\mathbf{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$. Let us assume that the true distribution of the data is $f(\mathbf{y}|\mathbf{x})$. From a Bayesian point of view, the optimal label assignment (i.e., the one with the largest probability of being the true one) for a given test instance, $\mathbf{x}^*$, is provided by the maximum a posteriori (MAP) label estimate:

$$\mathbf{y}_{\text{MAP}}^* = \mathbf{h}_{\text{MAP}}(\mathbf{x}^*) = \underset{\mathbf{y}}{\operatorname{argmax}} f(\mathbf{y}|\mathbf{x}^*). \qquad (1)$$

Unfortunately, the true distribution, $f(\mathbf{y}|\mathbf{x})$, is usually unknown, and the classifier has to work with an approximation, $p(\mathbf{y}|\mathbf{x})$, constructed from the training data. Hence, the (possibly sub-optimal) label prediction is finally given by

$$\mathbf{y}^* = \mathbf{h}(\mathbf{x}^*) = \underset{\mathbf{y}}{\operatorname{argmax}} p(\mathbf{y}|\mathbf{x}^*). \qquad (2)$$

2.1. Classifier Chains (`CC`)

Classifier chains (`CC`) is based on modeling the correlation among labels using the chain rule of probability. Given a data instance, $\mathbf{x}$, and a vector of label indexes, $\mathbf{s} = [s_1, \ldots, s_L]^{\top}$, obtained as a permutation of $\{1, \ldots, L\}$, $p(\mathbf{y}|\mathbf{x}, \mathbf{s})$ may be expressed as[1]

$$p(\tilde{\mathbf{y}}|\mathbf{x}^*, \mathbf{s}) = p(\tilde{y}_1|\mathbf{x}^*) \prod_{j=2}^{L} p(\tilde{y}_j|\mathbf{x}^*, \tilde{y}_1, \ldots, \tilde{y}_{j-1}), \qquad (3)$$

where $\tilde{\mathbf{y}} = [\tilde{y}_1, \ldots, \tilde{y}_L]^{\top}$ is the permuted label vector, $\tilde{y}_j = y_{s_j}$ is the j-th label in the permutation, and the probabilities in (3) are learnt from the labelled data during the training stage.

During the test stage, `CC` follows a single path greedily down the chain of L binary classifiers, with the j-th classifier, h_j, predicting the j-th label's relevance, $\tilde{y}_j^*$, using the test instance, $\mathbf{x}^*$, and all previous predictions $(\tilde{y}_1^*, \ldots, \tilde{y}_{j-1}^*)$, as

$$\tilde{y}_j^* = h_j(\mathbf{x}^*|\mathbf{s}) = \underset{\tilde{y}_j}{\operatorname{argmax}} p(\tilde{y}_j|\mathbf{x}^*, \tilde{y}_1^*, \ldots, \tilde{y}_{j-1}^*). \qquad (4)$$

In carrying out classification down a chain in this way, `CC` models label dependencies and, as a result, usually performs much better than `BR`, while being similar in memory and time requirements in practice. However, due to its greedy approach it is susceptible to errors in the initial links of the chain [5].

---

[1] Theoretically, Eq. (3) does not depend on the label order. However, since all the probabilities in (3) are estimated from the training data, the label order can have a large effect in practice, as recognized by [5].

2.2. Probabilistic Classifier Chains (`PCC`)

Probabilistic classifier chains (`PCC`) was introduced in [5]. In the training phase, `PCC` is identical to `CC`. However, during the test stage `PCC` provides Bayes-optimal inference by exploring all the 2^L possible paths of the chain. Hence, for a given test instance, $\mathbf{x}^*$, `PCC` provides the optimum label estimate, obtained maximizing the label vector, $\mathbf{y}$, rather than the individual labels, y_j:

$$\tilde{\mathbf{y}}^* = \mathbf{h}(\mathbf{x}^*|\mathbf{s}) = \underset{\tilde{\mathbf{y}}}{\operatorname{argmax}} p(\tilde{\mathbf{y}}|\mathbf{x}^*, \mathbf{s}), \qquad (5)$$

where $p(\tilde{\mathbf{y}}|\mathbf{x}^*, \mathbf{s})$ is given by (3). In [5] an overall improvement of `PCC` over `CC` is reported, but at the price of high computational complexity: it is intractable for more than about 10 labels ($\equiv 2^{10}$ paths), which represents the majority of problems in the multi-label domain.

3. EFFICIENT DOUBLE MONTE CARLO TECHNIQUE FOR CLASSIFIER CHAINS

In chain-based MLC problems, for any given test instance, $\mathbf{x}^*$, and label order, $\mathbf{s}$, we wish to find the best label-relevance vector, $\mathbf{y}^* = [\tilde{y}_1^*, \ldots, \tilde{y}_L^*]$, out of the 2^L possible label vectors or paths. However, the best inference on a poor model will not be as good as the best inference on a good model. Therefore, at training time we also wish to find the best chain order or *sequence*, $\mathbf{s} = [s_1, \ldots, s_L]$, out of the $L!$ possible chains.

Unfortunately, the optimal solution of these two problems is not feasible for large values of L. Hence, in this section we introduce an efficient double Monte Carlo strategy for quasi-optimal inference in Classifier Chains. We present both a tractable label prediction scheme at test time (`MCC`) and a method that performs an additional search for the optimal chain sequence at build time (`M2CC`); an issue which, to the best of our knowledge, has not yet been successfully tackled, except by means of avoiding it using a network, such as the conditional dependency network (`CDN`) of [6].

3.1. Training step: finding the best chain

In order to obtain the best chain (i.e., the optimal label order) during the training step we introduce a *payoff function*,

$$J(\mathbf{s}) = \sum_{i=1}^{N} p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \mathbf{s}), \qquad (6)$$

and the optimal sequence, $\hat{\mathbf{s}}$, is the one that maximizes (6) over the set of $L!$ possible sequences, i.e.,

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmax}} J(\mathbf{s}) = \underset{\mathbf{s}}{\operatorname{argmax}} \sum_{i=1}^{N} p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \mathbf{s}). \qquad (7)$$

The exact solution of (7) is intractable even for medium values of L. Therefore, we propose using the Monte Carlo

Algorithm 1 Finding a suitable $\hat{\mathbf{s}}$

Input:
- $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{D}$: training data
- $\pi(\mathbf{s}|\mathbf{s}_{t-1})$: proposal function
- T': number of iterations

Algorithm:
1. Start with some random sequence, $\mathbf{s}_0$, and build an initial model, $p(\mathbf{y}|\mathbf{x}, \mathbf{s}_0)$.
2. For $t = 1, \ldots, T'$:
   - (a) Draw $\mathbf{s}' \sim \pi(\mathbf{s}|\mathbf{s}_{t-1})$ and build model $p(\mathbf{y}|\mathbf{x}, \mathbf{s}')$.
   - (b) `if` $J(\mathbf{s}') > J(\mathbf{s}_{t-1})$
     - $\mathbf{s}_t \leftarrow \mathbf{s}'$ accept.
   - (c) `else`
     - $\mathbf{s}_t \leftarrow \mathbf{s}_{t-1}$ reject.

Output:
- $\hat{\mathbf{s}} = \mathbf{s}_{T'}$: estimated label sequence.

Algorithm 2 Finding $\mathbf{y}^*$ for a given test instance $\mathbf{x}^*$.

Input:
- $\mathbf{x}^*$: test instance.
- $\hat{\mathbf{s}}$: label order (estimated or chosen randomly).
- $p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{s}})$: probabilistic model (from training stage).

Algorithm:
1. Obtain an initial path, $\mathbf{y}_0$, using `CC`.
2. For $t = 1, \ldots, T$:
   - (a) Draw $\mathbf{y}' \sim p(\mathbf{y}|\mathbf{x}^*, \hat{\mathbf{s}})$
   - (b) `if` $p(\mathbf{y}'|\mathbf{x}^*, \hat{\mathbf{s}}) > p(\mathbf{y}_t|\mathbf{x}^*, \hat{\mathbf{s}})$
     - $\mathbf{y}_t \leftarrow \mathbf{y}'$ accept.
   - (c) `else`
     - $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1}$ reject.

Output:
- $\mathbf{y}^* = \mathbf{y}_T$: predicted label assignment.

---

approach summarized in Algorithm 1 to perform an efficient exploration of the label-sequence space. This algorithm starts with a randomly chosen label sequence, $\mathbf{s}_0$, which is then modified trying to find local maximum of the payoff function at least. More specifically, given a sequence $\mathbf{s}_{t-1}$ the proposal function $\pi(\mathbf{s}_t|\mathbf{s}_{t-1})$ consists of choosing uniformly two positions of the label sequence ($1 \le \ell, m \le L$) and swapping the labels corresponding to those positions, so that $\mathbf{s}_t(\ell) = \mathbf{s}_{t-1}(m)$ and $\mathbf{s}_t(m) = \mathbf{s}_{t-1}(\ell - 1)$.

3.2. Inference (test) step: finding the best path $\mathbf{y}^*$

In the test step, for a given test instance, $\mathbf{x}^*$, for which the true label association is unknown, and a label order (either estimated for `M2CC` or randomly chosen for `MCC`), we wish to find the optimal label vector that maximizes (5). In general, this problem can be solved analytically for low values of L by exploring all the 2^L possible paths, as in the `PCC` method [5]. However, when L grows this method quickly becomes computationally intractable. Therefore, we propose here using the random search Monte Carlo approach shown in Algorithm 2 to approximate (5). This algorithm starts from the greedy inference offered by standard `CC`, draws samples $\mathbf{y}^{(i)}$, $i = 1, ..., T$ according to the model $p(\tilde{\mathbf{y}}_t|\mathbf{x}^*, \hat{\mathbf{s}})$, providing a predicted label sequence

$$\mathbf{y}^* = \arg\max p(\tilde{\mathbf{y}}_t^*|\mathbf{x}^*, \hat{\mathbf{s}}), \tag{8}$$

where $\tilde{\mathbf{y}}_t^*$ ($1 \le t \le T$) are the samples accepted by the algorithm.

4. EXPERIMENTS

We perform experiments on a collection of real world data sets familiar in the multi-label literature [3, 4, 5], whose characteristics are shown in Table 1. We compare our two novel methods (`MCC` and `M2CC`) to baseline `BR` [7], the original

Table 1. Multi-label datasets characteristics: n indicates numeric variables; b indicates binary variables, `LC` is *label cardinality*: average number of relevant labels per example.

	N	L	D	LC	Type
Music	593	6	$72n$	1.87	audio
Scene	2407	6	$294n$	1.07	image
Yeast	2417	14	$103n$	4.24	biology
Genbase	661	27	$1185b$	1.25	biology
Medical	978	45	$1449b$	1.25	text
Enron	1702	53	$1001b$	3.38	text
Reuters	6000	103	$500n$	1.46	text

classifier chains method `CC` [4], the Bayes-optimal rendition `PCC` [5]; and also the conditional dependency networks method `CDN` of [6] under $I = 1000$ total iterations. For our methods, we use $T = 100$ (inference $\mathbf{y}$-step) and just $T' = 10$ for `M2CC` (training $\mathbf{s}$-step).[2] As a base classifier we use support vector machines fitted with logistic models in order to have a probabilistic output [11].[3]

We carry out 5-fold cross validation (CV). Results for predictive performance are displayed in Table 2. As a performance measure we have used the *exact match* score (inversely equivalent to subset zero-one loss),

$$\text{EXACT MATCH} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\mathbf{y}^{(i)} = \mathbf{y}^{*(i)}),$$

where $\mathbb{I}(\cdot)$ is an indicator function (returning 1 iff the logical condition is fulfilled and zero otherwise), as this is the loss function minimized by the MAP estimator [5].Results under other measures of evaluation can be seen in [13]. Note that, since `PCC` is only tractable on datasets where $L < 10$, we can

---

[2]Better results can be obtained by increasing T' at the cost of more running time; however even $T = 10'$ proves enough to improve the predictive performance under our method.

[3]All methods are implemented and will be made available within the MEKA framework (http://meka.sourceforge.net).

Table 2. Average exact match over 5-fold CV.

Dataset	BR	CC	PCC	CDN	MCC	M2CC
Music	0.299	0.287	0.346	0.297	0.346	0.361
Scene	0.538	0.545	0.636	0.531	0.636	0.657
Yeast	0.140	0.151	DNF	0.069	0.209	0.206
Genbase	0.941	0.964	DNF	0.945	0.964	0.967
Medical	0.585	0.622	DNF	0.602	0.629	0.627
Enron	0.065	0.099	DNF	0.073	0.101	0.103
Reuters	0.287	0.346	DNF	0.271	0.366	0.364
avg. rank	4.57	3.43		4.71	1.57	1.43

Table 3. Average running time (seconds) over 5-fold CV.

Dataset	BR	CC	PCC	CDN	MCC	M2CC
Music	0	0	0	5	1	4
Scene	12	10	15	92	25	170
Yeast	10	10	DNF	88	32	222
Genbase	10	7	DNF	572	201	382
Medical	9	10	DNF	1546	338	506
Enron	102	91	DNF	3091	706	1399
Reuters	106	119	DNF	14734	1831	20593

Table 4. Average exact match over 5-fold CV.

Dataset	ECC	EM2CC
Music	0.314 (2)	0.329 (1)
Scene	0.608 (2)	0.633 (1)
Yeast	0.186 (2)	0.193 (1)
Genbase	0.945 (1)	0.945 (1)
Medical	0.643 (2)	0.649 (1)
Enron	0.112 (2)	0.116 (1)
Reuters	0.364 (1)	0.360 (2)
avg. rank	1.71	1.14

only provide results for the first two data sets, with DNF (Did Not Finish) in Table 2 indicating this fact. Results for running time performance are also given in Table 3. Furthermore, the original CC paper [4] also presented CC in Bagging ensembles (ECC) to improve predictive performance. We also bag M2CC to create the ensemble method EM2CC. We use 10 models for each ensemble, each one starting with a different random initiation of the chain sequence (s_0). Results for predictive performance of EM2CC vs. M2CC are given in Table 4.

As claimed in the literature, CC improves over BR in all cases. PCC in turn improves on CC in the two cases where it is tractable. The MCC methods perform the best overall. Both of them outperform CC on every occasion – with the exception of ties on Genbase. We note that MCC provides identical results to PCC on both datasets that it finishes on. M2CC obtains even higher performance than PCC on these datasets, underlining the importance of the chain sequence in constructing classifier chains, and the fact that we have been able to leverage this to create a better model. As expected, M2CC also out-

performs MCC in most cases, and overall, precisely because it optimises the chain-sequence space, improving the sequence of labels at training time.

Clearly MCC and M2CC take much longer than the standard greedy CC method, but they are still tractable on all the data sets we looked at (unlike PCC) and the improvement in predictive performance is well worth the trade off. Furthermore, we note that our methods are generally faster than the conditional dependency network CDN (with the exception of M2CC on some datasets).

Finally, we note that, although ECC is able to offer an improvement over CC (particularly on Yeast, Medical and Enron), EM2CC still maintains a clear advantage over ECC on all data sets. We also notice that, while a Bagging ensemble can raise the accuracy of CC, even this additional accuracy does not always compete well with a *single* MCC or M2CC model (if we compare between Tables 2 and 4).

5. CONCLUSIONS AND FUTURE WORK

We have introduced two novel efficient Monte Carlo (MC) algorithms (MCC and M2CC) for multi-label learning using classifier chains. The proposed approaches use MC techniques to efficiently search the label-path space at inference time and also the chain-sequence space at training time in the case of M2CC. We show through an empirical evaluation that using these methods results in better predictive performance than related methods while remaining computationally tractable. In future work, we intend to look at more advanced random search algorithms and dependency structures other than chain models, as well different payoff functions. We also plan to extend this work to multi-valued target attributes and hierarchical MLC problems.

6. RELATION TO PRIOR WORK

This work builds on the classifier chains (CC) framework for multi-label classification (MLC) [4] and its recent probabilistic extension, probabilistic classifier chains (PCC) [5]. More specifically, since the Bayes-optimal approach proposed by PCC is unfeasible in practice due to its computational cost, we propose a tractable inference scheme, based on Monte Carlo (MC) methods, which attains a similar performance to PCC. Furthermore, we also introduce an MC approach for the optimization of the chain of classifiers during the training stage, an issue that has not been tackled before as far as we know, except by avoiding it altogether (e.g., by using conditional dependency networks [6]). Finally, ensemble versions of the two MC approaches proposed have been developed following the line of ECC and EPCC [4, 5].

7. REFERENCES

[1] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. 2010, 2nd edition, Springer.

[2] Jesse Read, *Scalable Multi-label Classification*, Ph.D. thesis, University of Waikato, 2010.

[3] Grigorios Tsoumakas and Ioannis P. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *ECML '07: 18th European Conference on Machine Learning*. 2007, pp. 406–417, Springer.

[4] Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank, "Classifier chains for multi-label classification," *Machine Learning*, 2011.

[5] Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *ICML '10: 27th International Conference on Machine Learning*, Haifa, Israel, June 2010, Omnipress.

[6] Yuhong Guo and Suicheng Gu, "Multi-label classification using conditional dependency networks.," in *IJCAI '11: Proceedings of the 24th International Conference on Artificial Intelligence*. 2011, pp. 1300–1305, IJCAI/AAAI.

[7] Grigorios Tsoumakas and Ioannis Katakis, "Multi label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[8] Jesse Read, Bernhard Pfahringer, and Geoff Holmes, "Multi-label classification using ensembles of pruned sets," in *ICDM'08: Eighth IEEE International Conference on Data Mining*. 2008, pp. 995–1000, IEEE.

[9] Julio H. Zaragoza, Luis Enrique Sucar, Eduardo F. Morales, Concha Bielza, and Pedro Larrañaga, "Bayesian chain classifiers for multidimensional classification," in *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI '11)*, 2011.

[10] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Deroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, Sept. 2012.

[11] Trevor Hastie and Robert Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems*, Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, Eds. 1998, vol. 10, MIT Press.

[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Reutemann Peter, and Ian H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[13] Jesse Read, Luca Martino, and David Luengo, "Efficient Monte Carlo optimization for multi-label classifier chains," Tech. Rep., Universidad Carlos III of Madrid, Dic. 2012, arXiv:1211.2190.