Work on Multi-label Classification

Jesse Read

Acknowledgements: (University of Waikato, NZ) Bernhard Pfahringer, Geoff Holmes, Albert Bifet, Eibe Frank; (UC3M, Madrid) Fernando Perez Cruz, Joaquín Miguez

Department of Signal Theory and Communications
Universidad Carlos III
Madrid, Spain

May 10, 2011

Multi-label Classification: An Introduction

Data instance:

- Binary Classification: e.g. is this a beach? $\in \{\texttt{No}, \texttt{Yes}\}$

Multi-label Classification: An Introduction

Data instance:

- Binary Classification: e.g. is this a beach? $\in \{\texttt{No}, \texttt{Yes}\}$
- Multi-class Classification: e.g. what is this? $\in \{\texttt{Beach}, \texttt{Forest}, \texttt{City}, \texttt{People}\}$

Multi-label Classification: An Introduction

Data instance:

- Binary Classification: e.g. is this a beach? $\in \{\texttt{No}, \texttt{Yes}\}$
- Multi-class Classification: e.g. what is this? $\in \{\texttt{Beach}, \texttt{Forest}, \texttt{City}, \texttt{People}\}$
- Multi-label Classification: e.g. which of these? $\subseteq \{\texttt{Beach}, \texttt{Forest}, \texttt{City}, \texttt{People}\}$

Multi-label classification is the supervised classification task where each data instance may be associated with *multiple* class labels.

Multi-label Classification: An Introduction

- Input space: $\mathcal{X} = \mathbb{R}^d$
- Instance $\mathbf{x} = [x_1, \ldots, x_d] \in \mathcal{X}$
- Class labels: $\mathcal{L} = \{1, 2, \ldots, L\}$
- Label space: $\mathcal{Y} = \{0, 1\}^L$
- Labelset: $\mathbf{y} = [y_1, \ldots, y_L] \in \mathcal{Y}$; $y_j = 1$ if jth label relevant to $\mathbf{x}$; else 0)
- Training set: $\{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, \ldots, N\} \subset (\mathcal{X} \times \mathcal{Y})$
- Classification: $h : \mathcal{X} \rightarrow \mathcal{Y}$
- Prediction: $\hat{\mathbf{y}} = h(\mathbf{x})$
- Evaluation:
    - compare each $\hat{\mathbf{y}}_i$ with each $\mathbf{y}_i$ (*labelset accuracy*); OR
    - compare each $\hat{y}_j$ with each y_j (*label accuracy*).

Example Applications

Multi-label classification is relevant to many domains:

- Text
  - text documents → subject categories
  - e-mails → labels
  - medical description of symptoms → diagnoses
- Vision
  - images/video → scene concepts
  - images/video → objects identified/recognised
- Audio
  - music → genres / moods
  - sound signals → events / concepts
- Bioinformatics
  - genes → biological functions
- Sensor Networks / Robotics
  - sensor inputs → states / error diagnoses

Methods for Multi-label Classification

Problem Transformation: Using off-the-shelf binary / multi-class classifiers for multi-label learning.

Methods for Multi-label Classification

Problem Transformation: Using off-the-shelf binary / multi-class classifiers for multi-label learning.

- Binary Relevance method (BR) One binary classifier for each label:
  - $\hat{\mathbf{y}} \equiv [\hat{y}_1, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$
    - where each $y_j \in \{0, 1\}$
  - simple; flexible; fast
  - *but* does not explicitly model label dependencies

Methods for Multi-label Classification

Problem Transformation: Using off-the-shelf binary / multi-class classifiers for multi-label learning.

- Binary Relevance method (BR) One binary classifier for each label:
  - $\hat{\mathbf{y}} \equiv [\hat{y}_1, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$
    - where each $y_j \in \{0, 1\}$
  - simple; flexible; fast
  - *but* does not explicitly model label dependencies
- Label Powerset method (LP): One multi-class classifier; one class for each *labelset*:
  - $\hat{\mathbf{y}} \equiv \hat{c} = h(\mathbf{x})$
    - where class $\hat{c} \in \mathcal{C}$, $\mathcal{C} = 2^{\mathcal{L}}$ (in practice $\mathcal{C} = \mathrm{distinct}(\{\mathbf{y}_1, \ldots, \mathbf{y}_N\})$)
  - models label dependencies; good accuracy
  - *but* high complexity ($\min(N, 2^L)$); overfitting; difficult for incremental classification

Ensembles Of Classifier Chains

Efficiently modelling label dependencies in a BR-like way using Classifier Chains (CC) [Read et al., 2009].

- BR: $\hat{\mathbf{y}} \equiv [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_L(\mathbf{x})]$
- CC: $\hat{\mathbf{y}} \equiv [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), h_2(\mathbf{x}, \hat{y}_1), \ldots, h_L(\mathbf{x}, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{L-1})]$

Ensembles Of Classifier Chains

Efficiently modelling label dependencies in a BR-like way using Classifier Chains (CC) [Read et al., 2009].

- BR: $\hat{\mathbf{y}} \equiv [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_L(\mathbf{x})]$
- CC: $\hat{\mathbf{y}} \equiv [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), h_2(\mathbf{x}, \hat{y}_1), \ldots, h_L(\mathbf{x}, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{L-1})]$

Predictive performance depends on how labels are indexed in $\mathbf{Y}$. We used an *Ensemble of (random) Classifier Chains* (ECC):

- competitive predictive performance
- scales to large datasets ($100,000$s of examples, 100s of labels)

Ensembles Of Classifier Chains

Efficiently modelling label dependencies in a BR-like way using Classifier Chains (CC) [Read et al., 2009].

- BR: $\hat{\mathbf{y}} \equiv [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_L(\mathbf{x})]$
- CC: $\hat{\mathbf{y}} \equiv [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L] = [h_1(\mathbf{x}), h_2(\mathbf{x}, \hat{y}_1), \ldots, h_L(\mathbf{x}, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{L-1})]$

Predictive performance depends on how labels are indexed in $\mathbf{Y}$. We used an *Ensemble of (random) Classifier Chains* (ECC):

- competitive predictive performance
- scales to large datasets ($100,000$s of examples, 100s of labels)

A probabilistic interpretation [Cheng et al., 2010]:

$$P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}}(y_1) \prod_{j=1}^{L} P_{\mathbf{x}}(y_j | y_1, \ldots, y_{j-1})$$

- Improves on CC, but not ECC; complex (searches all 2^L combinations)

Labelset Mapping

An issue with classifier chains:

- high memory use (large L = large feature space).

New method: k-Labelset Mapping (kLM):

- k-Nearest Neighbors in the label space of BR:
  - $\hat{\mathbf{w}} = \mathbf{h}(\mathbf{x})$, $\mathbf{w} \in \mathbb{R}^L$
  - $d_i = \texttt{euclidean\_dist}(\mathbf{y}_i, \hat{\mathbf{w}})$ for $\mathbf{y}_i \in \texttt{distinct}(\{\mathbf{y}_1, \ldots, \mathbf{y}_N\})$
  - map $\hat{\mathbf{w}}' = \mathrm{avg}(\mathbf{y}_1, \ldots, \mathbf{y}_k)$ from smallest $d_1, \ldots, d_k$
  - map $\hat{\mathbf{y}} = f_t(\hat{\mathbf{w}}')$
- e.g. $[0.9, 0.0, 0.5, 0.8] \mapsto \{[1, 0, 0, 1]_{0.8}, [1, 0, 0, 0]_{1.4}, [0, 0, 0, 1]_{1.6}\} \mapsto [0.6, 0.0, 0.0, 0.6] \mapsto [1, 0, 0, 1]$
- Models label dependency like LP, fast like BR
- Faster than, and competitive with CC

But, same story: different classifiers perform better on different datasets:

- BR: fast, best when labels are independent
- LP-based, (e.g., kLM): good when only a few *distinct* labelsets (indicating *label interdependence*)
- CC: performs well overall by modelling label interdependence approximately (but doesn't specialise)

Label dependencies are the biggest influencing factor on performance, directly and indirectly.

Label Dependence — A Closer Look

It was always clear that:

- there are *dependencies* (i.e., correlations) between labels; and
- modelling these dependencies *improves predictive performance*; but
- is inherently *expensive* ($\frac{L(L-1)}{2}$ *pairwise*, 2^L *all*).

Label Dependence — A Closer Look

It was always clear that:

- there are *dependencies* (i.e., correlations) between labels; and
- modelling these dependencies *improves predictive performance*; but
- is inherently *expensive* ($\frac{L(L-1)}{2}$ *pairwise*, 2^L *all*).

If we can discover *significant* dependencies, we can model *only these*, and model them appropriately.

- smaller, better chains for CC
- only map dependent labels with kLM
- etc.

Label Dependence — A Closer Look

It was always clear that:

- there are *dependencies* (i.e., correlations) between labels; and
- modelling these dependencies *improves predictive performance*; but
- is inherently *expensive* ($\frac{L(L-1)}{2}$ *pairwise*, 2^L *all*).

If we can discover *significant* dependencies, we can model *only these*, and model them appropriately.

- smaller, better chains for CC
- only map dependent labels with kLM
- etc.

In multi-label data [Read, 2010, Dembczyński et al., 2010]:

- Unconditional dependence: *correlations* in $\mathbf{Y}$.
- Conditional dependence: dependencies in $\mathbf{Y}$ given a specific $\mathbf{x} \in \mathcal{X}$

Unconditional (In)dependence

- A vector of labels $\mathbf{Y}$ is is **unconditionally** ***L*-independent** if:

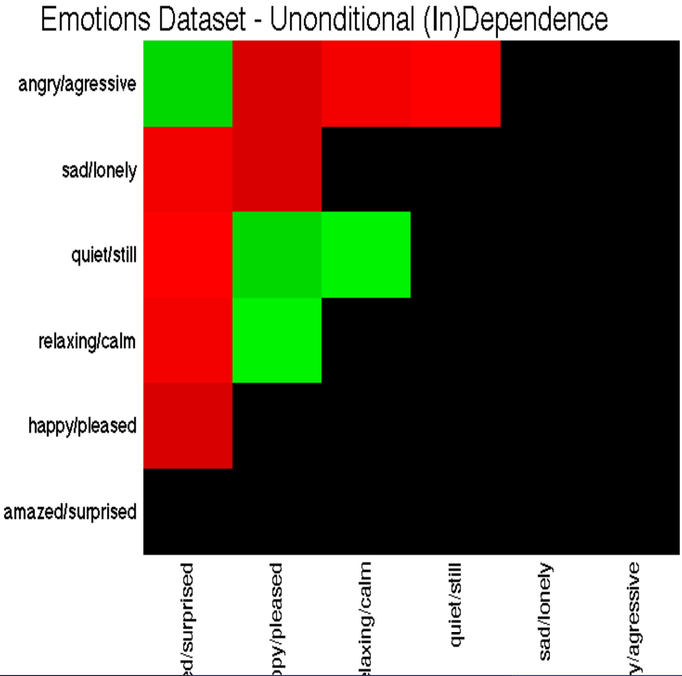
$$p(\mathbf{Y}) = \prod_{j=1}^{L} p(Y_j)$$

- Can measure with *Mutual information*,

$$I(Y_j; Y_k) = \sum_{y_j \in \{0,1\}} \sum_{y_k \in \{0,1\}} p(y_j, y_k) \log\left(\frac{p(y_j, y_k)}{p_1(y_j) p_2(y_k)}\right)$$

- or *Pearson's Correlation Coefficient*

$$P_{Y_j, Y_k} = \frac{\mathrm{cov}(Y_j, Y_k)}{\sigma_{Y_j} \sigma_{Y_k}}$$


Emotions Dataset - Unonditional (In)Dependence
angry/agressive
sad/lonely
quiet/still
relaxing/calm
happy/pleased
amazed/surprised
ed/surprised
ppy/pleased
elaxing/calm
quiet/still
sad/lonely
y/agressive

Conditional (In)dependence

Label dependence taking into account a specific instance: $P(y_j|y_k, \mathbf{x})$. Can use, for example:

- *Four-class pairWise classification* method (FW):
  - Models $y_{jk} \in \{00, 01, 10, 11\}$ for *each pair* of labels $y_{1 \leq j < k \leq L}$
  - prediction $\hat{\mathbf{y}} =$ average label votes for each y_j (and use a threshold).

and compare to performance vs. BR to measure significant dependence.

Conditional (In)dependence

Label dependence taking into account a specific instance: $P(y_j|y_k, \mathbf{x})$. Can use, for example:

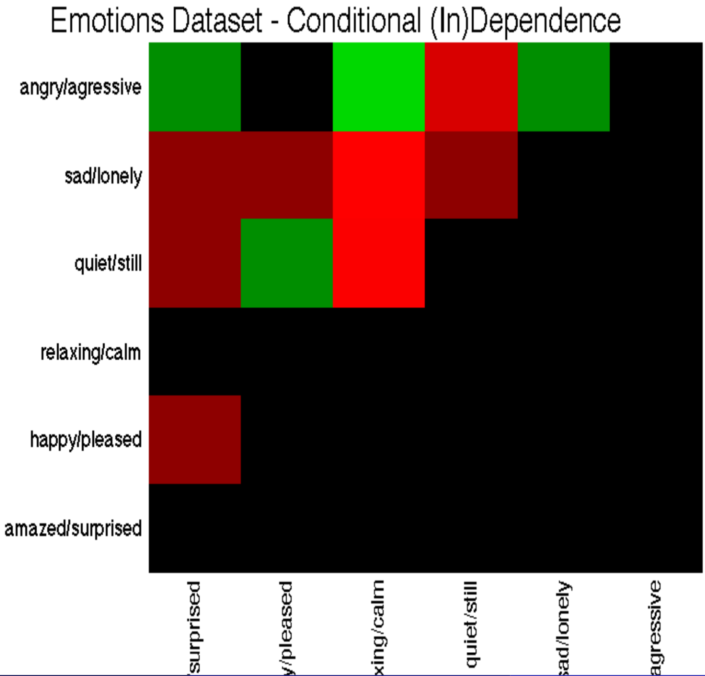
- *Four-class pairWise classification* method (FW):
  - Models $y_{jk} \in \{00, 01, 10, 11\}$ for *each pair* of labels $y_{1 \leq j < k \leq L}$
  - prediction $\hat{\mathbf{y}}$ = average label votes for each y_j (and use a threshold).

and compare to performance vs. BR to measure significant dependence.

Table: On synthetic data with strong conditional dependence and independence.

	Conditional Dependence				Conditional Independence			
	FW	BR	LP	CC	FW	BR	LP	CC
Subset Acc.	**0.77**	0.70	0.69	0.74	0.84	**0.89**	0.59	0.88
Labelset Acc.	**0.45**	0.38	0.38	0.43	0.59	**0.61**	0.43	0.60
Label Acc.	**0.94**	0.92	0.91	**0.94**	0.97	**0.98**	0.90	**0.98**

- FW (also CC) best at modelling conditional dependence.
- If complete independence, BR is best choice.


Emotions Dataset - Conditional (In)Dependence
angry/agressive
sad/lonely
quiet/still
relaxing/calm
happy/pleased
amazed/surprised
surprised
y/pleased
xing/calm
quiet/still
sad/lonely
agressive

Conditional[1] and unconditional dependence and independence.

label pair	conditional.	uncond.
{amazed-suprised, happy-pleased}	-0.565 ± 0.33	0.0
{amazed-suprised, relaxing-clam}	3.806 ± 1.028	0.476
{amazed-suprised, quiet-still}	-0.422 ± 0.020	0.59
{amazed-suprised, sad-lonely}	0.566 ± 0.742	0.362
{amazed-suprised, angry-aggresive}	-2.258 ± 2.071	0.126
{happy-pleased, relaxing-clam}	2.26 ± 5.283	0.028
{happy-pleased, quiet-still}	2.534 ± 0.078	0.225
{happy-pleased, sad-lonely}	3.512 ± 10.79	0.427
{happy-pleased, angry-aggresive}	1.685 ± 2.939	0.369
{relaxing-clam, quiet-still}	0.986 ± 0.021	0.455
{relaxing-clam, sad-lonely}	-0.281 ± 0.082	0.19
{relaxing-clam, angry-aggresive}	1.554 ± 3.485	0.8
{quiet-still, sad-lonely}	0.425 ± 0.515	0.547
{quiet-still, angry-aggresive}	1.276 ± 9.072	0.395
{sad-lonely, angry-aggresive}	1.13 ± 1.321	0.215

[1] 15×2 CV eval(FW,$\mathcal{D}$)−eval(BR,$\mathcal{D}$) on training set $\mathcal{D}$

Preliminary Results; Current & Future Work

For each pair Y_j, Y_k, if $I(Y_j; Y_k) > 0.4$, train FW ($y_{jk} \in \{00, 01, 10, 11\}$) for this pair, else train (BR $y_{jk} \in \{0, 1\}^2$). Sum votes for each label (and use a threshold) to get a labelset prediction $\hat{\mathbf{y}}$:

Table: Comparing performance on the *Emotions* dataset

	BR	FW	BR/FW
Subset Accuracy	0.46	0.54	**0.57**
Labelset Accuracy	0.21	0.27	**0.29**
Label Accuracy	0.78	**0.79**	**0.79**

Preliminary Results; Current & Future Work

For each pair Y_j, Y_k, if $I(Y_j; Y_k) > 0.4$, train FW ($y_{jk} \in \{00, 01, 10, 11\}$) for this pair, else train (BR $y_{jk} \in \{0, 1\}^2$). Sum votes for each label (and use a threshold) to get a labelset prediction $\hat{\mathbf{y}}$:

Table: Comparing performance on the *Emotions* dataset

	BR	FW	BR/FW
Subset Accuracy	0.46	0.54	**0.57**
Labelset Accuracy	0.21	0.27	**0.29**
Label Accuracy	0.78	**0.79**	**0.79**

Current/Future Work:

- Combining BR with FW, LP, CC, etc.
- Improving kLM, classifier chains (CC), etc.
- Learning Conditional Random Fields (CRFs).
- Feature selection/reduction.

Sidetrack: Feature Selection

So far, many of the best/popular methods are based on:

- binary relevance; or
- LP-*subsets*.

For example predicting the relevance of label y_1 under BR:

$$\hat{y}_1 = h_1(\mathbf{x})$$

Are *all* features in $\mathbf{X}$ relevant to predicting only Y_1?

- Answer so far: often not!
  - For BR, accuracy peaks using the top 30-50% of features.
- Big savings to be had if we can figure out which feature subset to use (efficiently).

Multi-label Classification in Datastreams

Initial work ([Read et al., 2010]; from [Read, 2010] & [Bifet et al., 2009]) on *instance*-incremental multi-label classification in datastreams (data arriving continuously and rapidly; concept drift expected):

- Adaptive Ensembles of Classifier Chains (ECC)
  - Hoeffding trees as base-classifiers
  - reset classifiers based on current performance / concept drift
- Multi-label Hoeffding Tree:
  - Label Powerset method (LP) at the leaves
  - an ensemble strategy to deal with concept drift
- Experiments/Results:
  - generating synthetic multi-label data streams
  - setting a benchmark on real-world and synthetic data

Current / Future Work:

- beating the benchmark
- modelling label dependencies incrementally, and drift within
- incremental feature selection

Summary / Other areas of interest

Summary:

- multi-label classification
- multi-label methods
- dependencies in multi-labeled data
- more accurate / efficient methods using these dependencies
- special considerations for data streams

Summary / Other areas of interest

Summary:

- multi-label classification
- multi-label methods
- dependencies in multi-labeled data
- more accurate / efficient methods using these dependencies
- special considerations for data streams

Other areas of interest:

- Distributed Algorithms for Tracking in Sensor Networks
  - testbed implementation
  - particle filters
  - machine learning
  - multi-label classification

The End

References:

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., and Gavaldà, R. (2009).
New ensemble methods for evolving data streams.
In *KDD '09: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*. ACM.

Cheng, W., Dembczyński, K., and Hüllermeier, E. (2010).
Bayes optimal multilabel classification via probabilistic classifier chains.
In *ICML '10: 27th International Conference on Machine Learning*, Haifa, Israel. Omnipress.

Dembczyński, K., Waegeman, W., Cheng, W., and Hüllermeier, E. (2010).
On label dependence in multi-label classification.
In *Workshop Proceedings of Learning from Multi-Label Data*, pages 5–12, Haifa, Israel.

Read, J. (2010).
Scalable Multilabel Classification.
PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

Read, J., Bifet, A., Holmes, G., and Pfahringer, B. (2010).
Efficient multi-label classification for evolving data streams.
Technical report, University of Waikato, Hamilton, New Zealand.
Working Paper 2010/04.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009).
Classifier chains for multi-label classification.
In *ECML '09: 20th European Conference on Machine Learning*, pages 254–269. Springer.

Spyromitros Xioufis, E. (2011).
Dealing with concept drift and class imbalance in multi-label stream classification.
Master's thesis, Department of Computer Science, Aristotle University of Thessaloniki.