Streaming Multi-label Classification

Jesse Read†, Albert Bifet, Geoff Holmes, Bernhard Pfahringer

University of Waikato, Hamilton, New Zealand

† currently at: Universidad Carlos III, Madrid

October 19, 2011
Multi-label Classification

Each data instance is associated with a subset of class labels (as opposed to a single class label).

- dependencies between labels
- greater dimensionality ($2^L$ instead of $L$)
- evaluation: different measures

Music labeled with emotions dataset; co-occurrences
Introduction: Streaming Multi-label Classification

Data Stream Classification

Data instances arrive **continually** (often automatic / collaborative process) and potentially **infinitely**.

- cannot store everything
- ready to predict at any point
- concept drift
- evaluation: different methods, getting labelled data
Applications of Multi-label Learning

- **Text**
  - text documents → subject categories
  - e-mails → labels
  - medical description of symptoms → diagnoses

- **Vision**
  - images/video → scene concepts
  - images/video → objects identified; objects recognised

- **Audio**
  - music → genres; moods
  - sound signals → events; concepts

- **Bioinformatics**
  - genes → biological functions

- **Robotics**
  - sensor inputs → states; object recognition; error diagnoses

Many of these applications exist in a streaming context!
Applications of Multi-label Learning

- Text
  - text documents $\rightarrow$ subject categories
  - e-mails $\rightarrow$ labels
  - medical description of symptoms $\rightarrow$ diagnoses

- Vision
  - images/video $\rightarrow$ scene concepts
  - images/video $\rightarrow$ objects identified; objects recognised

- Audio
  - music $\rightarrow$ genres; moods
  - sound signals $\rightarrow$ events; concepts

- Bioinformatics
  - genes $\rightarrow$ biological functions

- Robotics
  - sensor inputs $\rightarrow$ states; object recognition; error diagnoses

Many of these applications exist in a streaming context!
Methods for Multi-label Classification

Problem Transformation

- **Transform** a multi-label problem into single-label (multi-class) problems
- Use any off-the-shelf single-label classifier to suit requirements: Decision Trees, SVMs, Naive Bayes, \( kNN \), *etc.*
Methods for Multi-label Classification

**Problem Transformation**
- **Transform** a multi-label problem into single-label (multi-class) problems
- Use any off-the-shelf single-label classifier to suit requirements: Decision Trees, SVMs, Naive Bayes, kNN, etc.

**Algorithm Adaptation**
- **Adapt** a single-label method directly for multi-label classification
- Often for a specific domain; incorporating the advantages/disadvantages of chosen method
If we have $L$ labels . . .

**Binary Relevance (BR)**

$L$ separate binary-class problems: e.g.

$$(x, \{l_1, l_3\}) \rightarrow (x, 1)_1, (x, 0)_2, (x, 1)_3, \ldots, (x, 0)_L$$

- simple, flexible, fast
- no explicit modelling of label dependencies; poor accuracy

**Classifier Chains (CC)** [Read et al., 2009]: model label dependencies along a BR ‘chain’; in ensemble (ECC).

- high predictive performance, approximately as fast as BR

Run BR twice (2BR): once on the input data, and again on the initially predicted output labels [Qu et al., 2009]

- learn label dependencies
Problem Transformation Methods

If we have \( L \) labels . . .

**Label Powerset (LP)**

All of the \( 2^L \) possible labelset combinations\(^a\) are treated as single labels in a multi-class problem: e.g. \((x, \{l_1, l_5\}) \rightarrow (x, y)\) where \( y = \{l_1, l_5\} \)

- explicit modelling of label dependencies; high accuracy
- overfitting and sparsity; *can be* very slow if many unique labelsets

\(^a\)in practice, only the combinations found in the training data

**Pruned sets (PS) [Read et al., 2008]:** Prune and subsample *infrequent* labelsets before running LP; in ensemble (EPS).

- *much* faster, reduces label sparsity and overfitting over LP

Using random \( k \)-label subsets (RAkEL) for LP instead of the full label set [Tsoumakas and Vlahavas, 2007]

- \( m2^k \) worst-case complexity instead of \( 2^L \)
Algorithm Adaptation

Multi-label C4.5 decision trees

Adapted C4.5 decision trees to multi-label classification by modifying the entropy calculation to allow multi-label predictions at the leaves [Clare and King, 2001]

- Fast, works very well,
- Most success in specific domains (e.g. biological data).
How can we use multi-label methods on data streams?

- **Binary Relevance** methods: just use an incremental binary classifier e.g. Naive Bayes, Hoeffding Trees, chunked-SVMs (‘batch-incremental’)

- **Label Powerset** methods: the known labelsets change over time! use Pruned Sets for fewer labelsets

  - assume we can learn the distribution of labelsets from the first $n$ examples

  - when the distribution changes, so has the concept!

Multi-label C4.5: can create multi-label Hoeffding trees!
Multi-label Learning in Data Streams

How can we use multi-label methods on data streams?

- **Binary Relevance** methods: just use an incremental binary classifier e.g. Naive Bayes, Hoeffding Trees, chunked-SVMs (‘batch-incremental’)
- **Label Powerset** methods: the known labelsets change over time!
  - use **Pruned Sets** for fewer labelsets
  - assume we can learn the distribution of labelsets from the first \( n \) examples
  - when the distribution changes, so has the concept!
How can we use multi-label methods on data streams?

- **Binary Relevance** methods: just use an incremental binary classifier e.g. Naive Bayes, Hoeffding Trees, chunked-SVMs (‘batch-incremental’)
- **Label Powerset** methods: the known labelsets change over time!
  - use **Pruned Sets** for fewer labelsets
  - assume we can learn the distribution of labelsets from the first $n$ examples
  - when the distribution changes, so has the concept!
- **Multi-label C4.5**: can create multi-label Hoeffding trees!
Using a drift-detector

- Use an ensemble (Bagging), and
- employ a drift-detection method of your choice; we use ADWIN [Bifet and Gavalda, 2007]
  - an ADaptive sliding WINdow with rigorous guarantees
- when drift is detected, the worst model is reset.
Dealing with Concept Drift

Using a drift-detector

- Use an ensemble (Bagging), and
- employ a drift-detection method of your choice; we use ADWIN [Bifet and Gavaldà, 2007]
  - an ADaptive sliding WINdow with rigorous guarantees
- when drift is detected, the worst model is reset.

Alternative method – batch-incremental (e.g. [Qu et al., 2009]):

- Assume there is always drift, and
- reset a classifier every $n$ instances.
• **Waikato Environment for Knowledge Analysis**
  • Collection of state-of-the-art machine learning algorithms and data processing tools implemented in Java
    • Released under the GPL
  • Support for the whole process of experimental data mining
    • Preparation of input data
    • Statistical evaluation of learning schemes
    • Visualization of input data and the result of learning

• Used for education, research and applications
• Complements Data Mining by Witten & Frank & Hall

---

Massive Online Analysis is a framework for online learning from data streams.

- Closely related to WEKA
- A collection of instance-incremental and batch-incremental methods for classification
- ADWIN for adapting to concept drift
- Tools for evaluation, and generation of evolving data streams
- MOA is easy to use and extend
  - void resetLearningImpl()
  - void trainOnInstanceImpl(Instance inst)
  - double[] getVotesForInstance(Instance i)

http://moa.cs.waikato.ac.nz
Multi-label extension to WEKA

- Very closely integrated with WEKA
  - extend MultilabelClassifier
  - void buildClassifier(Instances X)
  - double[] distributionForInstance(Instance x)
    (plus threshold function)
- Problem transformation methods using any WEKA base-classifier
- Generic ensemble and thresholding methods
- Provides a wrapper around Mulan\textsuperscript{3} classifiers
- Multi-label evaluation

\textsuperscript{3}http://mulan.sourceforge.net
\textsuperscript{4}http://meka.sourceforge.net
A Multi-label Learning Framework for Data Streams

- MOA wrapper for WEKA (+MEKA) classifiers.
- MEKA wrapper for MOA classifiers.
- Real multi-label data + multi-label synthetic data streams
- Multi-label evaluation measures with data-stream evaluation methods
Multi-label Evaluation Measures

Given labelset $\hat{Y}$ for a test example . . .

- Example Accuracy $\hat{Y} = Y$?
- Label Accuracy $(l \in \hat{Y}) = (l \in Y)$? for $l = 1, \ldots, L$
- Subset Accuracy $\frac{|\hat{Y} \cap Y|}{|\hat{Y} \cup Y|}$?

Also need to consider a threshold if a classifier outputs $\in \mathbb{R}^L$:

- $l \in Y \iff y_l > t$ for some threshold $t$

Data stream Evaluation Methods

- Holdout
- Interleaved Test-Then-Train
- Prequential
  - output evaluation statistics from a sliding window
Generating Synthetic Data

Unfortunately large sources of real-world data are:

- sensitive; difficult to parse; or
- too large.

\[
Y = f(\theta)
\]

where \(\theta\) describes label dependencies

\[
x = f(Y, g)
\]

where \(g\) is any MOA binary-class generator e.g.:

- Random RBF (Radial Basis Function) Generator
- Random Tree Generator

Concept drift is introduced by changing \(\theta\) (label space) over time, and by introducing drift in \(g\) (input space)—standard in MOA.
Generating Synthetic Data

Unfortunately large sources of real-world data are:

- sensitive; difficult to parse; or
- too large.

Our framework can synthesis evolving multi-label data streams.

Generate example \((x, Y)\) (an input \(x\) and associated labelset \(Y\))

1. \(Y = f(\theta)\) where \(\theta\) describes label dependencies
2. \(x = f(Y, g)\) where \(g\) is any MOA binary-class generator e.g.:
   - Random RBF (Radial Basis Function) Generator
   - Random Tree Generator

Concept drift is introduced by changing \(\theta\) (label space) over time, and by introducing drift in \(g\) (input space)—standard in MOA.
GUI: Configuring a multi-label classifier

MOA Graphical User Interface

Configure task

Purpose
Evaluates a classifier on a stream by testing then training with each example in sequence.

learner: HoeffdingTree

stream: E-IMDB-F.arff

evaluator:

instanceLimit: 1,000

timeLimit: 1,000

sampleFrequency: 1

maxMemory: 3

memCheckFrequency: 1

MOA Classifier: HoeffdingTree

Export as .txt file...
GUI: Setting a multi-label stream generator

Class: `moa.streams.generators.multilabel.MetaMultilabelGenerator`

Purpose:
Generates a multi-label stream using a binary generator.

- **binaryGenerator**: `RandomTreeGenerator`
- **metaRandomSeed**: 1
- **numLabels**: 8
- **skew**: 1
- **labelCardinality**: 2.5

Buttons:
- Help
- Reset to defaults
- Cancel
- OK
Adapted current methods to data streams:

- Ensembles of Binary Relevance (EBR)
- Ensembles of Classifier Chains (ECC)
- Ensembles of Pruned Sets (EPS)
  - model the first 1000 labelset combinations
- 2x Binary Relevance (2BR) [Qu et al., 2009]
- Multi-label Hoeffding Trees (HT)

Created a novel method:

- Ensembles of Multi-label Hoeffding Trees with Pruned Sets at the leaves (EHT$_{PS}$) [Read et al., 2010].
# Data sources

**Table:** Multi-label data sources.

| Source       | N      | L  | D     | $\sum_{i} |Y_i|$ |
|--------------|--------|----|-------|---------|
| TMC2007      | 28596  | 22 | 500b  | 2.2     |
| MediaMill    | 43907  | 101| 120n  | 4.4     |
| 20NG         | 19300  | 20 | 1001b | 1.1     |
| IMDB         | 120919 | 28 | 1001b | 2.0     |
| Slashdot     | 3782   | 22 | 1079b | 1.2     |
| Enron        | 1702   | 53 | 1001b | 3.4     |
| Ohsumed      | 13929  | 23 | 1002n | 1.7     |
| SynG($g=$RBF)| 1E5    | 25 | 80n   | 2.8     |
| SynT($g=$RTG)| 1E6    | 8  | 30b   | 1.6     |
| SynGa($g=$RBF)| 1E5  | 25 | 80n   | 1.5$\rightarrow$3.5 |
| SynTa($g=$RTG)| 1E6  | 8  | 30b   | 1.8$\rightarrow$3.0 |

$n$ indicates numeric attributes, and $b$ binary.
**Evaluation**

**Table:** Number of wins over 11 datasets; 3 evaluation measures

<table>
<thead>
<tr>
<th></th>
<th>ex-acc</th>
<th>lbl-acc</th>
<th>set-acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>EHT&lt;sub&gt;PS&lt;/sub&gt;</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>EBR</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>HT</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>EPS</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2BR</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table:** Average running time (seconds) over 11 datasets

<table>
<thead>
<tr>
<th></th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>EHT&lt;sub&gt;PS&lt;/sub&gt;</td>
<td>1824</td>
</tr>
<tr>
<td>EBR</td>
<td>1580</td>
</tr>
<tr>
<td>HT</td>
<td>59</td>
</tr>
<tr>
<td>EPS</td>
<td>2209</td>
</tr>
<tr>
<td>2BR</td>
<td>4388</td>
</tr>
</tbody>
</table>

- Problem Transformation methods (EBR, EPS) using HoeffdingTree classifiers, 2BR using J48 (WEKA’s C4.5).
- All use ADWIN to detect concept drift (except 2BR—every 1000 examples).
Summary and Future Work

A multi-label streaming framework:
- Streaming problem-transformation and algorithm-adaptation methods
- Multi-label and data-stream-specific evaluation
- Synthetic multilabel-data generation
- A novel method; setting a benchmark.

Future Work:
- label space and attribute space is dynamic
- more drift-detection and thresholding methods
Learning from time-changing data with adaptive windowing.
In *SDM ’07: 2007 SIAM International Conference on Data Mining*.

Knowledge discovery in multi-label phenotype data.
*Lecture Notes in Computer Science*, 2168.

Mining multi-label concept-drifting data streams using dynamic classifier ensemble.
In *ACML ’09: 1st Asian Conference on Machine Learning*.

Efficient multi-label classification for evolving data streams.

Multi-label classification using ensembles of pruned sets.
In *ICDM’08: Eighth IEEE International Conference on Data Mining*, pages 995–1000. IEEE.

Classifier chains for multi-label classification.

Random k-labelsets: An ensemble method for multilabel classification.

http://www.tsc.uc3m.es/~jesse/