Introduction to Data Mining

Multi-Label Classification

Jun Huang

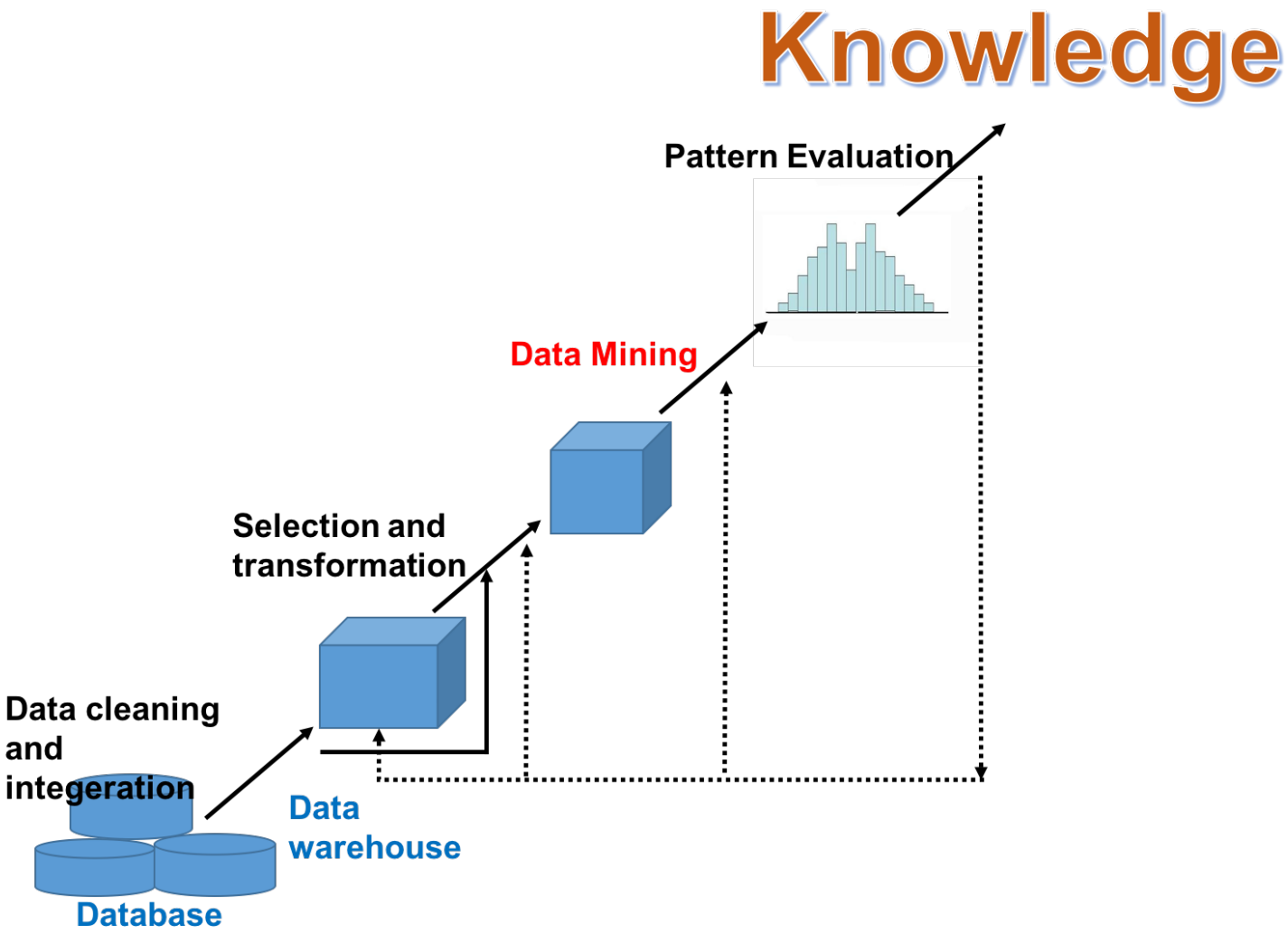
Anhui University of Technology

Spring 2018

huangjun_cs@163.com

KDD Process

Data Mining-Core of Knowledge discovery process



Multi-Label Classification

- **Single-label classification**: Is this a picture of beach?

$$\in \{\text{yes,no}\}$$

- **Multi-label classification**: Which labels are relevant to this picture?

$$\subseteq \{\text{beach, sunset, foliage, field, mountain, urban}\}$$

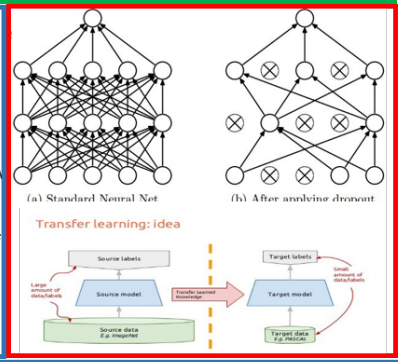
- i.e., each instance can have multiple labels instead of a single one

Multi-Label Examples

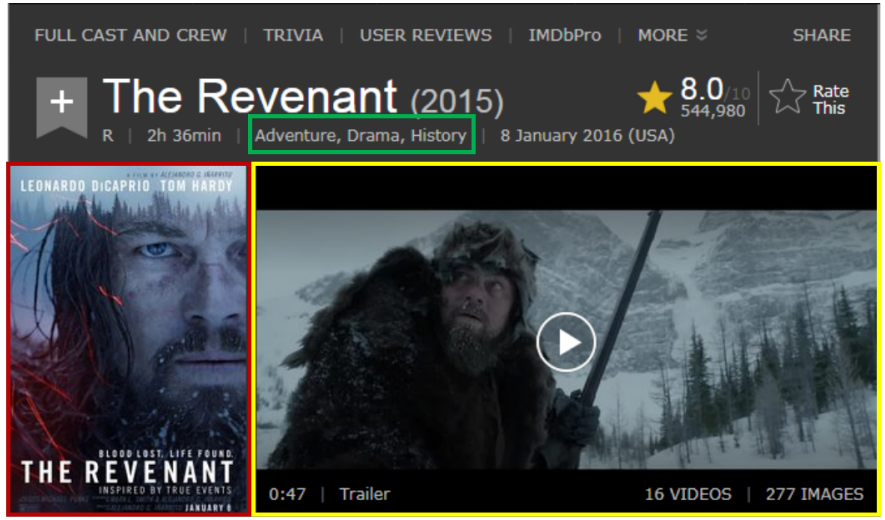
The 10 Deep Learning Methods AI Practitioners Need to Apply

Tags: Backpropagation, Convolutional Neural Networks, Deep Learning, Gradient Descent, LSTM, Neural Networks, Transfer Learning

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem.



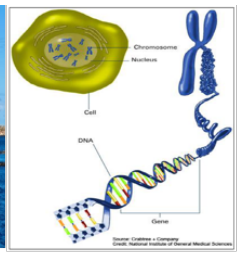
- Andrew Beam's **"Deep Learning 101"**
- Andrey Kurenkov's **"A Brief History of Neural Nets and Deep Learning"**
- Adit Deshpande's **"A Beginner's Guide to Understanding Convolutional Neural Networks"**
- Chris Olah's **"Understanding LSTM Networks"**
- Algobean's **"Artificial Neural Networks"**
- Andrej Karpathy's **"The Unreasonable Effectiveness of Recurrent Neural Networks"**

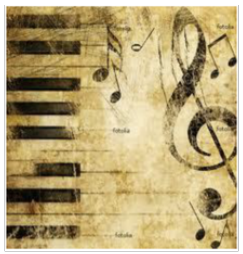


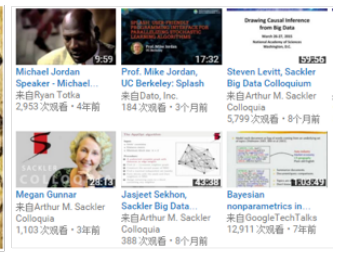
Director: Alejandro G. Iñárritu
Writers: Mark L. Smith (screenplay), Alejandro G. Iñárritu (screenplay) | 1 more credit »
Stars: Leonardo DiCaprio, Tom Hardy, Will Poulter | See full cast & crew »

Storyline

While exploring uncharted wilderness in 1823, legendary frontiersman Hugh Glass sustains injuries from a brutal bear attack. When his hunting team leaves him for dead, Glass must utilize his survival skills to find a way back home while avoiding natives on their own hunt. Grief-stricken and fueled by vengeance, Glass treks through the wintry terrain to track down John Fitzgerald, the former confidant who betrayed and abandoned him. *Written by Jwelch5742*







Introduction: Single-label vs. Multi-label

Table: Single-label $Y \in \{0, 1\}$

X1	X2	X3	X4	X5	Y
1	0.1	3	1	0	0
0	0.9	1	0	1	1
0	0.0	1	1	0	0
1	0.8	2	0	1	1
1	0.0	2	0	1	0
0	0.0	3	1	1	?

Table: Multi-label $Y \subset \{\lambda_1, ..., \lambda_L\}$

X1	X2	X3	X4	X5	Y
1	0.1	3	1	0	$\{\lambda_2, \lambda_3\}$
0	0.9	1	0	1	$\{\lambda_1\}$
0	0.0	1	1	0	$\{\lambda_2\}$
1	0.8	2	0	1	$\{\lambda_1, \lambda_4\}$
1	0.0	2	0	1	$\{\lambda_4\}$
0	0.0	3	1	1	?

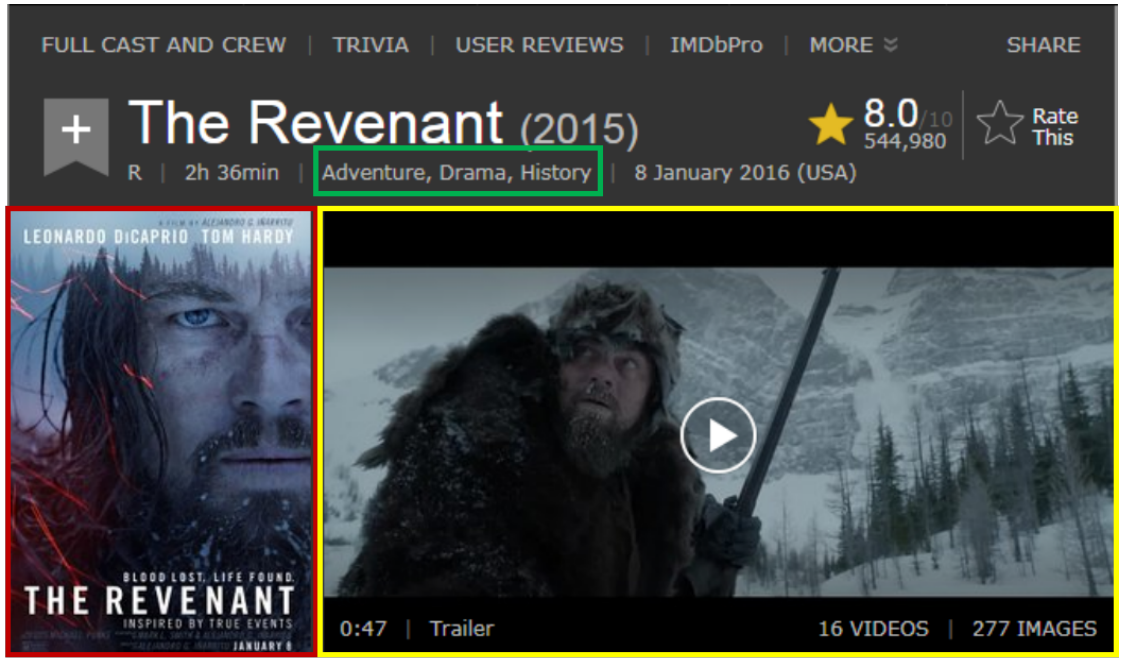
Introduction: Single-label vs. Multi-label

Table: Multi-label $Y \subset \{\lambda_1, ..., \lambda_L\}$

X1	X2	X3	X4	X5	Y1	Y2	Y3	Y4
1	0.1	3	1	0	0	1	1	0
0	0.9	1	0	1	1	0	0	0
0	0.0	1	1	0	0	1	0	0
1	0.8	2	0	1	1	0	0	1
1	0.0	2	0	1	0	0	0	1
0	0.0	3	1	1	?	?	?	?

Applications: Text Categorization

- For example, the IMDB dataset: Textual moive plot summaries associated with genres (labels)



Director: Alejandro G. Iñárritu
Writers: Mark L. Smith (screenplay), Alejandro G. Iñárritu (screenplay) | 1 more credit »
Stars: Leonardo DiCaprio, Tom Hardy, Will Poulter | See full cast & crew »

Storyline

While exploring uncharted wilderness in 1823, legendary frontiersman Hugh Glass sustains injuries from a brutal bear attack. When his hunting team leaves him for dead, Glass must utilize his survival skills to find a way back home while avoiding natives on their own hunt. Grief-stricken and fueled by vengeance, Glass treks through the wintry terrain to track down John Fitzgerald, the former confidant who betrayed and abandoned him.*Written by Jwelch5742*

Applications: Text Categorization

- For example, the IMDB dataset: Textual moive plot summaries associated with genres (labels)

example	*abandoned* X_1	*accident* X_2	...	*violent* X_{1000}	*wedding* X_{1001}	horror Y_1	romance Y_2	...	comedy Y_{27}	action Y_{28}
1	1	0	...	0	1	0	1	...	0	0
2	0	1	...	1	0	1	0	...	0	0
3	0	0	...	0	1	0	1	...	0	0
4	1	1	...	0	1	1	0	...	0	1
5	1	1	...	0	1	0	1	...	0	1
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋱	⋮	⋮
120919	1	1	...	0	0	0	0	...	0	1

- binary bag-of-words representation

Applications: Text Categorization

- For example, the IMDB dataset: Textual moive plot summaries associated with genres (labels)

example	*abandoned* X_1	*accident* X_2	...	*violent* X_{1000}	*wedding* X_{1001}	Y
1	1	0	...	1	0	{romance, comedy }
2	0	1	...	0	1	{horror}
3	0	0	...	1	0	{romance}
4	1	1	...	0	1	{horror, action}
5	1	0	...	0	1	{action}
⋮	⋮	⋮	⋱	⋮	⋮	⋮
120919	1	0	...	0	1	{action}

- binary bag-of-words representation

Applications: Text Categorization

- For example, the news...

- For example, REUTERS
- Reuters collection, newswire stories into 103 topic codes

Applications: E-mail

- Enron, e-mails messages made public from the Enron corporation.
- "a few beers after work?" $\boxed{work}$ $\boxed{personal}$ $\boxed{important}$
- For example, the **UC Berkeley Enron Email Analysis Project** multi-labeled 1702 Enron e-mails into 53 categories:
  - Company Business, Strategy, etc.
  - Purely Personal
  - Forwarded emails
  - ...
  - Company image - current
  - ...
  - Jokes, humor (related to business)

Multi-Label Classification

- Images are labeled to indicate
  - multiple concepts
  - multiple objects
  - multiple people

- e.g., Scene data with concept labels
  $\subseteq \{\text{beach, sunset, foliage, field, mountain, urban}\}$

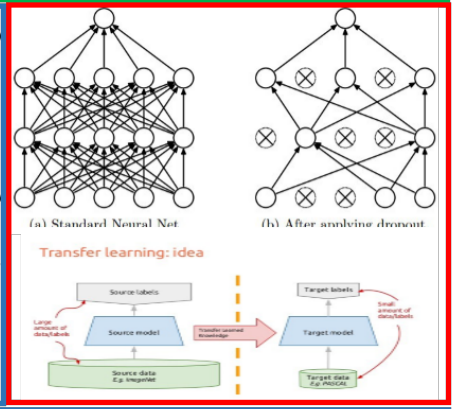
Applications: Webpages

- The content of a webpage may belong to multiple subjects: **Backpropagation, CNN, deep learning, gradient descent**, etc.

The 10 Deep Learning Methods AI Practitioners Need to Apply

Tags: Backpropagation, Convolutional Neural Networks, Deep Learning, Gradient Descent, LSTM, Neural Networks, Transfer Learning

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem.



•Andrew Beam's **"Deep Learning 101"**
•Andrey Kurenkov's **"A Brief History of Neural Nets and Deep Learning"**
•Adit Deshpande's **"A Beginner's Guide to Understanding Convolutional Neural Networks"**
•Chris Olah's **"Understanding LSTM Networks"**
•Algobean's **"Artificial Neural Networks"**
•Andrej Karpathy's **"The Unreasonable Effectiveness of Recurrent Neural Networks"**

Applications: Audio

- Labelling music/tracks with genres/voices, concepts, etc.

- e.g., Emotions dataset, audio tracks labelled with different moods, among:
  {
  - amazed-surprised
  - happy-pleased
  - relaxing-calm
  - quiet-still
  - sad-lonely
  - angry-aggressive

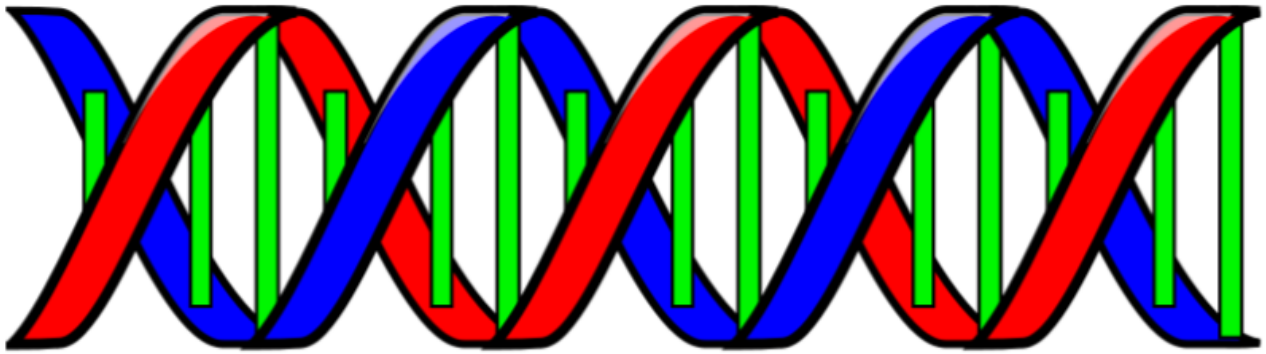
  }

Applications: Medical

- Medical Diagnosis

- medical history, symptoms $\rightarrow$ diseases/ailments, e.g., Medical dataset
- clinical free text reports by radiologists
- lalbel assignment out of 45 ICD-9-CM codes

Applications: Bioinformatics



- Genes are associated with biological functions
- E.g., the Yeast dataset: 2,417 genes,, described by 103 attributes, labeled into 14 groups of the FunCAt functional catalogue

Introduction: Notations/Labels as Items in a Set

- Input $\mathcal{X} = \mathbb{R}^D$, Labelset $\mathcal{Y} = \{\lambda_1, ..., \lambda_L\}$, label assignment $Y \subseteq \mathcal{Y}$
- We have set ofo training examples $\mathcal{D} = \{(\mathbf{x}^{(i)}, Y^{(i)})\}_{i=1}^{N} =$

$$\underbrace{\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix}}_{\mathbf{X} \in \mathcal{X}^N} \underbrace{\begin{bmatrix} Y^{(1)} \\ Y^{(2)} \\ \vdots \\ Y^{(N)} \end{bmatrix}}_{\mathbf{Y} \in \mathcal{Y}^N}$$

where

- $\mathbf{x}^{(i)} = [x_1, ..., x_D] \in \mathcal{X}$ is the representation of a data instance
- $Y^{(i)} \subset \mathcal{Y}$ is some label set. For example, $Y^{(1)} = \{\lambda_1, \lambda_4, \lambda_8\}$ are the labels relevant to $\mathbf{x}^{(1)}$

Introduction: Notations/Labels as Variables

- Input $\mathcal{X} = \mathbb{R}^D$, Output $\mathcal{Y} = \{0, 1\}^L$, label assignment $Y \subseteq \mathcal{Y}$
- We have set ofo training examples $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N} =$

$$\underbrace{\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix}}_{\mathbf{X} \in \mathcal{X}^N} \underbrace{\begin{bmatrix} y_1^{(1)} & y_2^{(1)} & \cdots & y_L^{(1)} \\ y_1^{(2)} & y_2^{(2)} & \cdots & y_L^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{(N)} & y_2^{(N)} & \cdots & y_L^{(N)} \end{bmatrix}}_{\mathbf{Y} \in \mathcal{Y}^N}$$

  - $\mathbf{x}^{(i)} = [x_1, ..., x_D] \in \mathcal{X}$ is the representation of a data instance
  - $\mathbf{y}^{(i)} = [y_1, ..., y_L] \in \mathcal{Y}$ is some label vector, where $y_j \in \{0, 1\}$.
  - Equivalent notation (for $L = 8$): $Y^{(i)} = \{\lambda_1, \lambda_4, \lambda_8\} \longleftrightarrow \mathbf{y}^{(i)} = [1, 0, 0, 1, 0, 0, 0, 1]$

Introduction: Notation / Labels as Variables

- **Training or Building a model**: Use training set $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N} =$ to build function or classifier

$$h : \mathcal{X} \to \mathcal{Y}$$

- **Testing or Prediction**: For a test instance $\mathbf{x}_t$, we obtain the prediction

$$\hat{\mathbf{y}} = h(\mathbf{x}_t)$$

- **Evaluation**: If we have the true classification $\mathbf{y}$ available, we then compare it to $\hat{\mathbf{y}}$ and gauge accuracy (more on this later)

Multi-Label Data: Datasets

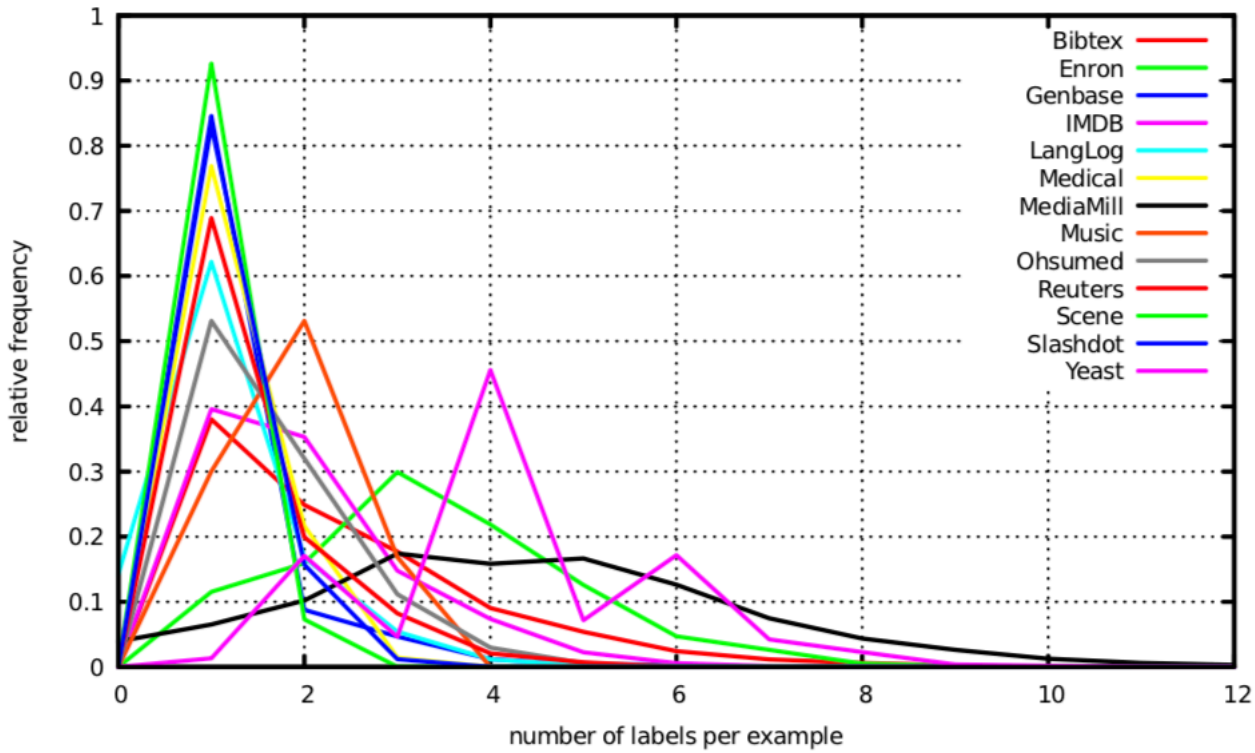
	$\mathcal{X}$ (data inst.)	$\mathcal{Y}$ (labels)	L	N	D	LC
Music	audio data	emotions	6	593	72	1.87
Scene	image data	scene labels	6	2407	294	1.07
Yeast	genes	biological fns	14	2417	103	4.24
Genbase	genes	biological fns	27	661	1185	1.25
Medical	medical text	diagnoses	45	978	1449	1.25
Enron	e-mails	labels, tags	53	1702	1001	3.38
Reuters	news articles	categories	103	6000	500	1.46
TMC07	textual reports	errors	22	28596	500	2.16
Ohsumed	medical articles	disease cats.	23	13929	1002	1.66
IMDB	plot summaries	genres	28	120919	1001	2.00
20NG	posts	news groups	20	19300	1006	1.03
MediaMill	video data	annotations	101	43907	120	4.38
Del.icio.us	bookmarks	tags	983	16105	500	19.02

- L: number of labels
- N: number of examples
- D: number of input feature attributes
- Label Cardinality (LC) $\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L} y_j^{(i)}$: Average number of labels per example

Multi-label Data: Statistics

- L: number of labels
- N: number of examples
- D: number of input feature attributes
- Label Cardinality (LC): $LC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L} y_j^{(i)}$: Average number of labels per example
- Label Density: $\frac{LC}{L}$ (LC divided by the number of labels)
- Distinct labelsets: proportion of labelsets that are distinct
- Most frequent labelset: proportion of instances that have most frequent labelset

Multi-Label Data



- The proportion of instances in each dataset relevant to 0,1,2,...,12 of L possible labels, most are relevant to only a few! i.e., Label Cardinality $\ll L$

Multi-Label Data

There are dependencies (i.e., correlations, relationships, co-occurences) among labels

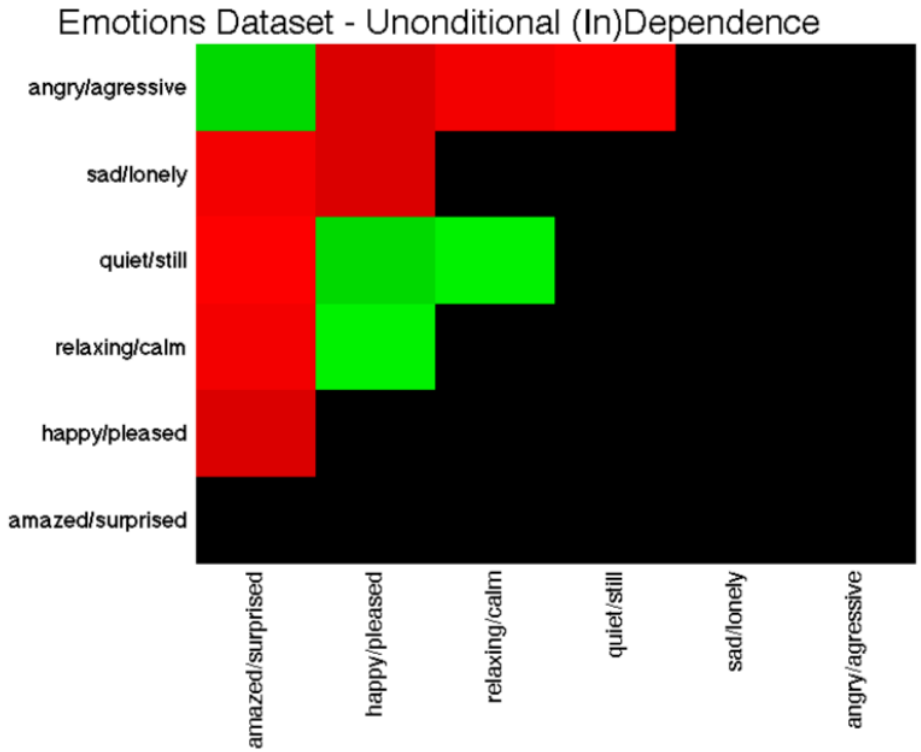
- e.g., {relaxing-calm, quiet-still} vs. {relaxing-calm, angry-aggressive}
- e.g., {beach, sunset} vs. {beach, field}

From the IMDb dataset:

- $P(family)P(adult) = 0.068 \cdot 0.015 = 0.001 (\approx 121$ movies)
- $P(family)P(adult) = 0.0$ (0 movies!)

On most datasets:

- $P(\mathbf{y} = [1, 1, 1, 1, 1, 1]) = 0$



Person's correlation coefficient $P_{Y_j, Y_k} = \frac{\text{cov}(Y_j, Y_k)}{\sigma_{Y_j} \sigma_{Y_k}}$ on Music

Challenges in Multi-label Classification

The main challenges are to

- exploit label dependencies
- do this efficiently

Introduction: Methods for Multi-label Classification

Problem Transformation Methods

- Transforms the multi-label problem into single-label problem(s)
- Use any off-the-shelf single-label classifier to suit requirements
- i.e., **Adapt your data to the algorithm**

Algorithm Adaptation Methods

- Adapt a single-label algorithm to produce multi-label outputs
- Benefit from specific classifier advantages (e.g., efficiency)
- i.e., **Adapt your algorithm to the data**

Many methods involve a mix of both approaches

Problem Transformation

For example,

- Binary Relevance: L binary problems (one vs. all)
- Label Powerset: one multi-class problem of 2^L class-values
- Pairwise: $\frac{L(L-1)}{2}$ binary problems (all vs. all)
- Copy-Weight: one multi-class problem of L class values

At training time, with $\mathcal{D}$:

- Transform the multi-label training data to single-label data
- Learn from the single-label transformed data

At testing time, for $\mathbf{x}_t$:

- Make single-label predictions
- Translate these into multi-label predictions

Binary Relevance (BR)

In the old days . . .

$\mathbf{X}$	Y_1	Y_2	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	1	0	0
$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	0	0	1

. . . just make L separate binary problems (one for each label):

$\mathbf{X}$	Y_1
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_2
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_3
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_4
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	1

and train with any off-the-shelf binary classifier.

Binary Relevance (BR)

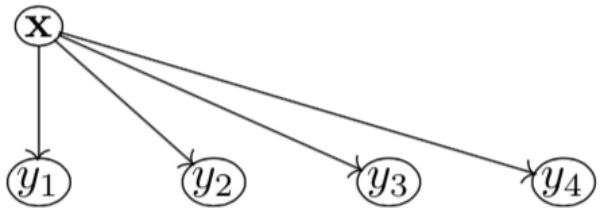
$\mathbf{X}$	Y_1
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_2
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_3
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_4
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	1

Prediction: $\hat{\mathbf{y}} = [h_1(\tilde{\mathbf{x}}), \ldots, h_L(\tilde{\mathbf{x}})]$



Binary Relevance (BR)

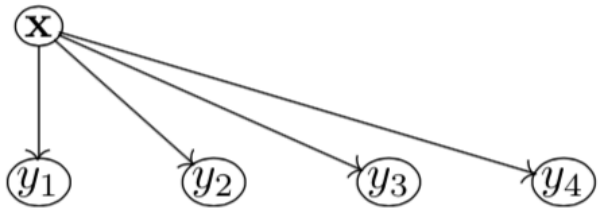
$\mathbf{X}$	Y_1
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_2
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_3
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_4
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	1

Prediction: $\hat{\mathbf{y}} = [h_1(\tilde{\mathbf{x}}), \ldots, h_L(\tilde{\mathbf{x}})]$



- Does not model label dependency
- Class imbalance, e.g., $P(\neg\text{family}) \gg P(\text{family})$

BR-Improvements

- Stacked BR (2BR) [Godbole and Sarawagi, 2004]: stack another BR on top, predict:

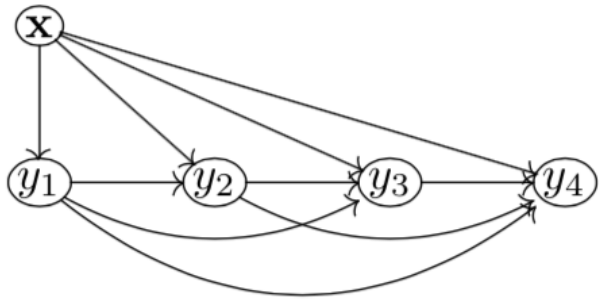
$$\hat{\mathbf{y}} = \mathbf{h}^2(\mathbf{h}^1(\mathbf{x}_t))$$

- For example, given $\mathbf{x}_t$

	$\hat{Y}_1$	$\hat{Y}_2$	$\hat{Y}_3$	$\hat{Y}_4$
$\mathbf{h}^1(\mathbf{x}_t)$	1	0	0	1
$\mathbf{h}^2(\mathbf{h}^1(\mathbf{x}_t))$	1	0	0	0

BR-Improvements

- Chain Classifier (CC) [Cheng et al., 2010, Read et al., 2011]



- Like BR, make L binary problems, but include previous predictions as feature attributes

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_1	Y_2
$x^{(1)}$	0	1
$x^{(2)}$	1	0
$x^{(3)}$	0	1
$x^{(4)}$	1	0
$x^{(5)}$	0	0

X	Y_1	Y_2	Y_3
$x^{(1)}$	0	1	1
$x^{(2)}$	1	0	0
$x^{(3)}$	0	1	0
$x^{(4)}$	1	0	0
$x^{(5)}$	0	0	0

X	Y_1	Y_3	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

Label Powerset Method (LP)

- To model label correlations, we can make a single multi-class problem with 2^L possible class values,
- and train with any off-the-shelf multi-class classifier

X	Y_1	Y_2	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	1	1	0
$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	0	0	1

$\Rightarrow$

X	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	1001
$\mathbf{x}^{(5)}$	0001

Issues with LP

$\mathbf{X}$	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	1001
$\mathbf{x}^{(5)}$	0001

- complexity: many class labels
- imbalance: not many examples per class label
- overfitting: how to predict new value?

LP Improvements

X	$Y \in 2^L$
$x^{(1)}$	0110
$x^{(2)}$	1000
$x^{(3)}$	0110
$x^{(4)}$	1001
$x^{(5)}$	0001

Ensembles of RAndom k-labEL subsets (RAkEL) [Tsoumakas and Vlahavas, 2007]

- Do LP on M subsets $\subset \{\lambda_1, ..., \lambda_L\}$ of size k

X	$Y \in 2^k$
$x^{(1)}$	011
$x^{(2)}$	100
$x^{(3)}$	011
$x^{(4)}$	100
$x^{(5)}$	000

X	$Y \in 2^k$
$x^{(1)}$	010
$x^{(2)}$	100
$x^{(3)}$	010
$x^{(4)}$	101
$x^{(5)}$	001

X	$Y \in 2^k$
$x^{(1)}$	010
$x^{(2)}$	100
$x^{(3)}$	010
$x^{(4)}$	101
$x^{(5)}$	001

X	$Y \in 2^k$
$x^{(1)}$	110
$x^{(2)}$	000
$x^{(3)}$	110
$x^{(4)}$	001
$x^{(5)}$	001

Ensembles of RAndom k-labEL subsets (RAkEL) [Tsoumakas and Vlahavas, 2007]

- Do LP on M subsets $\subset \{\lambda_1, ..., \lambda_L\}$ of size k

X	$Y \in 2^k$
$x^{(1)}$	011
$x^{(2)}$	100
$x^{(3)}$	011
$x^{(4)}$	100
$x^{(5)}$	000

X	$Y \in 2^k$
$x^{(1)}$	010
$x^{(2)}$	100
$x^{(3)}$	010
$x^{(4)}$	101
$x^{(5)}$	001

X	$Y \in 2^k$
$x^{(1)}$	010
$x^{(2)}$	100
$x^{(3)}$	010
$x^{(4)}$	101
$x^{(5)}$	001

X	$Y \in 2^k$
$x^{(1)}$	110
$x^{(2)}$	000
$x^{(3)}$	110
$x^{(4)}$	001
$x^{(5)}$	001

- 2^k problems much easier to deal with than 2^L (but still models label dependencies)
- use k and M (number of models) to trade-off dependency modelling and scalability

LP Improvements

$\mathbf{X}$	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	1001
$\mathbf{x}^{(5)}$	0001

Ensembles of Pruned Sets (EPS) [Read et al., 2008]

- prune out infrequent labelsets, replace with sampled frequent sets

$\mathbf{X}$	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	0001
$\mathbf{x}^{(5)}$	0001

$\mathbf{X}$	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	0001
$\mathbf{x}^{(4)}$	1000
$\mathbf{x}^{(5)}$	0001

LP Improvements

Ensembles of Pruned Sets (EPS) [Read et al., 2008]

- prune out infrequent labelsets, replace with sampled frequent sets

X	$Y \in 2^L$
$x^{(1)}$	0110
$x^{(3)}$	0110
$x^{(4)}$	0001
$x^{(5)}$	0001

X	$Y \in 2^L$
$x^{(1)}$	0110
$x^{(2)}$	1000
$x^{(3)}$	0110
$x^{(4)}$	0001
$x^{(4)}$	1000
$x^{(5)}$	0001

- best used in an ensemble (of M models), parameterised by
  - p: a combination occurring $\leq p$ is infrequent
  - n: replace them with n subsampled frequent sets (if available)
- keep (most) label dependency information, reduce complexity and other LP issues

Ensemble-based Voting

Ensemble methods (e.g., RAkEL, EPS) make prediction via a voting scheme. For some test instance $\mathbf{x}_t$:

	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
$\mathbf{h}^1(\mathbf{x}_t)$	1	0	1	
$\mathbf{h}^2(\mathbf{x}_t)$		1	1	0
$\mathbf{h}^3(\mathbf{x}_t)$	1		1	0
$\mathbf{h}^4(\mathbf{x}_t)$	1	0		0
$\mathbf{h}(\mathbf{x}_t)$	3	1	3	0
$\hat{\mathbf{y}}$	1	0	1	0

- more predictive power (ensemble effect)
- can predict new label combinations

Pairwise Binary (PW)

Another binary transformation, make $\frac{L(L-1)}{2}$ binary classifiers (all-vs-all)

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	1	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

X	Y_{1v2}
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1

X	Y_{1v3}
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(4)}$	1

X	Y_{1v4}
$x^{(2)}$	1
$x^{(5)}$	0

X	Y_{2v3}
$x^{(3)}$	1

X	Y_{2v4}
$x^{(1)}$	1
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_{3v4}
$x^{(1)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

where each model is trained based on examples annotated by at least one of the labels, but not both

Pairwise Binary (PW)

X	Y_{1v2}
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1

X	Y_{1v3}
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(4)}$	1

X	Y_{1v4}
$x^{(2)}$	1
$x^{(5)}$	0

X	Y_{2v3}
$x^{(3)}$	1

X	Y_{2v4}
$x^{(1)}$	1
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_{3v4}
$x^{(1)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

- **Prediction**: $y_{j,k} = \mathbf{h}_{j,k}(\mathbf{x}_t)$ for all $1 \leq j < k \leq L$

$$y_{j,k} = \begin{cases} 0, & \lambda_j > \lambda_k \\ 1, & \lambda_k > \lambda_j \end{cases}$$

- Issues:
  - this produces pairwise rankings, how to get a labelset?
  - how much sense does it make to find a decision boundary between overlapping labels?
  - can be expensive in terms of numbers of classifiers $\frac{(L(L-1))}{2}$

PW Improvements

- Calibrated Label Ranking CLR ([Fürnkranz et al., 2008]): Calibrate a 'virtual label' λ_0 to split the ranking:

$$\lambda_1 \succ \lambda_3 \succ \lambda_0 \succ \lambda_4 \succ \lambda_2 ...$$

- Can also have a four-class problem

$$Y_{j,k} \in \{00, 01, 10, 11\}$$

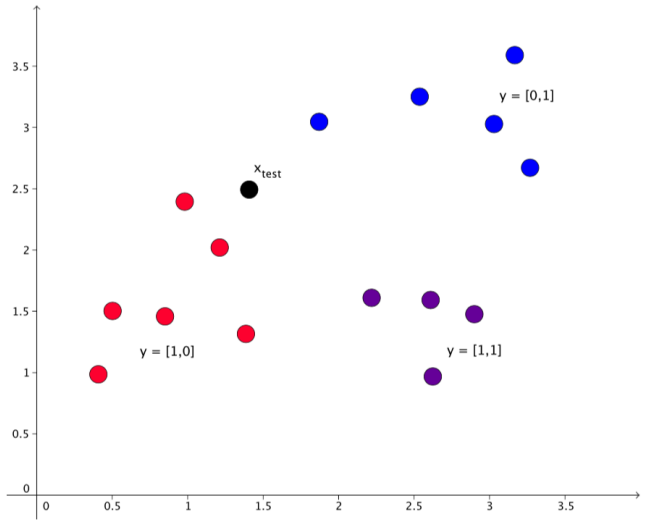
  - like pairwise LP
  - larger subproblems than PW

Algorithm Adaptaion

- Take your favorite classifier, make it multi-label capable
- Adapt the traditional single-label classification algorithms to multi-label classification
- Fit the algorithm to data
  - k-Nearest Neighbors
  - Decision Trees
  - Neural Networks
  - Support Vector Machines

k-Nearest Neighbours

- kNN assigns to $\mathbf{x}_t$ the majority class of the k nearest neighbours
- MLkNN [Zhang and Zhou, 2007] assigns to $\mathbf{x}_t$ the most common labels of the k nearest neighbours



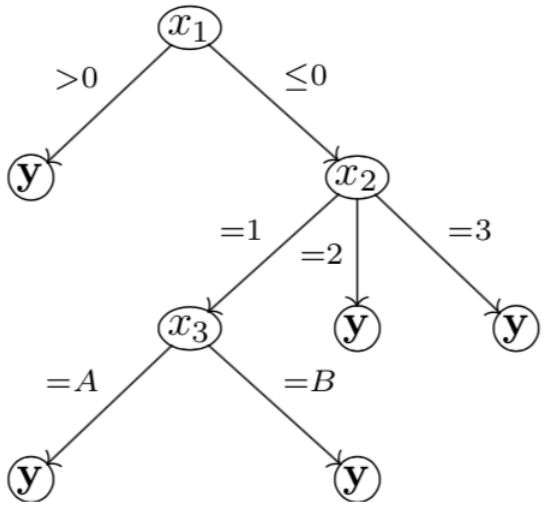
- MLkNN combined with Bayesian inference (MAP principle)
- $y_j = \begin{cases} 0, \text{if } P(c_{j,\mathbf{x}}|y_j = 1)p(y_j = 1) \geq P(c_{j,\mathbf{x}}|y_j = 0)p(y_j = 0) \\ 1, \text{otherwise} \end{cases}$
- where $c_{j,\mathbf{x}}$ is the number of examples in neighbourhood of $\mathbf{x}$ with $y_j = 1$, and probabilities are estimated from training data

Decision Tree

- Multi-label C4.5 [Clare and King, 2001]: Extension of the popular C4.5 decision tree algorithm; with multi-label entropy:

$$H_{ML}(\mathcal{D}) = \sum_{j=1}^{L} P(y_j)\log(P(y_j)) + (1-P(y_j))\log(1-P(y_j))$$

- construct just like C4.5
- allows multiple labels at the leaves
- work well in an ensemble / random forest



Maximum Margin Method

RankSVM, a Maximum Margin approach [Elisseeff and Weston, 2002]

- One classifier for each label $h_j(\mathbf{x}) = \mathbf{w}_j^T\mathbf{x} + b_j$
- use kernel trick for non-linearity
- define multi-label margin, for each $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ in the training set $\mathcal{D}$

$$\min_{(\mathbf{x}^{(i)},\mathbf{y}^{(i)})\in\mathcal{D}} \min_{j,k} \frac{\mathbf{w}_j^T\mathbf{x} + b_j - \mathbf{w}_k^T\mathbf{x} - b_k}{\|\mathbf{w}_j - \mathbf{w}_k\|}$$

- solve with quadratic programming
- improved performance over BR with SVMs

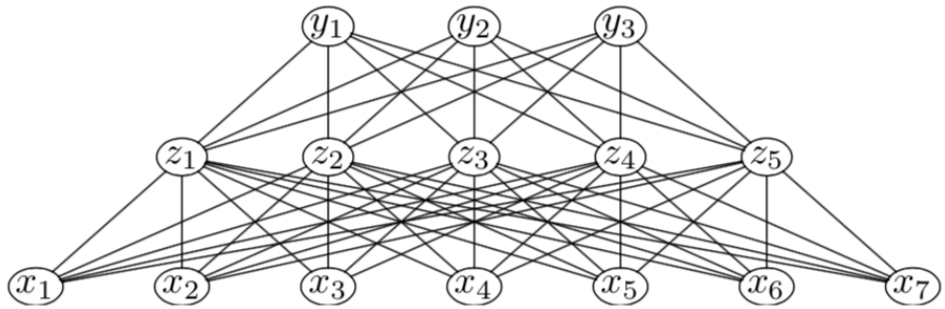
Neural Networks

BPMLL [Zhang and Zhou, 2006] is

- a regular back-prop. neural network with multiple outputs
- trained with gradient descent + error back-propagation
- with an error function based on ranking (relevant labels should be ranked higher than non-relevant labels)

$$E = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(j,k)\in Y_i \times \bar{Y}_i} \exp(-(y_k^{(i)} - y_j^{(i)}))$$

- one hidden layer
- one output per label



Multi-label Evaluation

In single-label classification, accuracy is just:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}[\hat{y}^{(i)} = y^{(i)}]$$

where $\mathcal{I}[\cdot]$ returns 1 if condition holds, 0 otherwise
In multi-label classification, e.g.,:

$$\hat{\mathbf{y}} = [0, 1, 0, 0, 1, 0]$$
$$\mathbf{y} = [1, 1, 0, 0, 1, 0]$$

How to evaluate the performance?

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

- Hamming Loss:

$$\begin{aligned} \text{HLoss} &= \frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}\mathcal{I}[\hat{y}_j^{(i)} = y_j^{(i)}] \\ &= 0.2 \end{aligned}$$

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

- 0/1 Loss:

$$\begin{aligned} \text{0/1 Loss} &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}[\hat{\mathbf{y}}^{(i)} = \mathbf{y}^{(i)}] \\ &= 0.6 \end{aligned}$$

- It is often used as Exact Match (1 - 0/1 Loss)

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

- Accuracy:

$$\begin{aligned} \text{Accuracy} &= \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{\mathbf{y}}^{(i)} \wedge \mathbf{y}^{(i)}|}{|\hat{\mathbf{y}}^{(i)} \vee \mathbf{y}^{(i)}|} \\ &= \frac{1}{5}(\frac{1}{3}+1+1+\frac{1}{2}+\frac{1}{2}) = 0.67 \end{aligned}$$

- where $\vee$ and $\wedge$ are the logical OR and AND operations, applied vector-wise

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.1 0.2]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]

- Ranking Loss: to encourage good ranking, evaluates the average fraction of label pairs miss-ordered for $\tilde{\mathbf{x}}$

Ranking Loss =

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{(j,k):y_j>y_k}\frac{1}{|Y_i||\bar{Y}_i|}\left(\mathcal{I}[r_i(j)<r_i(k)]+\frac{1}{2}\mathcal{I}[r_i(j)=r_i(k)]\right)$$

- where $r_i(j) :=$ ranking of label j for instance $\tilde{\mathbf{x}}^{(i)}$, Y_i (or $\bar{Y}_i$) is the set of related (or unrelated) label set

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	$r(1) < r(3) < r(4) < r(2)$
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	$r(2) = r(4) < r(1) < r(3)$
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	$r(1) < r(4) < r(3) < r(2)$
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.1 0.2]	$r(2) < r(4) < r(3) = r(1)$
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	$r(1) = r(5) < r(2) < r(3)$

- Ranking Loss: to encourage good ranking, evaluates the average fraction of label pairs miss-ordered for $\tilde{\mathbf{x}}$

$$\text{Ranking Loss} = \frac{1}{5}(\frac{1}{4} + \frac{0}{4} + \frac{0}{4} + \frac{1.5}{4} + \frac{1}{4})$$

Multi-label Evaluation Metrics

Other metrics used in the literature:

- One error –if top ranked label is not in set of true labels
- Coverage –average "depth" to cover all true labels
- Precision
- Recall
- Macro-averaged F1 (ordinary averaging of a binary measure)
- Micro-averaged F1 (labels as different instances of a 'global' label)
- Precision vs. Recall curves

Ref: TKDE2014 - A Review on Multi-Label Learning Algorithms

Multi-label Evaluation: Which Metric to Use?

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[1 0 1 1]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

- HAM. LOSS 0.3
- 0/1 LOSS 0.6

Multi-label Evaluation: Which Metric to Use?

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 **1** 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 **1** 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 1 **1** 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[1 0 1 **0**]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

Optimizing HAMMING LOSS ...

- HAM. LOSS 0.2
- 0/1 LOSS 0.8

Multi-label Evaluation: Which Metric to Use?

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[**0** **1** 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 **0** **1** 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[**0** **1** 1 1]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

Optimizing 0/1 LOSS ...

- HAM. LOSS 0.4
- 0/1 LOSS 0.4

Multi-label Evaluation: Which Metric to Use?

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[**0** **1** 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 **0** **1** 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[**0** **1** 1 1]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

- Hamming Loss can in principal be minimized without taking label dependence into account
- For 0/1 Loss label dependence must be taken into account
- Usually not be possible to minimize both at the same time
- For general evaluation, **use multiple and contrasting evaluation measures**

Methods that Output real values

Many methods return real values $\mathbf{h}(\tilde{\mathbf{x}}) \in \mathbb{R}^L$, which may be, e.g.,

- probabilistic information; or
- votes froom an ensemble process

Example: Prediction from ensemble of 3 multi-label models

For some test instance $\tilde{\mathbf{x}}$ ...

	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
$\mathbf{h}^1(\tilde{\mathbf{x}})$	1	0	1	0
$\mathbf{h}^2(\tilde{\mathbf{x}})$	0	1	1	0
$\mathbf{h}^3(\tilde{\mathbf{x}})$	1	0	1	0
$\mathbf{h}(\tilde{\mathbf{x}})$	2	1	3	0
$\equiv$	0.67	0.33	1.00	0.00
$\hat{\mathbf{y}} \in \{0,1\}^L$	?	?	?	?

We may want to evalute these directly, but we usually need to convert them to binary values $\hat{\mathbf{y}}$

Threshold Selection

Use a threshold of 0.5?

$$\hat{y}_j = \begin{cases} 1, & h_j(\widetilde{\mathbf{x}}) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathcal{I}[\mathbf{h}(\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Threshold Selection

Use a threshold of 0.5?

$$\hat{y}_j = \begin{cases} 1, & h_j(\widetilde{\mathbf{x}}) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathcal{I}[\mathbf{h}(\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

But it would eliminate two errors with a threshold of 0.4

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathcal{I}[\mathbf{h}(\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Use ad-hoc threshold, e.g., 0.5
- how to know wihch threshold to use?

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathcal{I}[\mathbf{h}(\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Select a threshold from an internal validation test, e.g., $\{0.1, 0.2, ..., 0.9\}$

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathcal{I}[\mathbf{h}(\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Calibrate a threshold such that $\text{LCARD}(\mathbf{Y}) \simeq \text{LCARD}(\widehat{\mathbf{Y}})$
  - e.g., training data has label cardinality of 1.7
  - set a threshold t such that the label cardinality of the test data is as close as possible to 1.7

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathcal{I}[\mathbf{h}(\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Calibrate L threshold such that $\text{LCARD}(\mathbf{Y}_j) \simeq \text{LCARD}(\widehat{\mathbf{Y}}_j)$
  - e.g., the frequency of label $y_j = 1$ is 0.3
  - set a threshold t_j such that $h_j(\tilde{\mathbf{x}}) \leq t_j \Leftrightarrow \hat{y}_j = 1$ with the frequency as close as possible to 0.3

SoftWare for Multi-Label Classification

- MEKA: A WEKA-based framework for multi-label classification and evaluation, http://meka.sourceforge.net
- MULAN: is an open-source Java library for learning from multi-label datasets, http://mulan.sourceforge.net/
- Many open resources of multi-label classification literatures are supplied by the researchers on their websites, e.g., http://cse.seu.edu.cn/people/zhangml/Resources.htm, http://manikvarma.org/downloads/XC/XMLRepository.html

Paper Reading

Read the following research papers, and give a presentation in class

1. M. Zhang and Z. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng., vol. 26, no. 8, pp. 1819–1837, 2014.
2. M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, Learning multi-label scene classification, Pattern Recognit., vol. 37, no. 9, pp. 1757–1771, 2004.
3. M. Zhang and Z. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognit., vol. 40, no. 7, pp. 2038–2048, 2007.
4. A. Elisseeff and W. Jason, A kernel method for multi-labelled classification, in Proc. Neural Inf. Process. Syst., 2001, pp. 681–687.
5. J. Fürnkranz, etc, Multilabel classification via calibrated label ranking, Mach. Learn., vol. 73, no. 2, pp. 133–153, 2008.
6. J. Read, B. Pfahringer, and G. Holmes, Multi-label classification using ensembles of pruned sets, in Proc. IEEE Int. Conf. Data Mining, 2008, pp. 995–1000.
7. M. Zhang and Z. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng., vol. 18, no. 10, pp. 1338–1351, 2006.
8. Y. K. Li, M. L. Zhang, and X. Geng, Leveraging implicit relative labeling importance information for effective multi-label learning, in Proc. IEEE Int. Conf. Data Mining, 2015, pp. 251–260.

Paper Reading

Outline of presentation, but not limilated

- Motivation
- Related work
- Method
- Result and Analysis
- Conclusion and future work