Data Mining & Machine Learning

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

Schedule

- Ensemble Methods
- Multi-Label Classifications

Schedule

- Ensemble Methods
- Multi-Label Classifications

Ensemble Methods

- **Basic idea** is to learn a set of models and to allow them to vote.
- **Advantage:** improvement in predictive accuracy.
- **Disadvantage:** it is difficult to understand an ensemble of models.
- **Note:** these ensemble methods can be used for both classifications and regressions

Ensemble Methods

- Bagging
- Boosting
  - AdaBoosting
  - Gradient Boosting
  - XGBoost (eXtreme Gradient Boosting)

Ensemble Methods

- Bagging
- Boosting
  - AdaBoosting
  - Gradient Boosting
  - XGBoost (eXtreme Gradient Boosting)

Bagging

- Process in bagging:
  - Define the size for training set, n
  - Sample several training sets of size *n* (instead of just having one training set of size *n*)
  - Build a classifier for each training set
  - Combine the classifier’s predictions by voting or averaging.
  - Note: you can use a same classification algorithm to build a classifier (e.g., KNN only) for each training set. Or, you can use different algorithms (e.g., KNN, Decision Tree, SVM, etc.) to build a classifier for each training set

Bagging

Sample Training sets

Build individual models

Voting or Averaging

Voting and Averaging

- Voting is used for classifications, and averaging is used for regressions
- Voting: Hard and Soft voting

Hard voting

Predictions:

Classifier 1 predicts class A

Classifier 2 predicts class B

Classifier 3 predicts class B

2/3 classifiers predict class B, so **class B is the ensemble decision**.

Soft voting

Predictions (identical to the earlier example, but now in terms of probabilities. Shown only for class A here because the problem is binary):

Classifier 1 predicts class A with probability 99%

Classifier 2 predicts class A with probability 49%

Classifier 3 predicts class A with probability 49%

The average probability of belonging to class A across the classifiers is `(99 + 49 + 49) / 3 = 65.67%`. Therefore, **class A is the ensemble decision**.

Example: Random Forest

- Random Forest is a bagging method where you utilize decision tree as classifiers

Sample Training sets

Build individual Trees

Voting or Averaging

Why does bagging work?

- Bagging reduces variance by voting/ averaging, thus reducing the overall expected error
  - In the case of classification there are pathological situations where the overall error might increase
  - Usually, the more classifiers the better

Ensemble Methods

- Bagging
- Boosting
  - AdaBoosting
  - Gradient Boosting
  - XGBoost (eXtreme Gradient Boosting)

Boosting

- No models is always the best learner. There are always weak learners – models which may have large classification errors
- General Ideas in Boosting
  - Learn a base model
  - Adjust training set based on the previous base model, and train the next model
  - Repeat the process above to get T models
  - Finally use all T models together to make predictions

Ensemble Methods

- Bagging
- Boosting
  - AdaBoosting
  - Gradient Boosting
  - XGBoost (eXtreme Gradient Boosting)

AdaBoosting

- Rough Idea
  1) Assign equal weights to all instances in training set
  2) Train a base model
  3) Adjust weights of instances in training set based on the previous model, e.g., assign more weights to the misclassified instances
  4) Train another model
  5) Repeat 3)-4) to get T models
  6) Combine all T models to make predictions

AdaBoosting

- 1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \dots, N$
- 2. For $m = 1, \dots, M$:
- (a) Fit a classifier $y_m(x)$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)$$

- Where $I(y_m(x_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(x_n) \neq t_n$ and 0 otherwise.

AdaBoosting

- (b) Evaluate the quantities

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

and then use these to evaluate

$$\alpha_m = \ln\{\frac{1-\varepsilon_m}{\varepsilon_m}\}$$

AdaBoosting

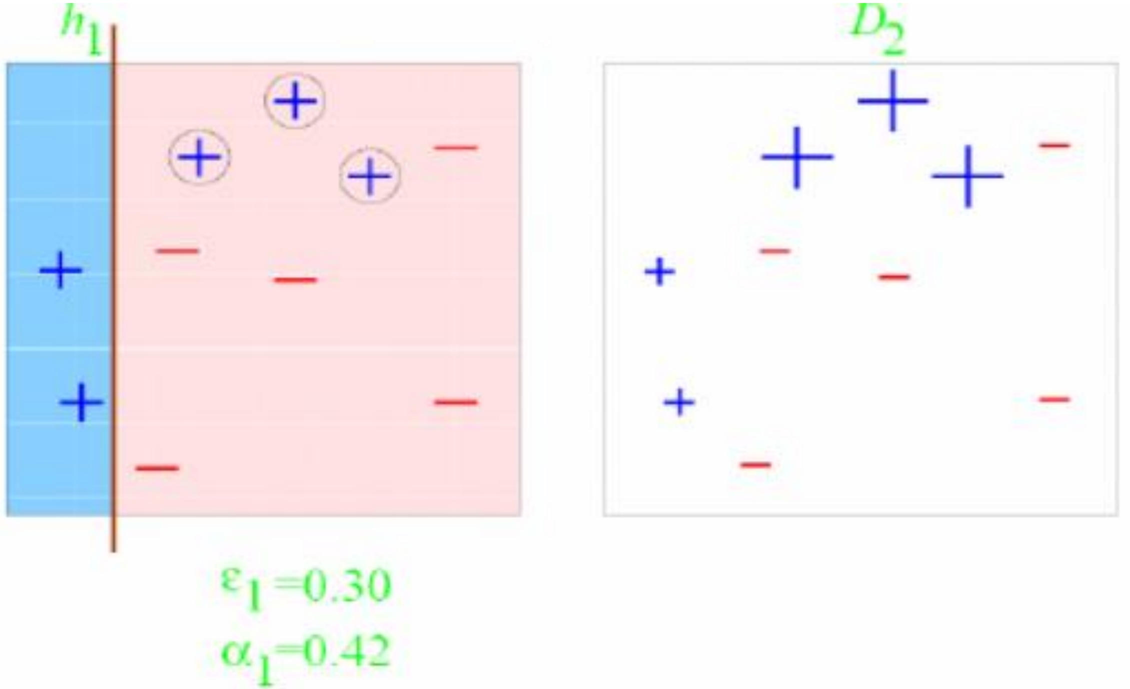
- (c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(x_n) \neq t_n)\}$$

- 3. Make predictions using the final model, which is given by

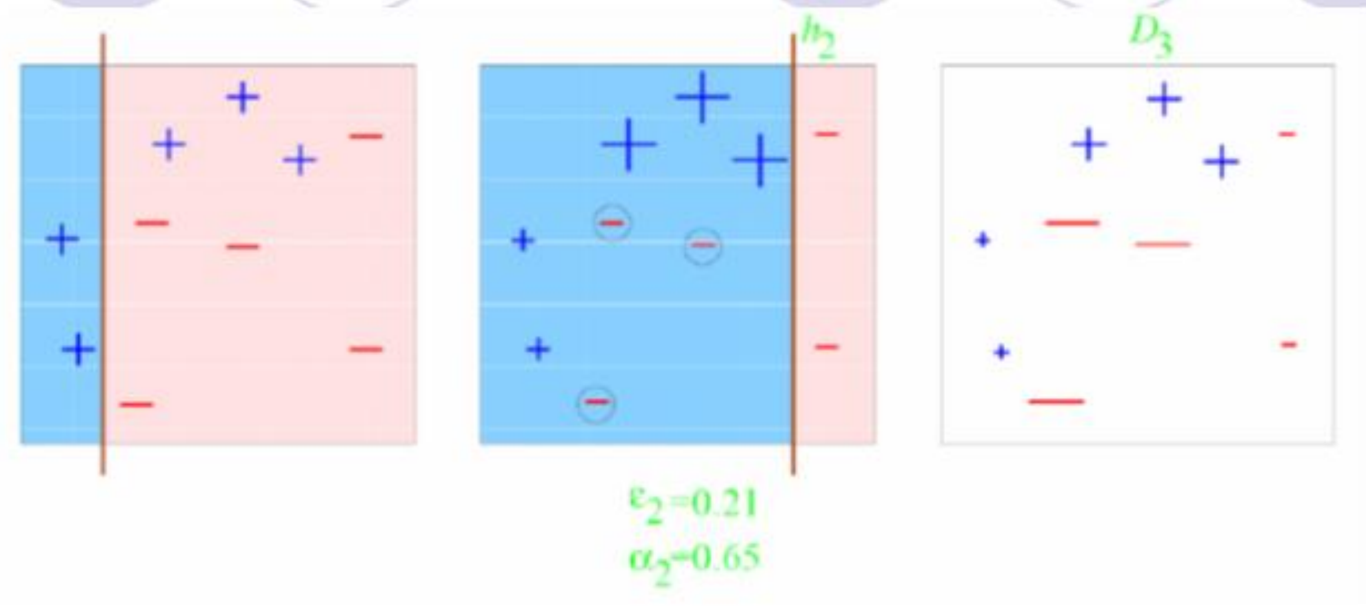
$$Y_M(x) = sign(\sum_{m=1}^{M} \alpha_m y_m(x))$$

AdaBoosting: Example



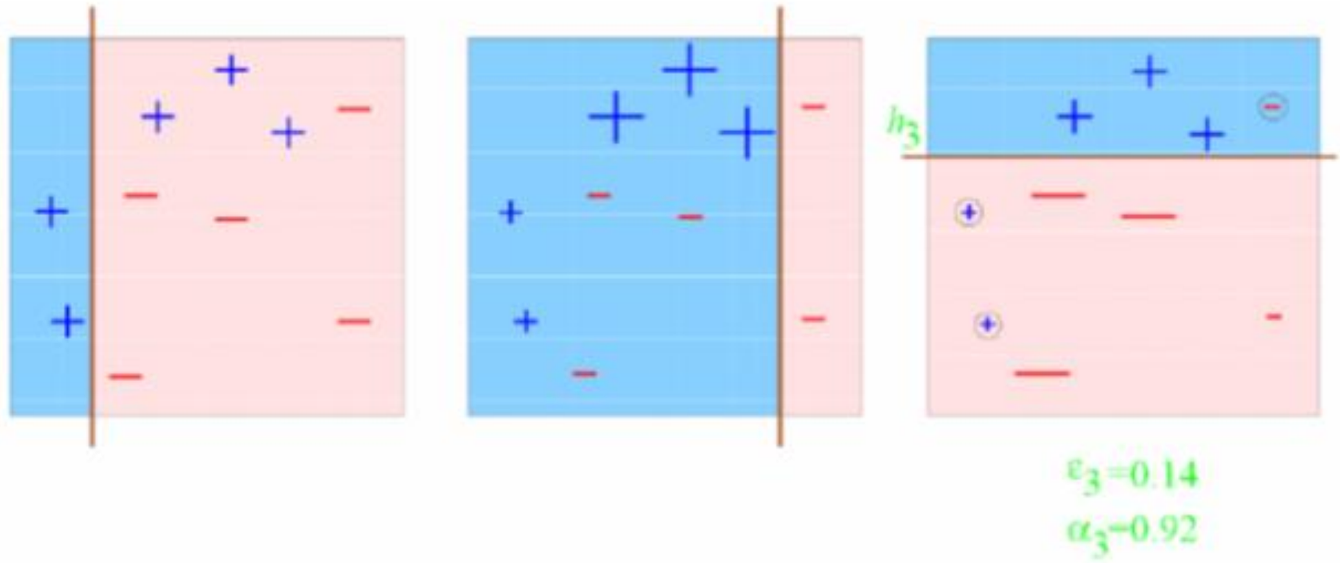
Round 1: Three “plus” points are not correctly classified;
They are given higher weights.

AdaBoosting: Example



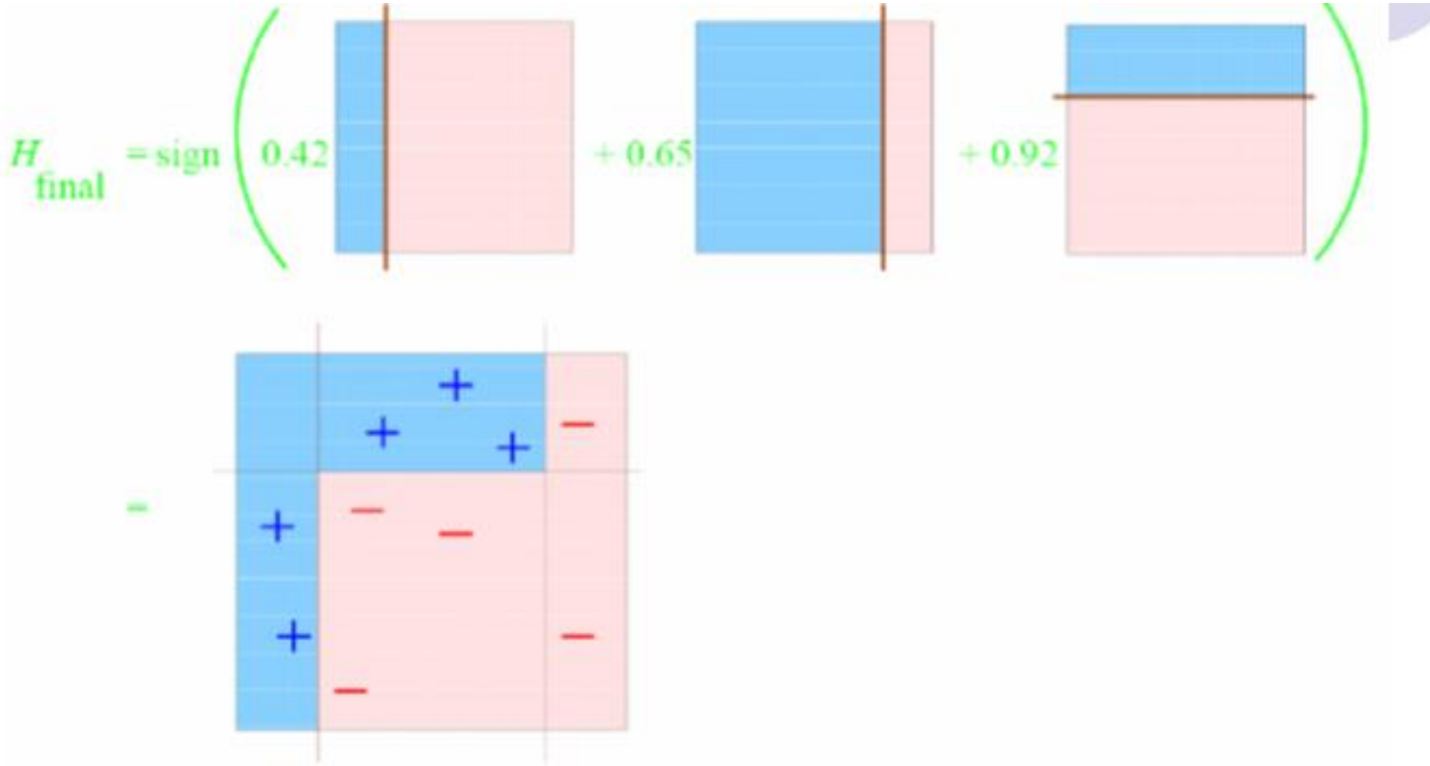
Round 2: Three "minuse" points are not correctly classified;
They are given higher weights.

AdaBoosting: Example



Round 3: One “minuse” and two “plus” points are not correctly classified;
They are given higher weights.

AdaBoosting: Example



Final Classifier: integrate the three "weak" classifiers and obtain a final strong classifier.

Ensemble Methods

- Bagging
- Boosting
  - AdaBoosting
  - Gradient Boosting
  - XGBoost (eXtreme Gradient Boosting)

Gradient Boosting

- Rough Idea
  - GB is still a boosting method
  - GB = Gradient Descent + Boosting
  - AdaBoosting vs GB
    - Both of them tried to improve weak learners iteratively
    - AdaBoosting tried to change the weights of misclassified instances. GB tried to take advantage of gradient descent of the loss functions (e.g., loss in SVM)
    - Both of them cane be used for classification and regressions, but GB is more powerful for regressions

Gradient Boosting

Gradient Boosting

- Fit an additive model (ensemble) $\sum_t \rho_t h_t(x)$ in a forward stage-wise manner.
- In each stage, introduce a weak learner to compensate the shortcomings of existing weak learners.
- In Gradient Boosting, "shortcomings" are identified by gradients. [negative gradients]
- Recall that, in Adaboost, "shortcomings" are identified by high-weight data points.
- Both high-weight data points and gradients tell us how to improve our model.

Ensemble Methods

- Bagging
- Boosting
  - AdaBoosting
  - Gradient Boosting
  - XGBoost (eXtreme Gradient Boosting)

More variants

- Gradient Boosting is much more powerful, and there are different variants of gradient boosting
  - GBDT (Gradient Boosting Decision Tree)
  - XGBoost (eXtreme Gradient Boosting)
  - LightGBM (Light Gradient Boosting Machine)
  - CatBoost (Categorical Boosting)

XGBoost (eXtreme Gradient Boosting)

- Gradient Boosting vs XGBoost
  - XGBoost is an improvement over GB
  - XGBoost supports distributed computing
  - XGBoost have several solutions to alleviate overfitting, e.g., L1, L2 regularization terms
  - XGBoost supports column subsampling which can improve performance
  - Many many more ….

Schedule

- Ensemble Methods
- Multi-Label Classifications

Classification

Binary classification: Is this a picture of the sea?

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

Classification

Multi-*class* classification: What is this a picture of?

$\in \{\texttt{sea}, \texttt{sunset}, \texttt{trees}, \texttt{people}, \texttt{mountain}, \texttt{urban}\}$

Classification

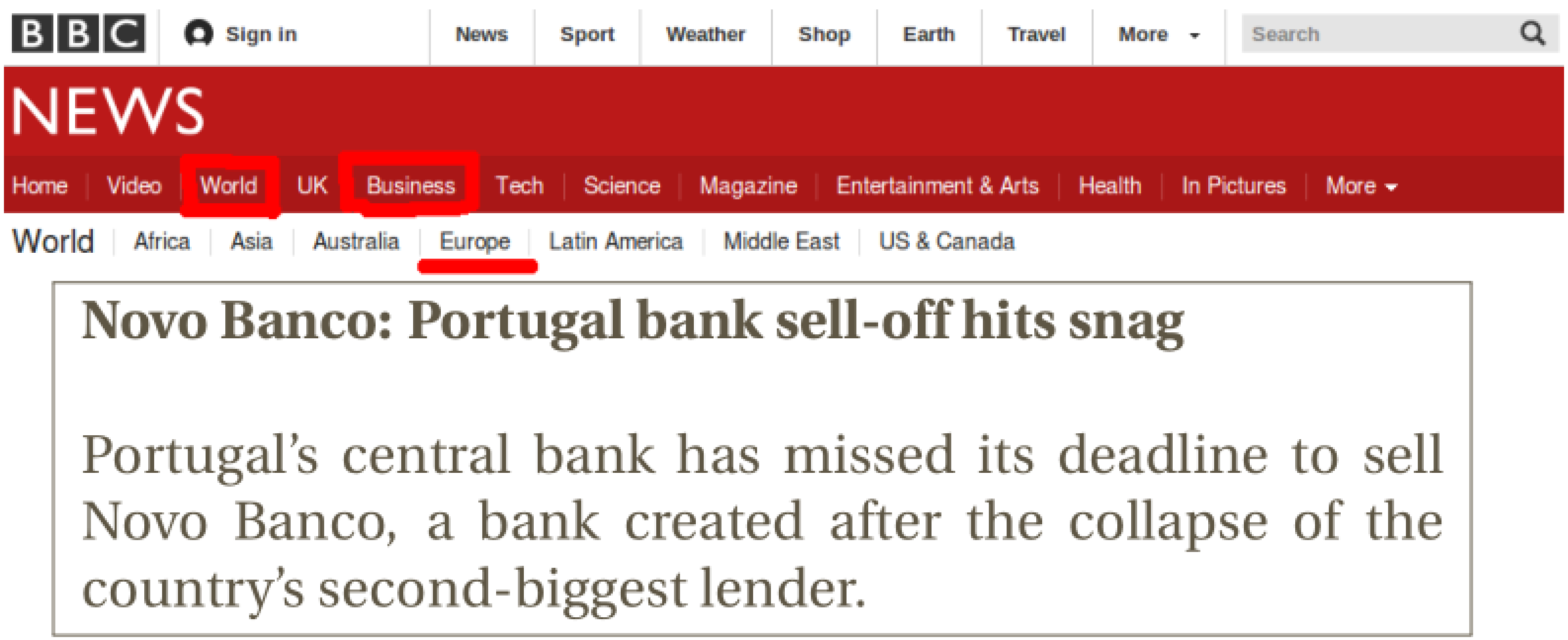
Multi-label classification: Which labels are relevant to this picture?

$$\subseteq \{\texttt{sea}, \texttt{sunset}, \texttt{trees}, \texttt{people}, \texttt{mountain}, \texttt{urban}\}$$

i.e., multiple labels per instance instead of a single label!

Multi-Label Classification: Applications

For example, the news . . .



Multi-Label Classification: Applications

For example, the IMDb dataset: Textual movie plot summaries associated with genres (labels).

Multi-Label Classification: Applications

Images are labelled to indicate

- multiple concepts
- multiple objects
- multiple people

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

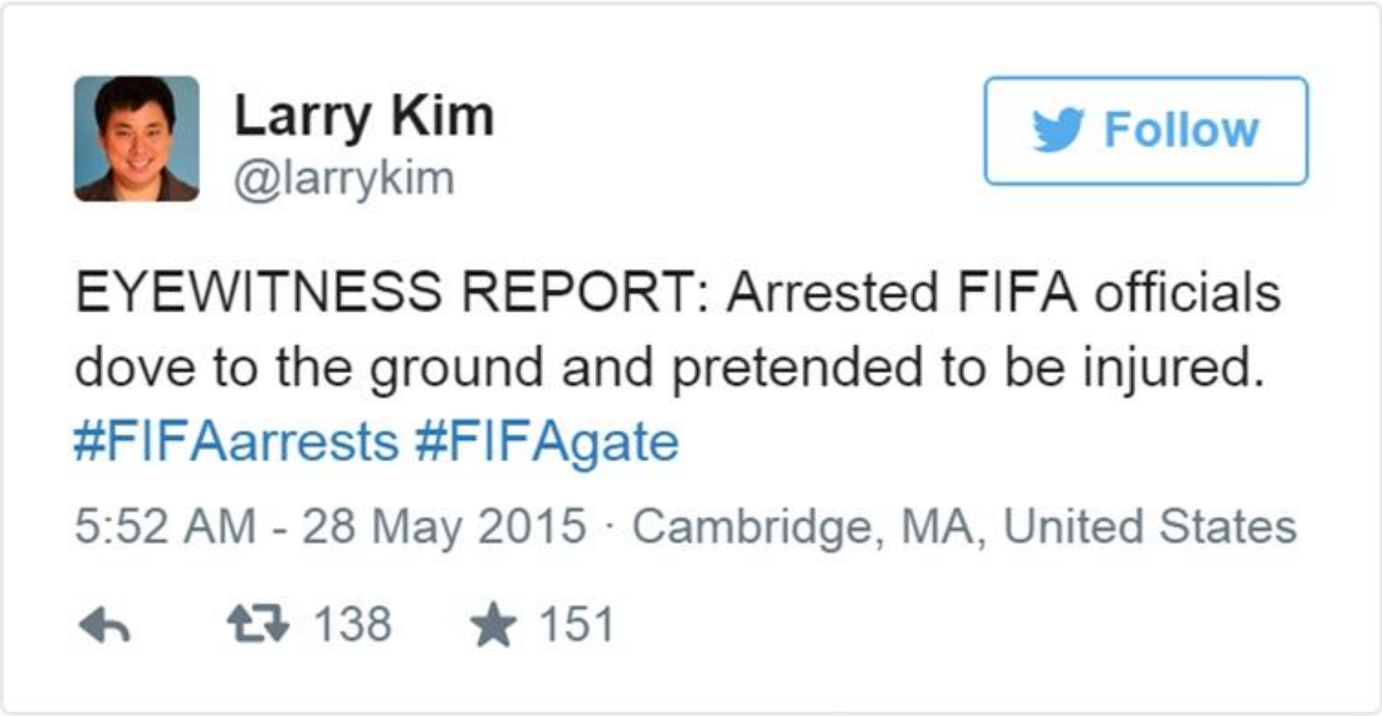
Multi-Label Classification: Applications

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

e.g., Music dataset, audio tracks labelled with different moods, among: {

- amazed-surprised,
- happy-pleased,
- relaxing-calm,
- quiet-still,
- sad-lonely,
- angry-aggressive

Multi-Label Classification: Applications



Multi-Label Classification: Example

- Difference in data sets

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

X_1	X_2	X_3	X_4	X_5	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 $Y \subseteq \{\lambda_1, \ldots, \lambda_L\}$

X_1	X_2	X_3	X_4	X_5	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	?

Multi-Label Classification: Example

- We usually convert labels to binary labels

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

X_1	X_2	X_3	X_4	X_5	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 $[Y_1, \ldots, Y_L] \in 2^L$

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
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	?	?	?	?

Multi-Label Classification

- Solutions
  - **Transformation Based Methods**
    Transform the task to binary/multi-class classifications
  - **Adaptation Based Methods**
    Develop new algorithms to solve the problem

Multi-Label Classification

- Transformation Based Methods
  - Binary Relevance
  - Classifier Chains
  - Label Powerset

Multi-Label Classification

- Binary Relevance
  If there are N labels, we have N binary classifications

$\mathbf{X}$	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

$\Rightarrow$

$\mathbf{X}$	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

- Drawback: it ignores the label depenence

Multi-Label Classification

- Classifier Chains
  - Classifier Chains build the model in a chain by taking label correlations into consideration
  - It uses the feature to perform binary classification on 1^{st} label, the prediction on 1^{st} label will be reused as the features into the 2^{nd} step to predict the 2^{nd} label
  - Repeat the process above until all of the labels are predicted

Multi-Label Classification

- Classifier Chains

$\mathbf{X}$	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

Predict Y1

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

Predict Y2

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

Predict Y3

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

Predict Y4

$\mathbf{X}$	Y_1	Y_3	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

- Use previous prediction results as new features

Multi-Label Classification

- Drawbacks in Classifier Chains
  - Difficult to define the sequence in the chain, though there are some methods (e.g., info gain)
  - If the previous predictions are incorrect, the following predictions may not be right too.

Multi-Label Classification

- Label Powerset
  - Each subset of the label set will be a single label
  - Assign binary classification or multi-class classification to them
  - Find a way to aggregate the results

Multi-Label Classification

- Label Powerset

❶ Transform dataset ...

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

...into a multi-*class* problem, taking 2^L possible values:

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

❷ ...and train any off-the-shelf multi-*class* classifier.

Multi-Label Classification

- Drawbacks in Label Powerset
  - Too many subsets if there are several labels
  - Highly possible to have imbalance issue
  - Overfitting: how to predict new values/labels?

Multi-Label Classification

- Solutions
  - **Transformation Based Methods**
    Transform the task to binary/multi-class classifications
  - **Adaptation Based Methods**
    Develop new algorithms to solve the problem

Algorithm adaptation techniques

- MLkNN. For each test instance:
  - Retrieve the top-k nearest neighbors to each instance
  - Compute the frequency of occurrence of each label
  - Assign a probability to each label and select the labels by using a probability cut-off value

Multi-Label Classification

- Notes
  - Both transformation and adaptation methods are the methods to solve MLC problem
  - They are not classification algorithms
  - For each method, you can use any traditional binary/multi-class classification algorithms to produce the predictions

Evaluation of multilabel learning

- There are multiple labels in the MLC problem
- Traditional evaluation metrics in the classification may not work for MLC
- We need to develop new evaluation metrics

Hamming Loss

Example

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[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]

Consider the misclassification in each bit

$$\textsc{Hamming loss} = \frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}\mathbb{I}[\hat{y}_j^{(i)} \neq y_j^{(i)}] \quad = 4/(4*5)$$

$$= 0.20$$

N = # of labels
L = # of data rows

0/1 Loss

Example

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[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]

Consider the misclassification in the whole label set

$$\text{0/1 LOSS} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{\mathbf{y}}^{(i)} \neq \mathbf{y}^{(i)}) \quad = 3/5$$

$$= 0.60$$

Other Metrics

- JACCARD INDEX – often called multi-label ACCURACY
- RANK LOSS – average fraction of pairs not correctly ordered
- ONE ERROR – if top ranked label is not in set of true labels
- COVERAGE – average “depth” to cover all true labels
- LOG LOSS – i.e., cross entropy
- PRECISION – predicted positive labels that are relevant
- RECALL – relevant labels which were predicted
- PRECISION vs. RECALL curves
- F-MEASURE
  - *micro-averaged* (‘global’ view)
  - *macro-averaged* by label (ordinary averaging of a binary measure, changes in infrequent labels have a big impact)
  - *macro-averaged* by example (one example at a time, average across examples)

Multi-Label Classification Tools

- Mulan
  - Java Based
  - Reuse Weka library
  - No UI
  - http://mulan.sourceforge.net/
- Meka
  - Similar to Weka
  - Java Based
  - With UI
  - http://meka.sourceforge.net/

Multi-Label Classification

- References
  - G Tsoumakas, I Katakis, I Vlahavas, Mining multi-label data
  - G Tsoumakas, I Katakis , Multi-label classification: An overview
  - G Tsoumakas, E Spyromitros-Xioufis, J Vilce, Mulan: A java library for multi-label learning