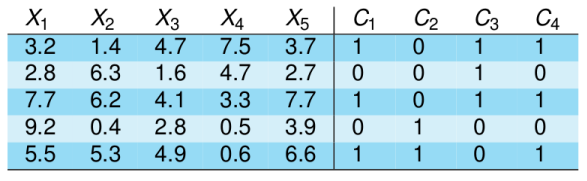
MULTI-LABEL CLASSIFICATION

# Introduction

Multi-label classification is when we have many class variables booleans.

 It is very common in sports journals. In this chase, C1 is a category like “football”, or something like that.

Our variables will be:

***X*** *:* d-dimensional input space

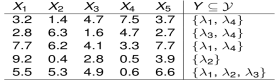
***Y****:* output space of *q* labels {λ1,λ2,...,λq}   
***S***: multi-label training set of *m* samples, *{(****xi,yi****)|1≤i≤m}*

***h****:* multi-label classifier*, h:* ***X*** *2 → Y*

or ranking the associated labels to a sample ***x,***

e.g. *r****x****(*λ2) < *r****x****(*λ4) < *r****x****(*λ1) < *r****x****(*λ3)

Realize that our output space (y) could be a vector. The idea is to associate each output for each input data to a label, i mean, classify that “0 1 0 0 1” (y) is a sport document.

Given a set of initial labels *L*={*λ*1, *λ*2, *λ*3, *λ*4, *λ*5} Given a new instance ***x .*** Multi-label classification outputs a → *bipartition* of the set of labels, i.e. positive and negative ones, *P*x: {*λ*1, *λ*4} and *N*x: {*λ*2, *λ*3, *λ*5} 

## APPLICATIONS

TEXT :***ACM COMPUTING CLASSIFICATION (Veloso et al. 2007)***  A *document* described by its title, abstract, citation, autorship: huge feature space. First hierarchy level, 11 labels: general literature, hardware, software, information systems... Second hierarchy level with 81 labels 81,251 digital archives (only informatics subject, not literature)  
CORPUS (Lewis et al. 2004): 804,414 newswire stories, To be indexed in 103 topic codes   
ENRON COMPANY e-MAILS: UC Berkeley Enron e-mail analysis project, Company-professional e-mails of about 150 Enron senior managers   
BIOLOGY: Annotation of protein functions, Gene ontology annotations (e.g. of a gene)   
IMAGE AND AUDIO: Simultaneous object class recognition, Demographic classification of facial images: sex, age, ethnicity... – Music categorization: instruments, country, rhythm... – Categorization of song emotions: happy, calm, amazed…

BENCHMARK FAMOUS DATASETS: [MULAN](http://mulan.sourceforge.net/) and [MEKA](http://meka).

# EVALUATION METRICS

How are we going to evaluate the algorithms? By example or by labeled?  
 X: d-dimensional input space ***X****=(X1,...Xm)   
 Y :* output space of *q* labels {λ1,λ2,...,λq}, real labels, *Y=*{λ7,λ9}

S : multi-label training set of *m* samples, {*(xi,Yi)|1≤i≤m}  
 h:* multi-label classifier*, h: X 2 → Y*

*Predicted labels for a example, Predh(Y)=*{λ6,λ9,λ11,λ15}

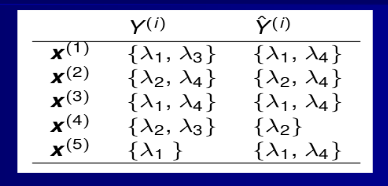
Evaluation metrics:

Example-based:

→ calculated separately **for each sample** and averaged

→ calculated separately **for each label** and averaged

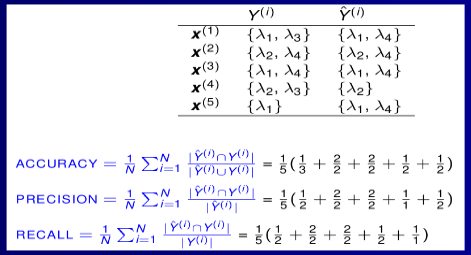
## 0/1 SUBSET ACCURACY : EXACT MATCH - PER SAMPLE

 EXACT MATCH = 1/5 x (0+1+1+0+0) 

 For each sample checks whether the predicted set of labels → is an exact match of the true set of labels

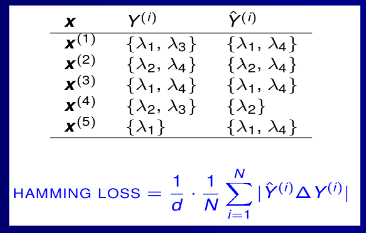
 Very strict evaluation: we only take into account the perfect matches. As x1 has not the same yi and Yi, we don’t take it into account.

## ACCURACY – PER SAMPLE



 Scores from the “Information Retrieval” area →  Using AND and OR logical operations.

## HAMMING LOSS PER SAMPLE

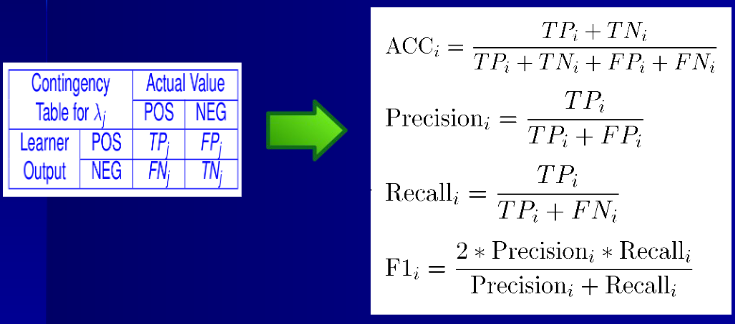


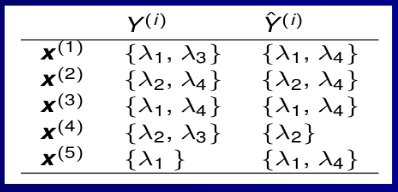
It is based in count the number of errors. This can be implemented with a XOR operation. YOu have to count the differences: in x1, there are 2 differences because lambda3 doesn't appear in Yi and lambda4 doesnt aperas in yi but does in Yi.

 HAMMING LOSS = 1/4 x 1/5 (2+0+0+1+1)

 Symmetric difference between both sets: XOR operation  Average binary classification error

## LABEL-BASED METRICS METRICS PER LABEL





 ACCURACY [LABEL4] = (2+1)/(2+1+2+0) PRECISION [LABEL4] = (2)/(2+2)

 RECALL-SENSITIVITY [LABEL4] = (2)/(2+0)

For lambda4 TP=2 and TN=1 (in x4) and FP=2 and FN=0.

# TECHNIQUES - TAXONOMY

1. Problem transformation methods:
   1. In several single-label tasks
   2. Algorithm independent
2. Algorithm adaptation methods:
   1. Extending supervised algorithms to multi-label data – Decision trees, SVM, Bayesian networks, K-NN..

## BINARY RELEVANCE - BR

## 

We assume that all classes are independents.

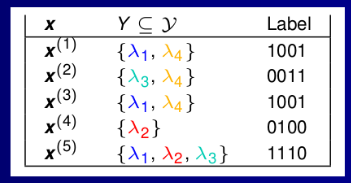
Learning one binary classifier per class

 Output the union of their predictions

 Not consider label relationships

 Ensemble of BR base classifiers common →

## LABEL POWERSET – LC-LP



We assume that every possible combination of labels is a new class. Each set of labels recodify as a different class value →  e.g. “1001”→ classA, “0011” → classB, etc...

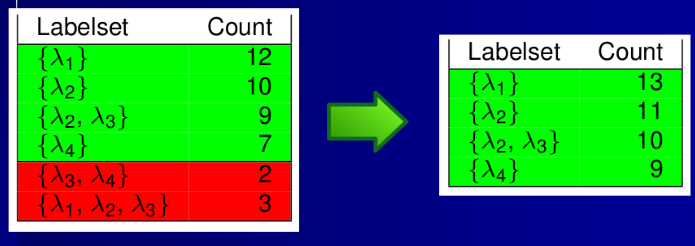


Limited training samples for many new labelsets 

High complexity

Can not predict unseen labelsets

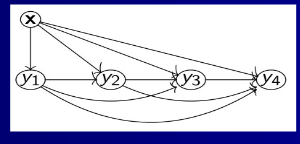
## PRUNED SETS – PS



We delete the uncommon examples but we distribute them in other examples. How do we distribute that? Anybody knows.

Start considering all labelsets too many!! Reduce labelsets → →  Prune examples belonging to less frequent classes (e.g. < 7). Distribute pruned examples along more frequent subsets of their labelset →  Reduce the number of labelsets and focus on frequent ones. Train a label Powerset multi-label classifier

## CLASSIFIER CHAINS - CC



 Given a dataset with |L| labels: y1,y2,...,y|L|

Dataset is transformed in |L| datasets where instances in the “j” dataset are of the form:

((xi,y1,y2,...,yj-1), yj)

Classifiers build a CHAIN → each learns a binary classification of a single label

Features in each classifier → EXTENDED with binary labels indicating the prediction of previous labels-classifiers in the chain

Partial label dependence is maintained, but... order of the chain?

The idea is, if you have n classes, make n classifiers where the first only knows the first class. The second classifiers knows the prediction of the first classifiers and the second class. And so on.

# ALGORITHM ADAPTATION

# Extending supervised algorithms to deal with multi-label data  Literature shows plenty of examples – Just a couple: – K-NN: assigns to ***x*** the most common labels of its K neighbours Decision trees: extending the concept of multi-label entropy. Multiple labels at leaves

Hot topic – specially in NLP

Closely related with “tagging”-”annotation”, news’ categories, web 2.0, multiple outputs, learning from crowds, recommender systems...

Many real world applications

# TOOLS

MULAN: WEKA-based library

MEKA: WEKA-based framework and GUI

Utiml, mldr, mldr.datasets R packages

Datasets' repositories:

http://mulan.sourceforge.net/datasets.html

http://meka.sourceforge.net/#datasets

http://www.uco.es/kdis/mllresources/

Talks-Tutorials:   
C. Bielza, P. Larrañaga, UPM-Madrid [link]   
J. Read, MEKA's programmer [link]   
G. Tsoumakas et al., tutorial [link]

Review:

M-L. Zhang, Z-H. Zhou (2013). “A review on multi-label learning algorithms”. IEEE Transactions on Knowledge and Data Engineering, 26(8), 1819-1837

F. Herrera, F. Charte, A.J. Rivera, M.J. Del Jesús (2016). Multi label Classification. Springer