Seminar aus Data Mining und Maschinellem Lernen

Multilabel text classification for automated tag suggestion

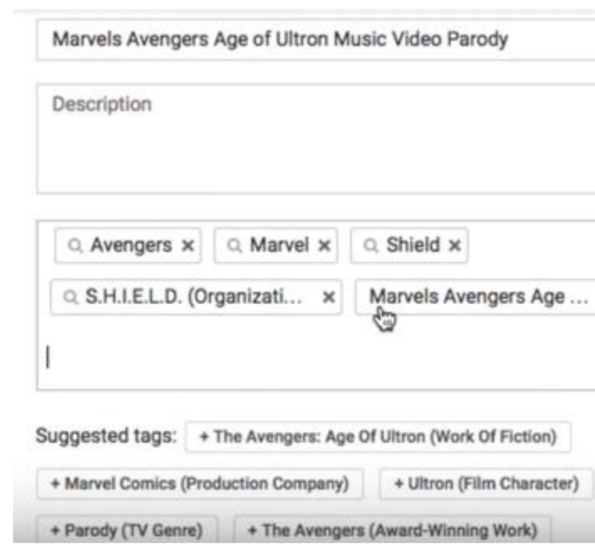
Adaptive selection of base classifiers in one-against-all learining for large multi-lableled collections

Hong Linh Thai

Motivation

- If you think of Web 2.0, e.g. tagging of images on Flickr or videos on You Tube, it would be really convenient if this process could be automated

Emanuele Toscano, CC BY-NC-ND- 2.0



https://www.youtube.com/watch?v=9B1-u4_XvX0

Motivation

- But tagging is not only useful for images, it can be also applied to text
- Examples:
  - Automated text categorization system for high energy physics papers, 2802 abstracts, 1093 keywords [1]
  - ECML/PKDD 2008 Discovery Challenge: Tag Recommendation in Social Bookmark System, User can tag bookmarks and BibTeX entries, $815{,}000$ Tags, $400{,}000$ bookmarks and BibTeX entries [2,3]
  - ECML/PKDD 2012 Discovery Challenge: Large scale hierarchical classification with data from Wikipedia, $36{,}500 \sim 325{,}000$ Categories, $380{,}000 \sim 2{,}400{,}000$ Documents [4]

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- But tagging is not only useful for images, it can be also applied to t
- Exa
  - A p
  - E e a W ents

Problem:

- Large data sets
- Large number of possible labels
- One object x can have multiple labels
- Want to find an automatic way to solve this problem

This calls for extreme classification / multi-label classification

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

I. Motivation
II. Formal Definition – Multi-Label Classification
III. Problems / Key Challenges
IV. Approaches / Solutions
   a) Adaptive Selection of Base Classifiers in One-Against-All Learning for Large Multi-Lableled Collections, Ráez et al. [1]
   b) Multilabel Text Classification for Automated Tag Suggestion, Katakis et al. [2]

Formal Definition – Multi-Label Classification

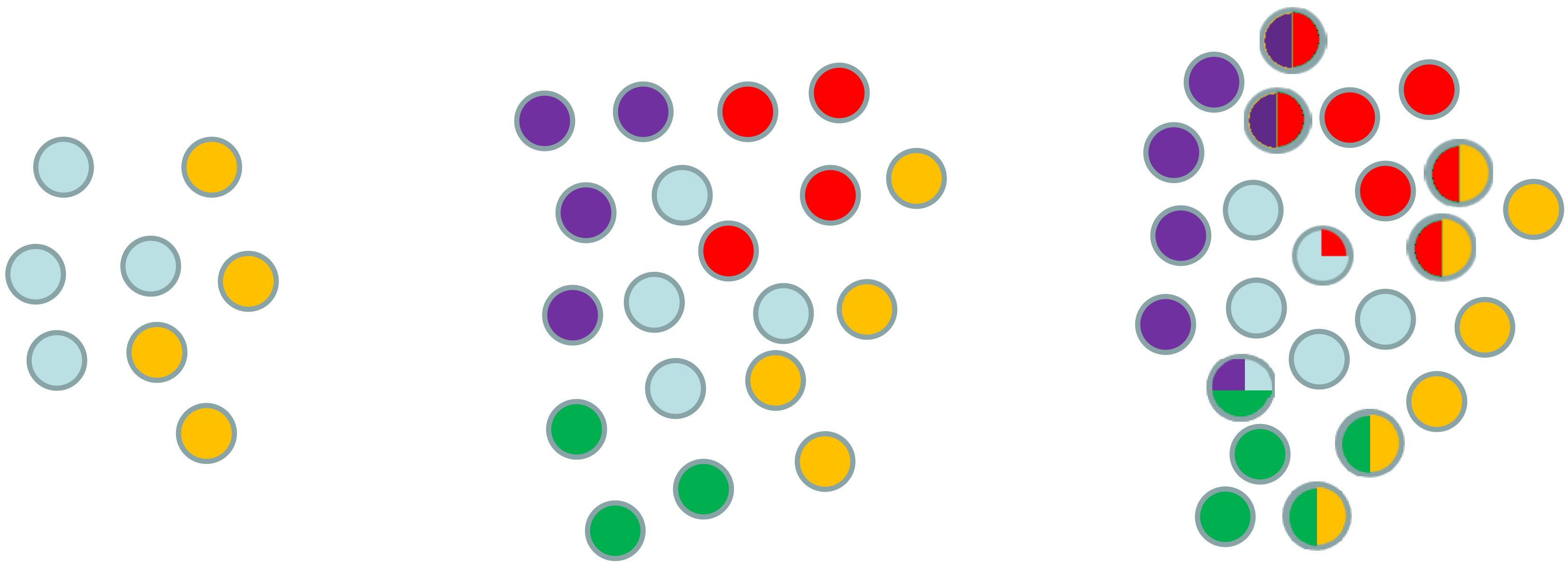
- Setting: Let $\mathcal{X} = \mathbb{R}^d$ be an d-dimensional instance space / feature space and $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ be the label space where k is the number of possible labels
- Goal: Given a Training Set, $S = (\boldsymbol{x_i}, \boldsymbol{Y_i}), 1 \leq i \leq n$ consisting of n training instances, where $\boldsymbol{x_i} \in \mathcal{X}$ and $\boldsymbol{Y_i} \subseteq \mathcal{L}$, to learn a function $f : \mathcal{X} \longrightarrow 2^{\mathcal{L}}$, which can predict the labels for any unseen instance

i	F_1	…	F_d	$\boldsymbol{Y_i}$
1	0.41	…	1	$\{\lambda_1, \lambda_2\}$
2	0.1	…	0	$\{\lambda_3\}$
3	0.75	…	1	$\{\lambda_1, \lambda_4, \lambda_k\}$
…	…	…	…	…

Example for a Training Set

Difference to Binary and Multi-Class Classification

- Binary Classification: Only 2 classes
- Multi-Class Classification: One instance can have only one class



Binary Classification   Multi-Class Classification   Multi-Label Classification

Multi-Label Classification Methods

TECHNISCHE UNIVERSITÄT DARMSTADT

According to Tsoumakas there are 2 groups of classification methods [2,5] :

- Algorithm Adaptation Methods
  - Extend known learining algorithms to handle multi-label data directly
  - Advantage: If we can do so, we can solve the problem in one large optimization problem
  - Disadvantage: It is difficult to extend an algorithm
- Problem Transformation Methods
  - Transform multi-label problem into one or mutliple single-label classification problems
  - Advantage: Single-label problems are well known
  - Disadvantage: Depending on the transformation we can lose information

Binary Relevance (BR)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Assumption: Prediction of each label as an independent binary classification task
- Learn one binary classifier for each different label $f_j : \mathcal{X} \longrightarrow \{\lambda_j, \neg\lambda_j\}$
  - Positive examples are the ones for which the label is positive
  - Negative examples are the remaining ones
- Transform data set into multiple ones:

$$
\begin{array}{ccl}
 & \xrightarrow{\lambda_1} & S_1 = (\boldsymbol{x_i}, \boldsymbol{\phi}(\boldsymbol{Y_i}, \lambda_1)), 1 \leq i \leq n \\
 & & \vdots \\
S = (\boldsymbol{x_i}, \boldsymbol{Y_i}), 1 \leq i \leq n & \xrightarrow{\lambda_j} & S_j = \left(\boldsymbol{x_i}, \boldsymbol{\phi}(\boldsymbol{Y_i}, \lambda_j)\right), 1 \leq i \leq n \\
 & & \vdots \\
 & \xrightarrow{\lambda_k} & S_m = (\boldsymbol{x_i}, \boldsymbol{\phi}(\boldsymbol{Y_i}, \lambda_k)), 1 \leq i \leq n
\end{array}
$$

Binary Relevance (BR)

Example:

i	F_1	…	F_d	Y_i
1	0.41	…	1	$\{\lambda_1, \lambda_2\}$
2	0.1	…	0	$\{\lambda_3\}$
3	0.75	…	1	$\{\lambda_1, \lambda_4, \lambda_n\}$
…	…	…	…	…

λ_1

i	F_1	…	F_d	λ_1
1	0.41	…	1	1
2	0.1	…	0	0
3	0.75	…	1	1
…	…	…	…	…

$f_1 : \mathcal{X} \longrightarrow \{\lambda_1, \neg\lambda_1\}$

λ_2

i	F_1	…	F_d	λ_2
1	0.41	…	1	1
2	0.1	…	0	0
3	0.75	…	1	0
…	…	…	…	…

$f_2 : \mathcal{X} \longrightarrow \{\lambda_2, \neg\lambda_2\}$

… …

Problem Transform Methods

TECHNISCHE UNIVERSITÄT DARMSTADT

According to Sorower and Zhang et al. there are 3 types [6,7] :

- Binary Relevance (BR) / First Order Strategy
  - Solve the problem label by label,
  - Transform the problem to k one vs all problems
- Ranking by Pairwise Comparison / Second Order Strategy
  - Solve the problem pairwise, voting for prediction
  - Transform the problem to $k(\mathrm{k}-1)/2$ one vs one problems
- Label Powerset / High Order Strategy
  - Create one new label for each label set
  - One large single label multi-class problem

Problems / Key Challenges

- High input / output dimension
  - Number of label sets grow exponentially, e.g. 10 label classes leads to 2^{10} possible label sets
- Highly imbalanced data
  - Also known as Class Imbalance Problem
- Very few data points per labels
- Inter-dependency of class labels
  - E.g. if something is labeled with *Frodo* and *Gandalf* the label *Lord of the Rings* will be much likely than *Titanic*
- Real time constraints

Class Imbalance Problem

- Usually multi-labeled data have an unequal distribution of classes
  - Imbalance between positive and negative samples (inner imbalance degree)
  - More frequent and less frequent classes (inter-class imbalance degree)

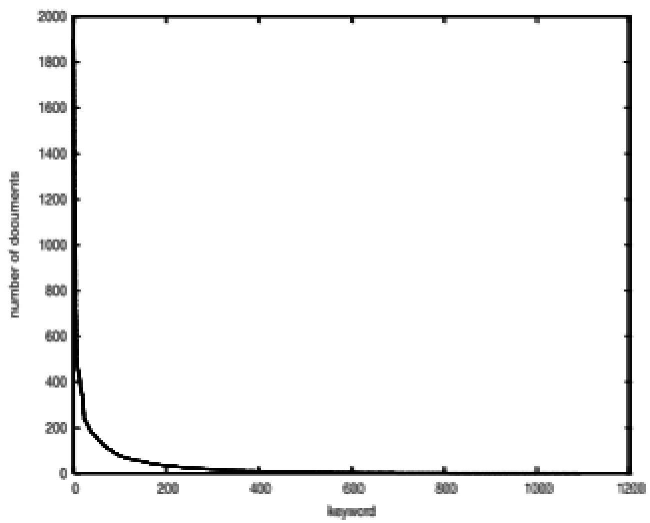
Class Imbalance Problem

Example: HEP Collection

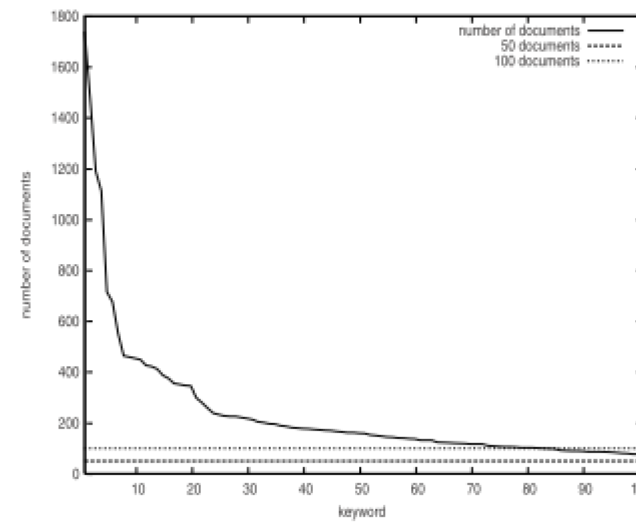
- high energy physics papers, 2802 abstracts, 1093 keywords
- Inter-class imbalance:

No. docs.	Keyword
1898 (67%)	electron positron
1739 (62%)	experimental results
1478 (52%)	magnetic detector
1190 (42%)	quark
1113 (39%)	talk
715 (25%)	Z0
676 (24%)	anti-p p
551 (19%)	neutrino
463 (16%)	W
458 (16%)	jet

10 most frequent keywords



all keywords



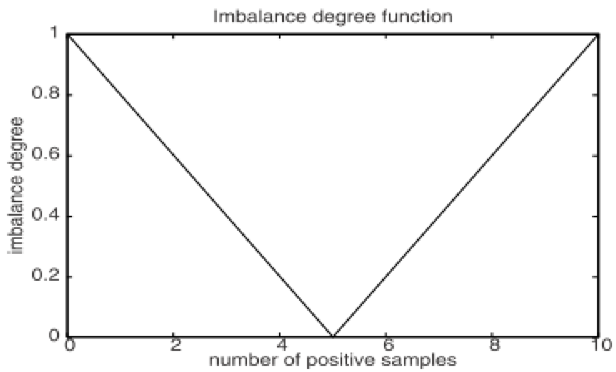
100 most frequent keywords

Class Imbalance Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Example: HEP Collection

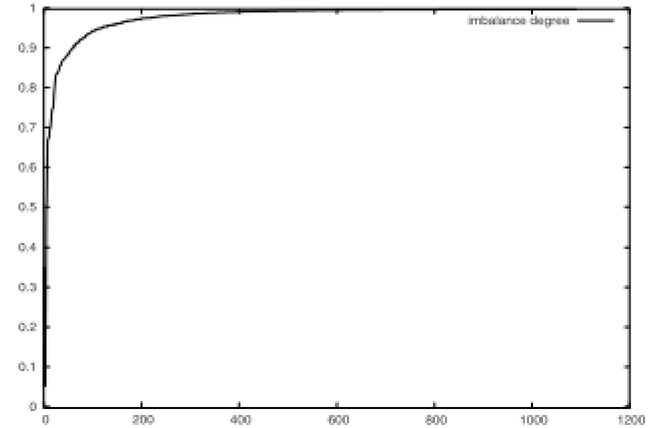
- high energy physics papers, 2802 abstracts, 1093 keywords
- Inner imbalance:



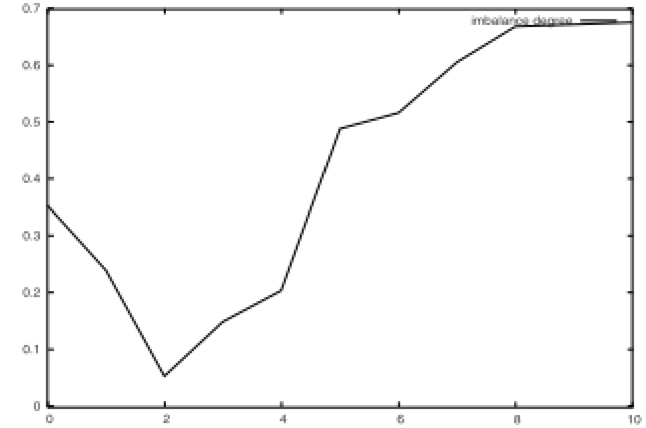
$$i_i = \left| \frac{1 - 2n_i}{n} \right|$$

n: number of samples

n_i: number of samples with label i



all keywords



10 most frequent keywords

Solution for the Class Imbalance Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Ráez et al. proposed :

- Filter out the rare classes using a parameter α
  - If a classifier performs worse than α the classifier and the whole class is discarded

- Overweighting of the positive samples
  - $w_{+} = C_{-} / C_{+}$

  C_{-}: number of neg. samples

  C_{+}: number of pos. samples

Evaluation Measures

- Precision: $Prec(C_i) = \frac{tp}{tp + fp}$
  - Ratio correct preditced c_i to total predicted c_i
- Recall: $\mathrm{Rec}(C_i) = \frac{tp}{tp + fn}$
  - Ratio correct predicted c_i to total existing c_i
- F_1: $F_1(C_i) = 2\frac{Prec(C_i)\ \cdot Rec(C_i)}{Prec(C_i) + Rec(C_i)}$
  - Harmonic mean

Category c_i		Expert judgments YES	Expert judgments NO
Classifier Judgments	YES	TP_i	FP_i
	NO	FN_i	TN_i

[8]

- Macroaveraging
  - First evaluate „locally" then average „globally"
- Microaveraging
  - Use „global" confusion matrix

Category set $\mathcal{C} = \{c_1, \ldots, c_{\lvert\mathcal{C}\rvert}\}$		Expert judgments YES	Expert judgments NO
Classifier Judgments	YES	$TP = \sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i$	$FP = \sum_{i=1}^{\lvert\mathcal{C}\rvert} FP_i$
	NO	$FN = \sum_{i=1}^{\lvert\mathcal{C}\rvert} FN_i$	$TN = \sum_{i=1}^{\lvert\mathcal{C}\rvert} TN_i$

[8]

Macroaveraging vs Microaveraging

Label	Tp	Fp	Fn	Tn	precision	recall
C1	10	10	10	800	0.5	0.5
C2	90	10	10	750	0.9	0.9
Global	100	20	20	1550		

$$Prec(C_i) = \frac{tp}{tp + fp}$$

$$Rec(C_i) = \frac{tp}{tp + fn}$$

Macroaveraging: $Prec^M(C) = \frac{0.5+0.9}{2} = 0.7$

- Meassure of effectiveness on small classes
- All classes have same weight even smaller ones

Microaveraging: $Prec^\mu(C) = \frac{100}{100+20} = 0.83$

- Meassure of efffectiveness on large classes

Tag Recommendation in Social Bookmark System, Katakis et al.

TECHNISCHE UNIVERSITÄT DARMSTADT

- Task: Tag Recommendation in Social Bookmark System, $815{,}000$ Tags, $400{,}000$ bookmarks and BibTeX entries
- Idea: Learn a personalized tag recommender, if item and user exist in training set just return, else predict
- Method: Naive Bayes BR Classifier from the Mulan [9] package
- Results:

Parameters			F-measure		
θ	M	N	All	Book	Bib
0.0	10	10	0.0716	0.0782	0.0633
0.0	5	5	0.0848	0.0940	0.0736
0.0	1	1	0.0700	0.0904	0.0453
0.9	10	10	0.0713	0.0752	0.066
0.9	3	3	0.0847	0.0940	0.0734
0.9	10	3	**0.0852**	**0.0942**	**0.0740**

Measure based on macroaveraging

θ: Confidence
M: Number of recommendations
N: Number of most popular tags

Automated text categorization system Ráez et al.

- Task : Automated text categorization system for high energy physics papers, 2802 abstracts, 1093 keywords
- Idea: Filter out not frequent labels to improve classification
- Method: BR Learning: SVM with over-weighting and filtering out non frequent classes + Scut to train the parameter of the SVM

Experiment	Precision	Recall	F1	Accuracy	Error	% of classes covered
No weight	74.07	33.96	43.92	98.23	1.77	33.96
No weight / Scut	**74.26**	34.44	44.38	98.24	1.76	99.95
Overweight 20	51.47	**45.84**	46.50	97.71	2.29	57.32
Auto weight	58.10	44.39	48.09	97.94	2.06	58.09
Overw. 2,5,10,20 / Scut	71.74	39.92	48.47	**98.25**	**1.75**	**100.00**
Auto weight / Scut	58.03	45.30	48.56	97.89	2.11	99.82
Overweight 2	70.74	40.45	48.78	98.21	1.79	53.36
Overweight 5	64.56	43.57	49.40	98.11	1.89	57.19
Overweight 10	62.30	45.22	50.14	98.08	1.92	57.30
Overw. 2,5,10,20	65.89	44.59	**50.53**	98.17	1.83	57.53

SVM with $\alpha = 0.0$, no filtering

Measure based on macroaveraging on documents

Automated text categorization system Ráez et al.

Results:

- Recall is lower than prediction due to rare classes
- Discarding bad classifiers improves precision, while recall does not get much worse

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Precision	65.89	70.04	70.41	70.88	71.90	71.96	71.02	67.96
Recall	44.59	44.49	43.95	42.95	40.54	36.65	31.80	23.02
F_1	50.53	**51.59**	51.32	50.77	49.21	46.11	41.70	32.83
Accuracy	98.17	98.25	98.25	98.25	98.24	98.21	98.15	98.03
Error	1.83	1.75	1.75	1.75	1.76	1.79	1.85	1.97
% classes trained	57.53	56.49	50.81	43.20	32.73	23.23	16.00	8.58

Handcrafted multiweighted SVM with filtering

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Precision	58.03	62.47	64.84	67.45	69.47	71.19	71.14	68.24
Recall	45.30	45.04	44.83	44.24	42.76	39.59	34.43	24.88
F_1	48.56	49.93	50.47	50.75	50.27	48.37	44.10	34.76
Accuracy	97.89	98.06	98.14	98.20	98.23	98.22	98.17	98.05
Error	2.11	1.94	1.86	1.80	1.77	1.78	1.83	1.95
% classes trained	99.82	85.30	77.10	68.47	55.74	42.34	30.82	16.72

Auto-weighted S-cut tresholded SVM with filtering

Measure based on macroaveraging on documents

Conclusion

- Binary Relevance learning approach is very simple, thus it does not performs quite very well on real data
- Reasons:
  - Class imbalance problem
  - Does not include any information about label dependencies
- Filtering can help to some extend, but ideally we want to be able to predict even rare labels
- Way how the measure is computed is very important for interpreting the results

→ **Nessesity of better methods**

Summary

- Introduced Multi-Label Classification
  - Showed 2 common approaches:
    - Algorithm Adaptation Methods : Fit algorithm to data
    - Problem Transformation Methods : Fit data to algorithm
  - Naive approach : Binary Relevance
  - Class Imbalance Problem
  - Basic evaluation measures, microaveraging vs macroaveraging
  - Results of BR Learning and filtering not frequent labels

References

[1] - Ráez, A. M., López, L. A. U., & Steinberger, R. (2004). Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In *Advances in Natural Language Processing* (pp. 1-12). Springer Berlin Heidelberg.

[2] - Katakis, Ioannis, Grigorios Tsoumakas, and Ioannis Vlahavas. "Multilabel text classification for automated tag suggestion." *ECML PKDD discovery challenge,* 75 (2008).

[3] - ECML PKDD Discovery Challenge 2008 http://www.kde.cs.uni-kassel.de/ws/rsdc08

[4] - Large Scale Hiearachical Text Classification Challenge, EMCL PKDD *discovery challenge 2012* http://lshtc.iit.demokritos.gr/LSHTC3_CALL

[5] - Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*(2006).

[6] - Sorower, Mohammad S. "A literature survey on algorithms for multi-label learning." *Oregon State University, Corvallis* (2010).

[7] - Zhang, Min-Ling, and Zhi-Hua Zhou. "A review on multi-label learning algorithms." *Knowledge and Data Engineering, IEEE Transactions on* 26.8 (2014): 1819-1837.

[8] - Sebastiani, Fabrizio. "Machine learning in automated text categorization." *ACM computing surveys (CSUR)* 34.1 (2002): 1-47.

[9] – Mulan - Multi Label Classification Java Library, http://mlkd.csd.auth.gr/multilabel.html

[10] – Loza Meneciá, Eneldo and Fürnkranz, Johannes "Tutorial on Multilabel Classification" http://www.ke.tu-darmstadt.de/staff/eneldo/MultilabelTutorial.pdf