
CSE 6339 – Introduction to Computational
Linguistics

Text Classification

Presenter: Nadine Dulisch

1. Text Classification – Definition
2. Representing text for classification
3. Text classification methods
4. Resources

- Also known as Text Categorization
- The classifier:
 - *Input*: a document x
 - *Output*: a predicted class y from some fixed set of labels y_1, \dots, y_K
- The learner:
 - *Input*: a set of m hand-labeled documents $(x_1, y_1), \dots, (x_m, y_m)$
 - *Output*: a learned classifier $f: x \rightarrow y$

Examples:

- Classify news stories as *World, US, Business, SciTech, Sports, Entertainment, Health, Other*
- Add MeSH terms to Medline abstracts e.g. “Conscious Sedation” [E03.250]
- Classify business names by industry.
- Classify student essays as *A, B, C, D, or F*.
- Classify email as *Spam, Other*.
- Classify email to tech staff as *Mac, Windows, ..., Other*.
- Classify pdf files as *ResearchPaper, Other*
- Classify documents as *WrittenByReagan, GhostWritten*
- Classify movie reviews as *Favorable, Unfavorable, Neutral*.
- Classify technical papers as *Interesting, Uninteresting*.
- Classify jokes as *Funny, NotFunny*.
- Classify web sites of companies by Standard Industrial Classification (SIC) code.

Examples:

- Best-studied benchmark: *Reuters-21578* newswire stories
 - 9603 train, 3299 test documents, 80-100 words each, 93 classes

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS

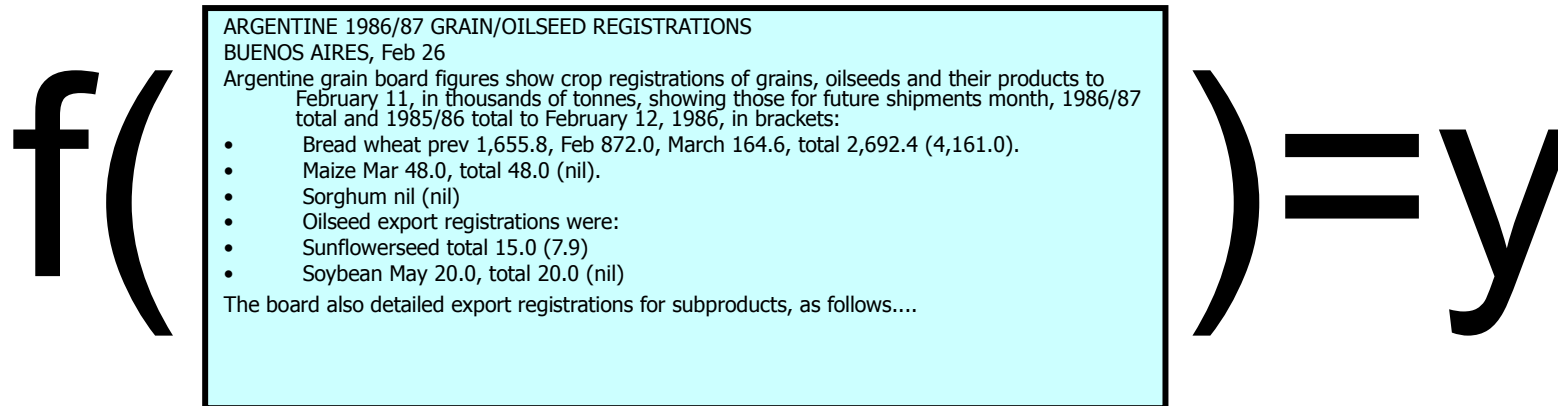
BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
 - Sunflowerseed total 15.0 (7.9)
 - Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

➔ Categories: **grain, wheat** (of 93 binary choices)



?

simplest useful

What is the ~~best~~ representation for the document x being classified?

Representing text: a list of words

f (
 ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
 BUENOS AIRES, Feb 26
 Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:
 • Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
 • Maize Mar 48.0, total 48.0 (nil).
 • Sorghum nil (nil)
 • Oilseed export registrations were:
 • Sunflowerseed total 15.0 (7.9)
 • Soybean May 20.0, total 20.0 (nil)
 The board also detailed export registrations for subproducts, as follows....
) = y

f (
 (argentine, 1986, 1987, grain, oilseed,
 registration█, buenos, aires, feb, 26,
 argentine, grain, board, figures█, show, crop,
 registrations, ~~of~~, grains, oilseeds█, ~~and~~, ~~their~~,
 products, ~~to~~, february, 11, ~~in~~, ...
) = y

Common refinements: **remove stopwords**, **stemming**, collapsing multiple occurrences of words into one....

Representing text: a bag of words

ARGENTINE 1986/87 **GRAIN/OILSEED** REGISTRATIONS
 BUENOS AIRES, Feb 26
 Argentine **grain** board figures show crop registrations of **grains**, **oilseeds** and their products to February 11, in thousands of **tonnes**, showing those for future **shipments** month, 1986/87 **total** and 1985/86 **total** to February 12, 1986, in brackets:

- Bread **wheat** prev 1,655.8, Feb 872.0, March 164.6, **total** 2,692.4 (4,161.0).
- **Maize** Mar 48.0, total 48.0 (nil).
- **Sorghum** nil (nil)
- **Oilseed** export registrations were:
- **Sunflowerseed** total 15.0 (7.9)
- **Soybean** May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

<i>word</i>	<i>freq</i>
grain(s)	3
oilseed(s)	2
total	3
wheat	1
maize	1
soybean	1
tonnes	1
...	...

If the order of words doesn't matter, \mathbf{x} can be a *vector* of word *frequencies*.

Categories: **grain, wheat**

“Bag of words”: a long sparse vector $\mathbf{x}=(, \dots, f_i, \dots)$ where f_i is the frequency of the i -th word in the vocabulary

- Various methods and techniques for text classification, such as:
 - Naive Bayes classifier
 - Support vector machines (SVM)
 - String Kernels
 - Expectation maximization (EM)
 - Tf-idf
 - Latent semantic indexing
 - Artificial neural network
 - K-nearest neighbour algorithms
 - Decision trees such as ID3 or C4.5

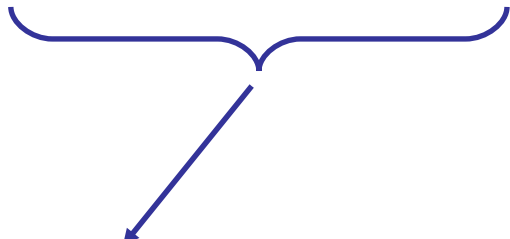
- Represent document x as list of words w_1, w_2, \dots
- For each y , build a probabilistic model $\Pr(X|Y=y)$ of “documents” in class y
 - $\Pr(X=\{argentine, grain\dots\}|Y=wheat) = \dots$
 - $\Pr(X=\{stocks, rose, in, heavy, \dots\}|Y=nonWheat) = \dots$
- To classify, find the y which was most likely to *generate* x —*i.e.*, which gives x the best score according to $\Pr(x|y)$
 - $f(x) = \operatorname{argmax}_y \Pr(x|y) * \Pr(y)$

- How to estimate $\Pr(X|Y)$?
- *Simplest useful* process to generate a bag of words:
 - pick word 1 according to $\Pr(W|Y)$
 - repeat for word 2, 3,
 - each word is generated *independently* of the others (which is clearly not true) but means

$$\Pr(w_1, \dots, w_n | Y = y) = \prod_{i=1}^n \Pr(w_i | Y = y)$$

How to estimate $\Pr(W|Y)$?

- How to estimate $\Pr(X|Y)$?

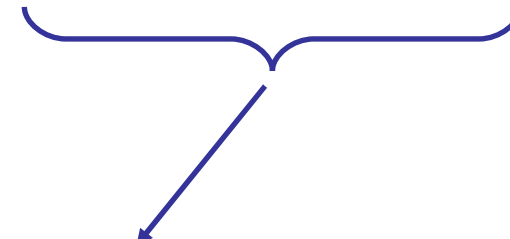
$$\Pr(w_1, \dots, w_n | Y = y) = \prod_{i=1}^n \Pr(w_i | Y = y)$$


Estimate $\Pr(w|y)$ by looking at the data...

$$\Pr(W = w | Y = y) = \frac{\text{count}(W = w \text{ and } Y = y)}{\text{count}(Y = y)}$$

This gives score of zero if x contains a brand-new word w_{new}

- How to estimate $\Pr(X|Y)$?

$$\Pr(w_1, \dots, w_n | Y = y) = \prod_{i=1}^n \Pr(w_i | Y = y)$$


... and also **imagine** m
examples with $\Pr(w|y)=p$

$$\Pr(W = w | Y = y) = \frac{\text{count}(W = w \text{ and } Y = y) + mp}{\text{count}(Y = y) + m}$$

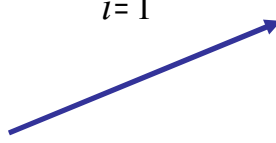
Terms:

- This $\Pr(W|Y)$ is a *multinomial distribution*
- This use of m and p is a *Dirichlet prior* for the multinomial

- Putting this together:
 - for each document x_i with label y_i
 - for each word w_{ij} in x_i
 - $\text{count}[w_{ij}][y_i]++$
 - $\text{count}[y_i]++$
 - $\text{count}++$
 - to classify a new $x=w_1\dots w_n$, pick y with top *score*:

$$\text{score}(y, w_1\dots w_k) = \lg \frac{\text{count}[y]}{\text{count}} + \sum_{i=1}^n \lg \frac{\text{count}[w_i][y] + 0.5}{\text{count}[y] + 1}$$

key point: we only need counts for words that actually appear in x



- **Pros:**
 - Very fast and easy-to-implement
 - Well-understood formally & experimentally
 - see “Naive (Bayes) at Forty”, Lewis, ECML98
- **Cons:**
 - Seldom gives the very best performance
 - “Probabilities” $Pr(y/x)$ are not accurate
 - e.g., $Pr(y|x)$ decreases with length of x
 - Probabilities tend to be close to zero or one

Reminder:

Representing text: a bag of words

word *freq*

ARGENTINE 1986/87 **GRAIN/OILSEED** REGISTRATIONS
 BUENOS AIRES, Feb 26
 Argentine **grain** board figures show crop registrations of **grains**,
oilseeds and their products to February 11, in thousands of
tonnes, showing those for future **shipments** month,
 1986/87 **total** and 1985/86 **total** to February 12, 1986, in
 brackets:

- Bread **wheat** prev 1,655.8, Feb 872.0, March 164.6, **total** 2,692.4 (4,161.0).
- **Maize** Mar 48.0, total 48.0 (nil).
- **Sorghum** nil (nil)
- **Oilseed** export registrations were:
- **Sunflowerseed** total 15.0 (7.9)
- **Soybean** May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

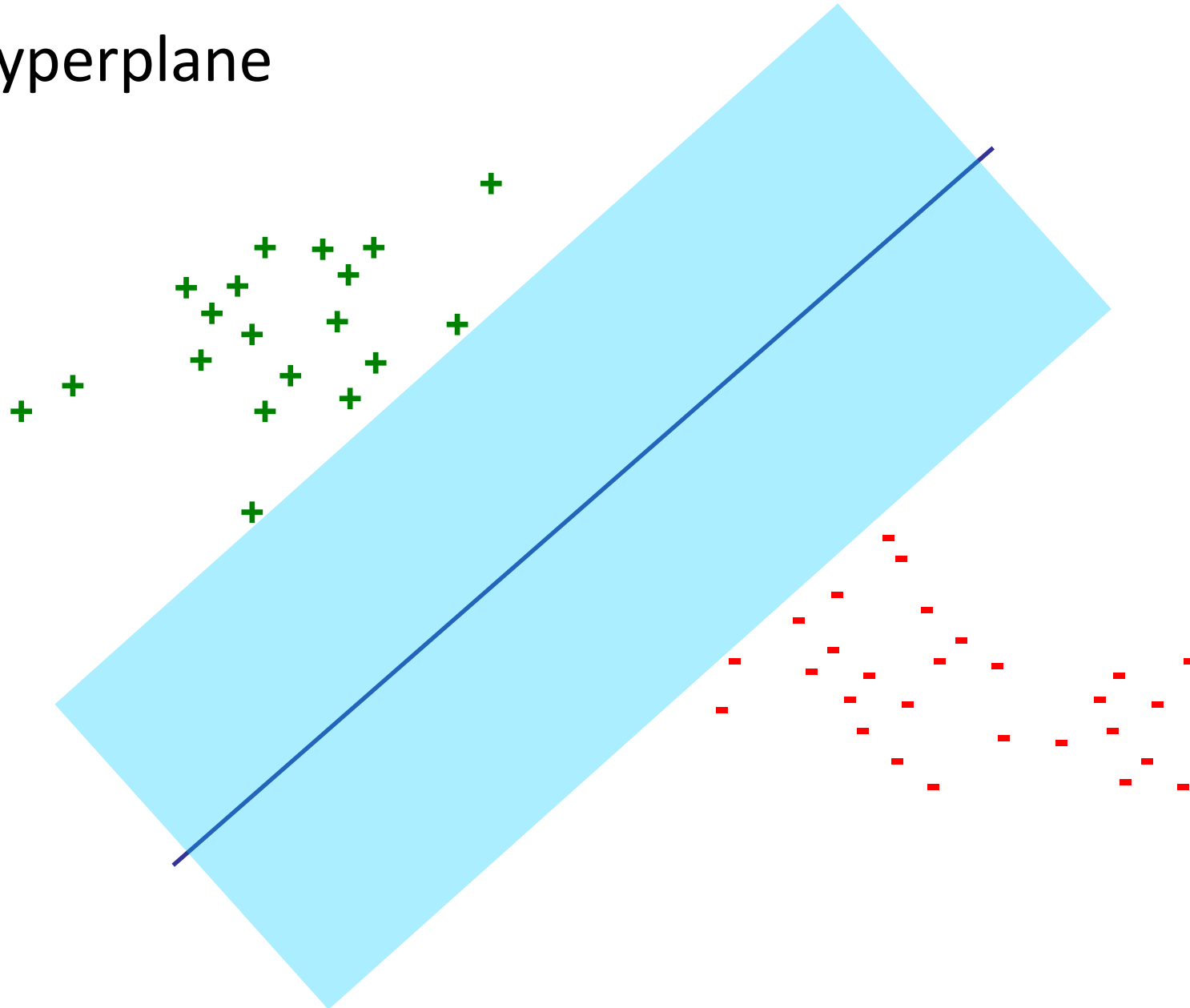
grain(s)	3
oilseed(s)	2
total	3
wheat	1
maize	1
soybean	1
tonnes	1
...	...

If the order of words doesn't matter, \mathbf{x} can be a *vector of word frequencies*.

Categories: **grain, wheat**

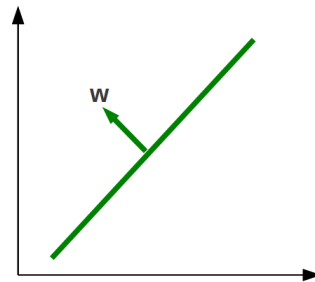
“Bag of words”: a long sparse vector $\mathbf{x}=(, \dots, f_i, \dots)$ where f_i is the frequency of the i -th word in the vocabulary

- Hyperplane



- Hyperplane

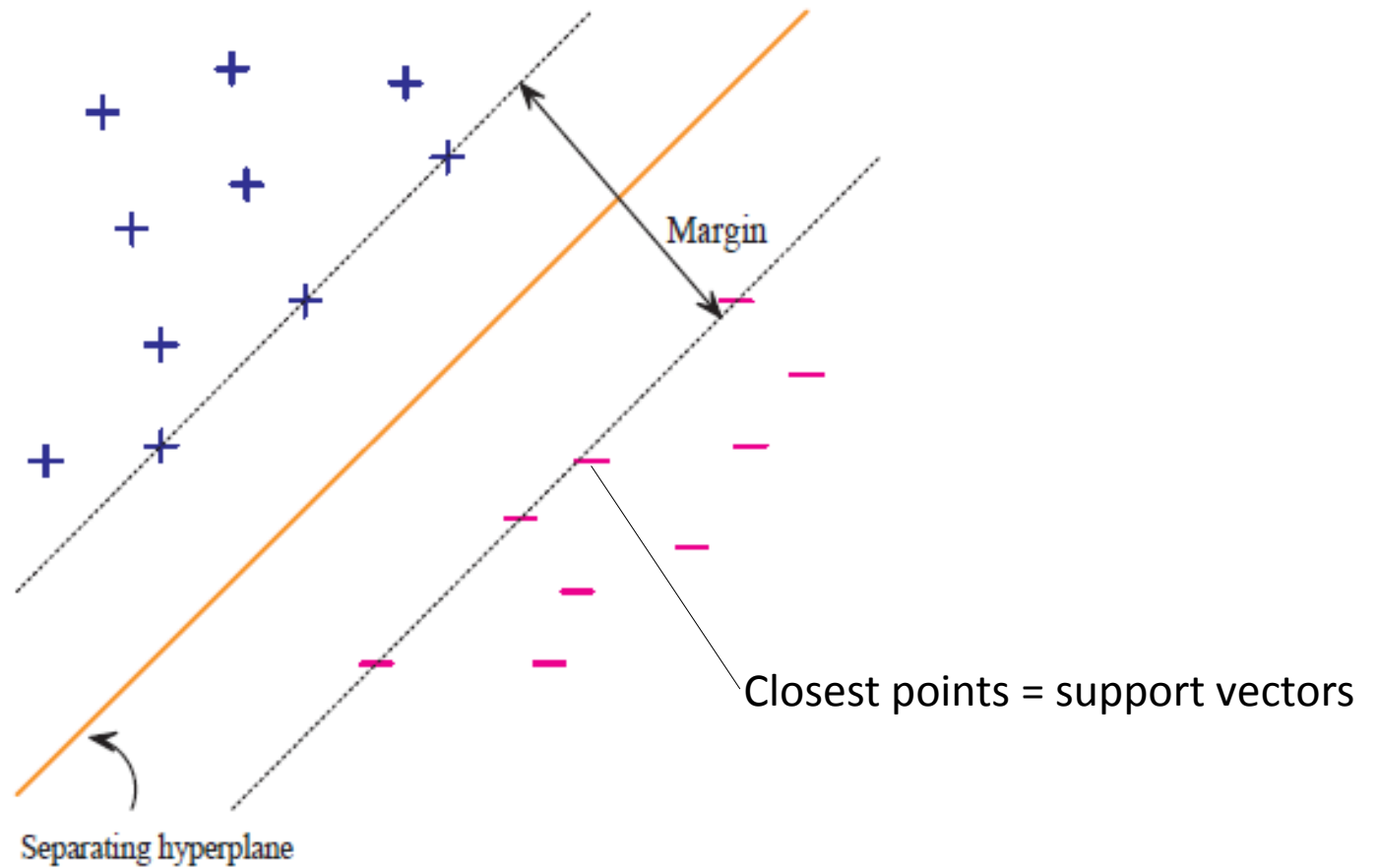
- Separates n-dimensional space into 2 half spaces



- All points on one side of the hyperplane are classified as positive and the ones on the other side as negative
- Defined by an outward pointing normal vector \mathbf{w} (weight)
- \mathbf{w} is orthogonal to any vector lying on the hyperplane
- Assumption: The hyperplane passes through origin
 - have a bias term b ; we will then need both \mathbf{w} and \mathbf{b} to define it
 - $b \neq 0$ means moving it parallel along \mathbf{w}

- Hyperplane
 - n dimensions $\rightarrow n = \#$ of classes
 - w = vector of weights
 - b = bias
 - x = vector of frequencies (feature vector)
 - Classification:
 - $y = \text{sign}(w * x + b)$
 - $wx + b > 0 \Rightarrow y = +1$
 - $wx + b < 0 \Rightarrow y = -1$

- Margin-based Learning



- $y = \text{sign}(w * x + b)$
- Goal:
 - To learn the hyperplane (w, b) by using the training data
- If such a hyperplane can be found, the data is **linearly separable**
- What constitutes the best classifier?
 - One that correctly classifies all training examples while maximizing the distance from the nearest example to the hyperplane
 - One that allows one or more examples to be misclassified while increasing the distance to the rest

- **Perceptron Algorithm**

- Finds separating hyperplane iteratively
- Starts with a weight vector w that is incremented or decremented for every example on the wrong side of the hyperplane specified by w
- Ignores correctly classified examples
- If examples are linearly separable, the algorithm converges after finite number of steps, otherwise it fails to terminate
- For practical purposes it's sufficient to stop training after some time (under the assumption that a good classifier has been found)
- Attractive for filtering (simple, incremental, adaptive)

- **Margin Perceptron Algorithm**

- Increments w for examples, that are near but both on the correct and on the wrong side of the hyperplane
- Margin is defined to be the distance to the nearest example in Euclidean space
- Sets a margin parameter τ so as to bias the method to prefer higher margin separators
- Where standard perceptron would stop, margin perceptron continues to adjust the hyperplane until margin of $\tau/|w|$ is achieved

- **Support Vector Machines (SVM)**
 - Directly computes the separating hyperplane that maximizes the margin or distance to the nearest example points
 - Several points will be at the same distance → support vectors
 - Resulting classifier is a linear combination of support vectors (other points may be ignored)
 - In the case of nonseparable data or of separable data in which few points dramatically affect the solution → relax requirement that all training data has to be correctly classified
 - Trade-off between maximizing the margin and minimizing the magnitude of training errors
 - Software packages implementing SVMs → e.g. Weka, SVM-light, LibSVM

- Slides:
 - Text Classification: An Advanced Tutorial; William W. Cohen
 - Hyperplane based Classification: Perceptron and (Intro to) Support Vector Machines; Piyush Rai
 - Introduction to Support Vector Machines; Colin Campbell
- Books:
 - Information Retrieval: Implementing and Evaluating Search Engines; Stefan Büttcher, Charles L.A. Clarke, Gordon V. Cormack; 2010; The MIT Press