

Feature Selection in Text

Applied Text Mining

Ayoub Bagheri

Table of Contents

Lecture's plan.....	2
An illustration of VS model.....	2
Feature selection: What.....	3
Feature selection: What.....	3
Feature selection: What.....	3
Feature selection: What.....	4
Feature selection: Why	4
Why accuracy reduces.....	4
Noise / Explosion	5
Feature selection	5
Feature selection for text	5
Feature Selection Methods.....	5
Feature selection methods	5
Filters, Wrappers, Embedded, and Hybrid.....	6
Wrapper Methods.....	6
Feature selection methods	6
Feature selection: Wrappers.....	7
Wrapper method	7
Wrappers for feature selection	8
Search strategies.....	8
Filter Methods	8
Filter method	8
Document frequency.....	9
Information gain	9
Gini index.....	9
Gini index.....	10
Feature scoring metrics	10

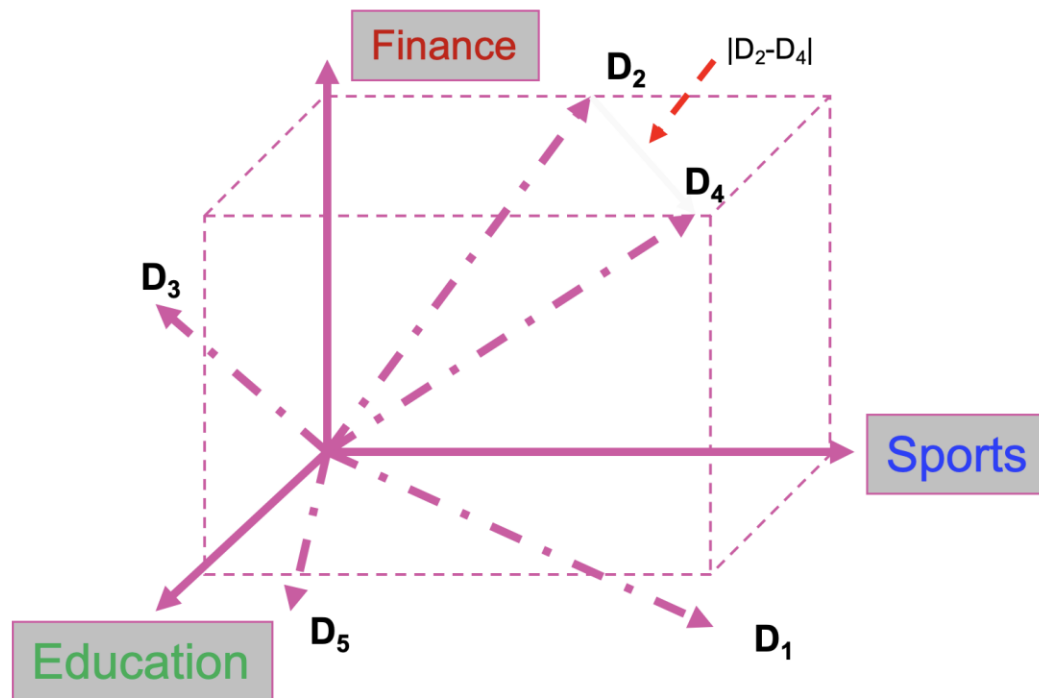
Other metrics	10
Embedded Methods	10
Formalism	10
Formalism	11
Lasso vs Ridge.....	11
The l_1 SVM	11
The l_0 SVM	12
Comparing methods	12
PCA.....	13
Feature selection vs feature reduction	13
PCA: Principal Component Analysis	13
PCA overview	13
PCA overview	14
Evaluation Supervised learning Which method to use?	14
Data Splitting.....	14
Cross Validation	15
Confusion matrix	16
Accuracy.....	17
Precision and recall	18
A combined measure: F.....	19
Summary	19
Summary.....	19
Time for Practical 3!.....	19

Lecture's plan

1. How to do feature selection for text data?
2. Is PCA a FS method for text?
3. Other methods?

An illustration of VS model

- All documents are projected into this concept space



Feature selection: What

You have some data, and you want to use it to build a classifier, so that you can predict something (e.g. email spam classification)

Feature selection: What

You have some data, and you want to use it to build a classifier, so that you can predict something (e.g. email spam classification)

The data has 10,000 fields (features)

Feature selection: What

You have some data, and you want to use it to build a classifier, so that you can predict something (e.g. email spam classification)

The data has 10,000 fields (features)

you need to cut it down to 1,000 fields before you try machine learning. Which 1,000?

Feature selection: What

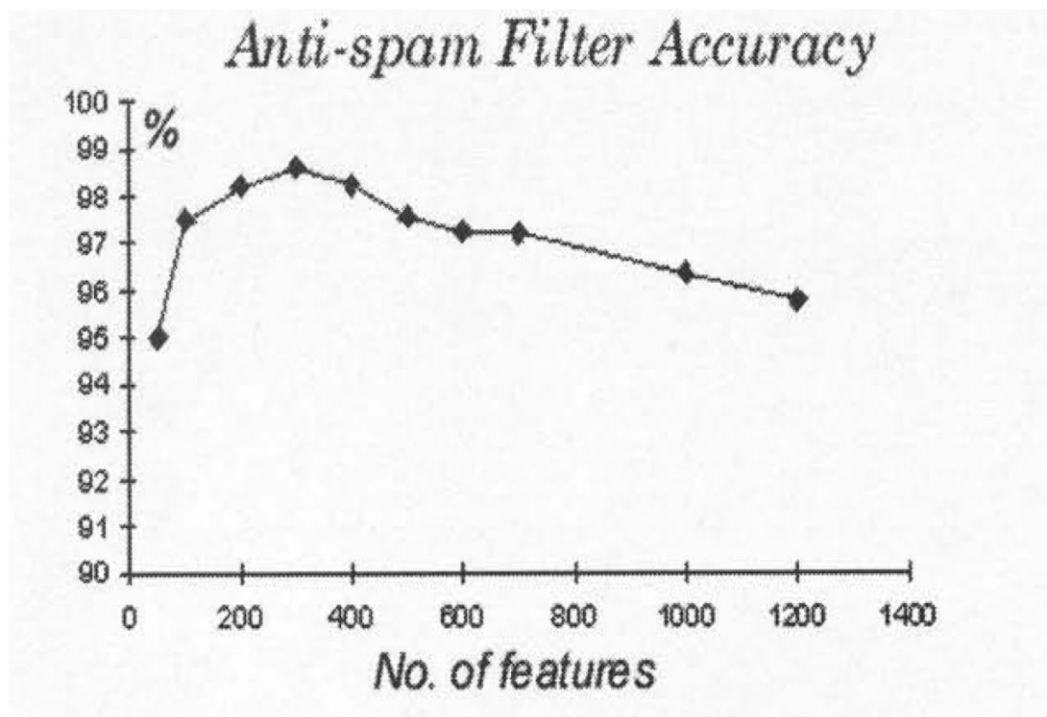
You have some data, and you want to use it to build a classifier, so that you can predict something (e.g. email spam classification)

The data has 10,000 fields (features)

you need to cut it down to 1,000 fields before you try machine learning. Which 1,000?

The process of choosing the 1,000 fields to use is called Feature Selection

Feature selection: Why



From <http://elpub.scix.net/data/works/att/02-28.content.pdf>

Why accuracy reduces

- Suppose the best feature set has 20 features.
- If you *add* another 5 features, typically the accuracy of machine learning may reduce.
- But you still have the original 20 features!
- Why does this happen?

Noise / Explosion

- The additional features typically add *noise*. Machine learning will pick up on spurious correlations, that might be true in the training set, but not in the test set.
- For some ML methods, more features means more *parameters* to learn (more NN weights, more decision tree nodes, etc...)
- The increased space of possibilities is more difficult to search.

Feature selection

Why we need FS:

1. To improve performance (in terms of speed, predictive power, simplicity of the model).
2. To visualize the data for model selection.
3. To reduce dimensionality and remove noise.

Feature Selection is a process that chooses an optimal subset of features according to a certain criterion.

Feature selection for text

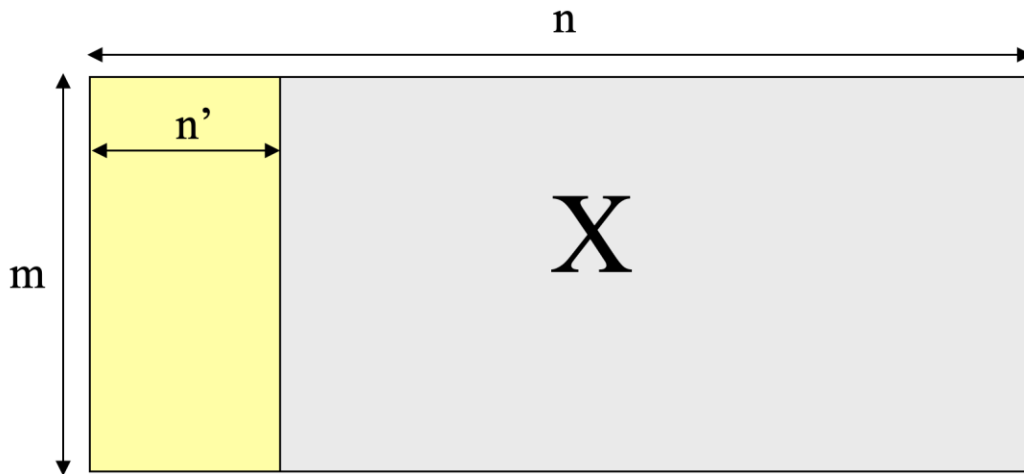
Feature selection is the process of selecting a specific subset of the terms of the training set and using only them in the classification algorithm.

- high dimensionality of text features
- Select the most informative features for model training
 - Reduce noise in feature representation
 - Improve final classification performance
 - Improve training/testing efficiency
 - Less time complexity
 - Fewer training data

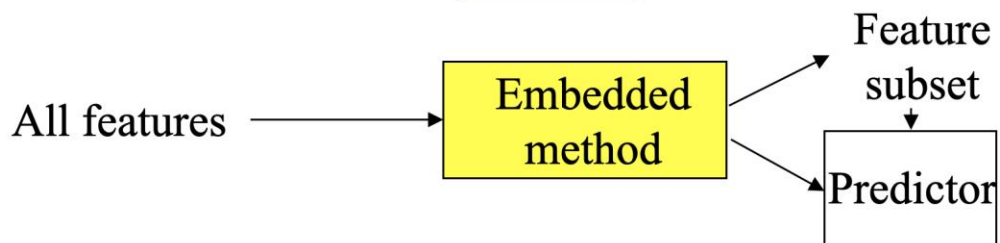
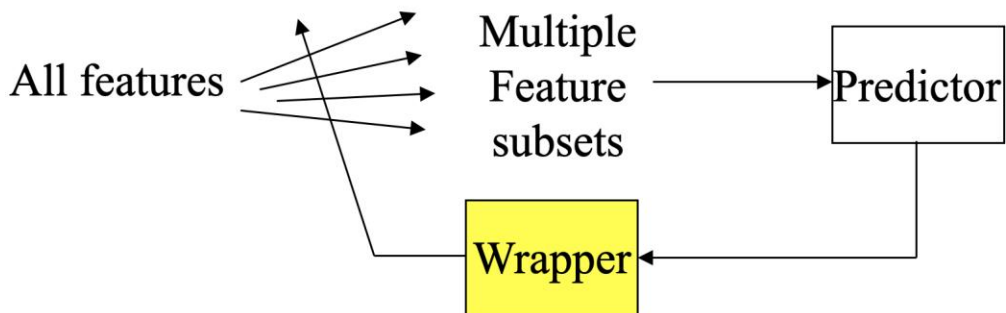
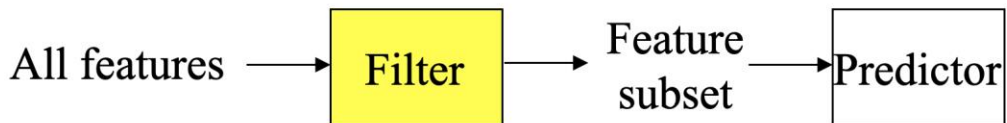
Feature Selection Methods

Feature selection methods

Thousands to millions of features: select the most relevant one to build better, faster, and easier to understand learning machines.



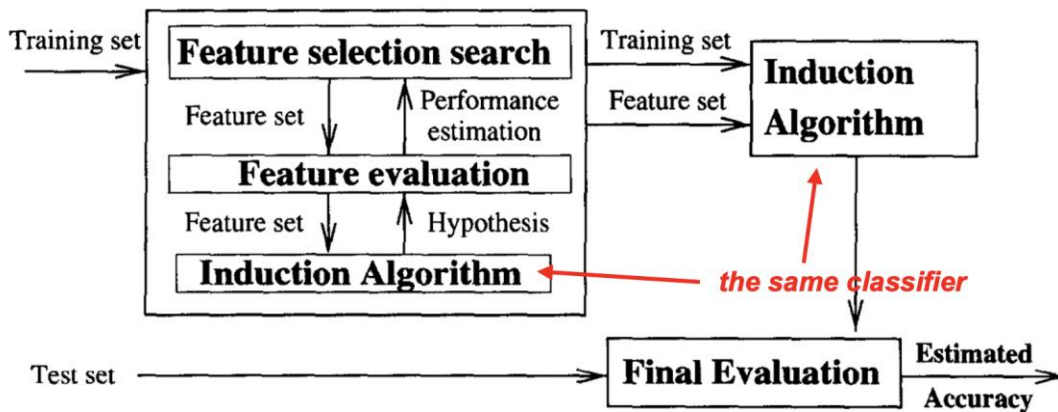
Filters, Wrappers, Embedded, and Hybrid



Wrapper Methods

Feature selection methods

- Wrapper method
 - Find the best subset of features for a particular classification method



R. Kohavi, G.H. John/Artificial Intelligence 97 (1997)
273-324

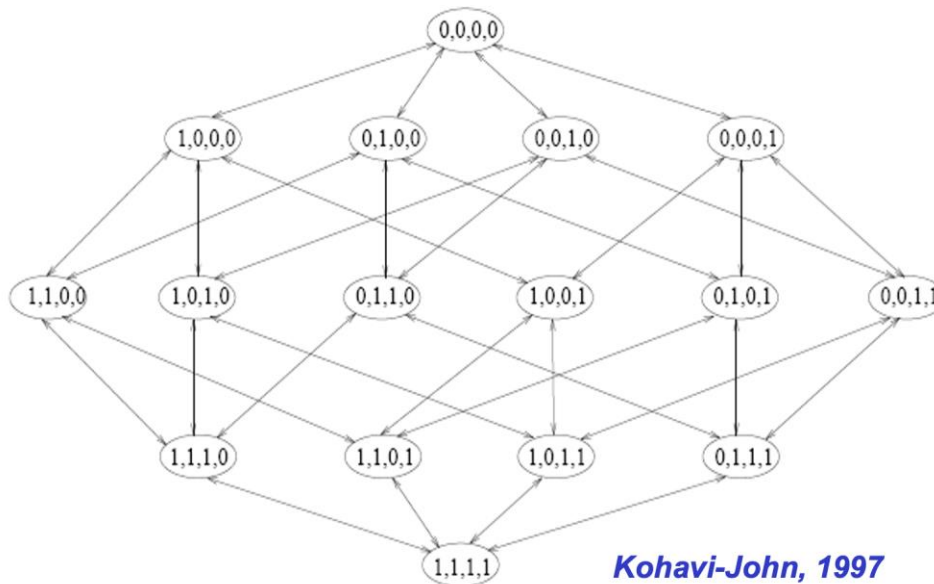
Feature selection: Wrappers

- Optimizes for a specific learning algorithm
- The feature subset selection algorithm is a “wrapper” around the learning algorithm
 1. Pick a feature subset and pass it in to learning algorithm
 2. Create training/test set based on the feature subset
 3. Train the learning algorithm with the training set
 4. Find accuracy (objective) with validation set
 5. Repeat for all feature subsets and pick the feature subset which led to the highest predictive accuracy (or other objective)
- Basic approach is simple
- Variations are based on how to select the feature subsets, since there are an exponential number of subsets

Wrapper method

- Wrapper method
 - Consider all possible dependencies among the features
 - Impractical for text categorization
 - Cannot deal with large feature set
 - A NP-complete problem
 - No direct relation between feature subset selection and evaluation

Wrappers for feature selection



N features, 2^N possible feature subsets!

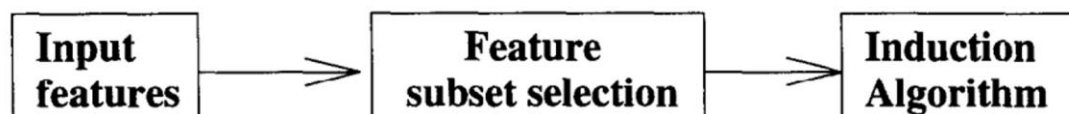
Search strategies

- Exhaustive search
- Greedy search: forward selection or backward elimination
- Simulated annealing
- Genetic algorithms

Filter Methods

Filter method

- Filter method
 - Evaluate the features independently from the classifier and other features
 - No indication of a classifier's performance on the selected features
 - No dependency among the features
 - Feasible for very large feature set



*R. Kohavi, G.H. John/Artificial Intelligence 97 (1997)
273-324*

Document frequency

- Rare words: non-influential for global prediction, reduce vocabulary size

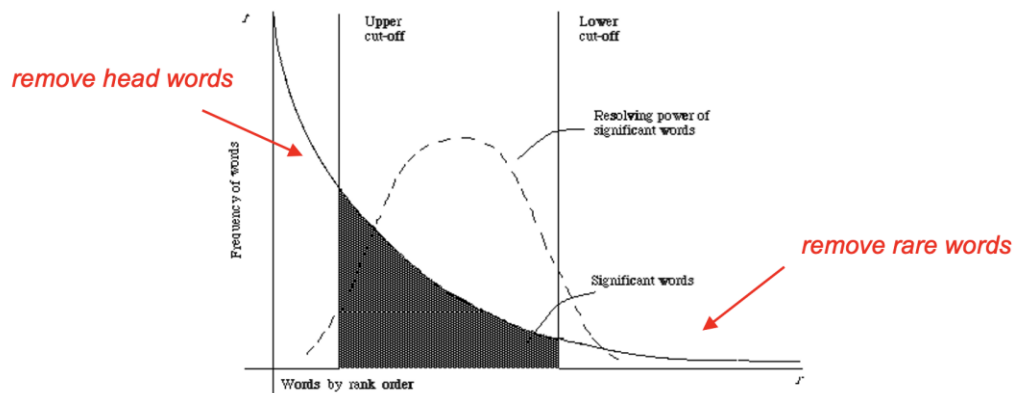


Figure 2.1. A plot of the hyperbolic curve relating f , the frequency of occurrence and r , the rank order (Adapted from Schultz, page 120)

Information gain

- Decrease in entropy of categorical prediction when the feature is present or absent

$$\begin{aligned}
 IG(t) = & - \sum_c p(c) \log p(c) && \text{Entropy of class label along} \\
 & + p(t) \sum_c p(c|t) \log p(c|t) && \text{Entropy of class label if } t \text{ is} \\
 & && \text{present} \\
 & + p(\bar{t}) \sum_c p(c|\bar{t}) \log p(c|\bar{t}) && \text{Entropy of class label if } t \text{ is} \\
 & && \text{absent}
 \end{aligned}$$

probability of seeing class label c in documents where t occurs
 probability of seeing class label c in documents where t does not occur

Gini index

Let $p(c|t)$ be the conditional probability that a document belongs to class c , given the fact that it contains the term t . Therefore, we have:

$$\sum_{c=1}^k p(c|t) = 1$$

Then, the gini-index for the term t , denoted by $G(t)$ is defined as:

$$G(t) = \sum_{c=1}^k p(c|t)^2$$

Gini index

- The value of the gini-index lies in the range $(1/k, 1)$.
- Higher values of the gini-index indicate a greater discriminative power of the term t .
- If the global class distribution is skewed, the gini-index may not accurately reflect the discriminative power of the underlying attributes.
- → normalized gini-index

Feature scoring metrics

- χ^2 statistics with multiple categories
 - $\chi^2 = \sum_c p(c) \chi^2(c, t)$
 - Expectation of χ^2 over all the categories
 - $\chi^2(t) = \max_c \chi^2(c, t)$
 - Strongest dependency between a category and a term

Other metrics

- Many other metrics (Same trick as in χ^2 statistics for multi-class cases)
 - Mutual information
 - Relatedness between term t and class c

$$PMI(t; c) = p(t, c) \log \left(\frac{p(t, c)}{p(t)p(c)} \right)$$

- Odds ratio
 - Odds of term t occurring with class c normalized by that without c

$$Odds(t; c) = \frac{p(t, c)}{1 - p(t, c)} \times \frac{1 - p(t, \bar{c})}{p(t, \bar{c})}$$

Embedded Methods

Formalism

- Many learning algorithms are cast into a minimization of some regularized functional:

$$\min_{\alpha} \hat{R}(\alpha, \sigma) = \min_{\alpha} \sum_{k=1}^m L(f(\alpha, \sigma \circ x_k), y_k) + \Omega(\alpha)$$

Formalism

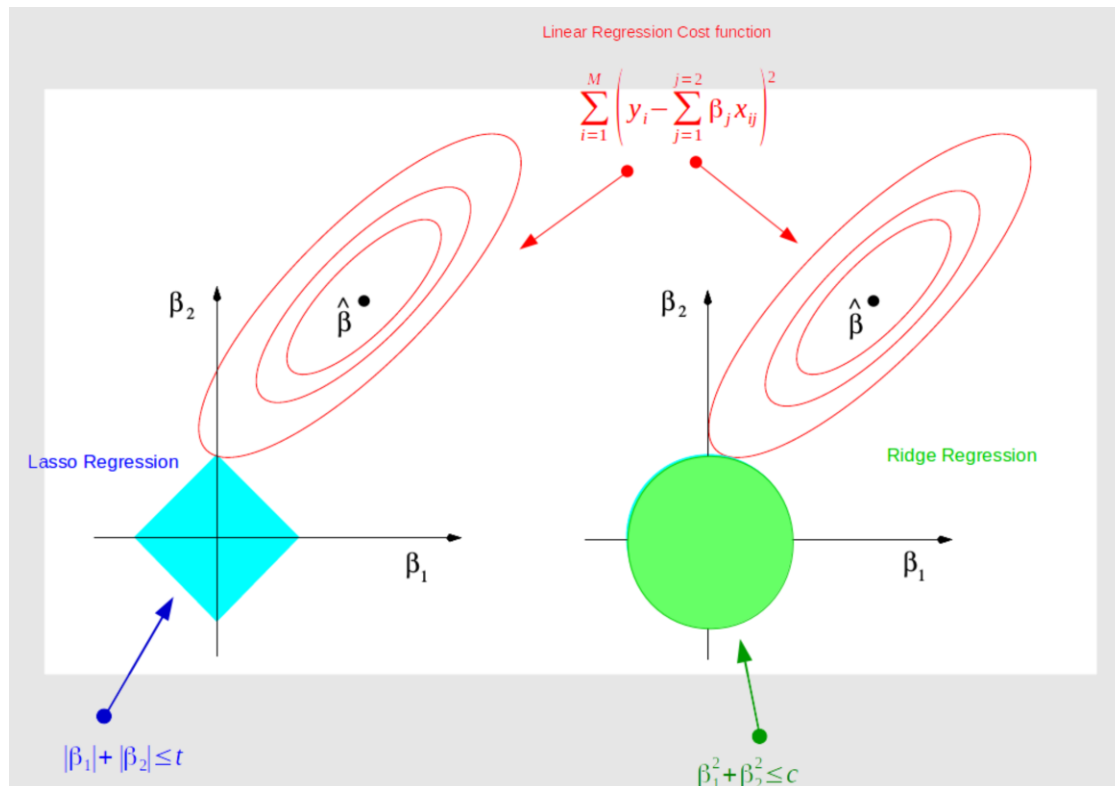
- Many learning algorithms are cast into a minimization of some regularized functional:

$$\underbrace{\min_{\alpha} \hat{R}(\alpha, \sigma)}_{G(\sigma)} = \min_{\alpha} \sum_{k=1}^m L(f(\alpha, \sigma \circ x_k), y_k) + \underbrace{\Omega(\alpha)}_{\text{Regularization capacity control}}$$

Empirical error
Regularization capacity control

Justification of RFE and many other embedded methods.

Lasso vs Ridge



The l_1 SVM

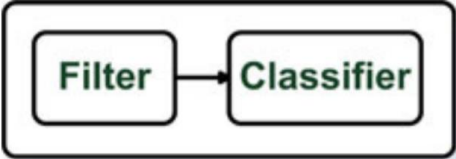
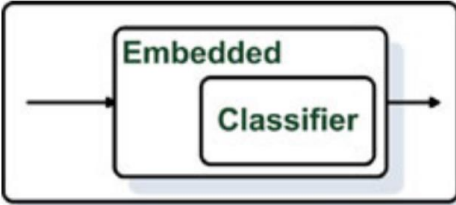
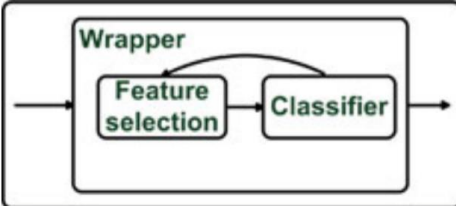
- A version of SVM where $\Omega(w) = ||w||^2$ is replaced by the l_1 norm $\Omega(w) = \sum_i |w_i|$
- Can be considered an embedded feature selection method:

- Some weights will be drawn to zero (tend to remove redundant features)
- Difference from the regular SVM where redundant features are included

The l_0 SVM

- Replace the regularizer $||w||^2$ by the l_0 norm $\sum_{i=1}^n 1_{w_i \neq 0}$
- Further replace $\sum_{i=1}^n 1_{w_i \neq 0}$ by $\sum_i \log(\epsilon + |w_i|)$
- Boils down to the following multiplicative update algorithm:
 1. Set $\sigma = (1, \dots, 1)$
 2. Get w^* solution of an SVM on data set where each input is scaled by σ
 3. Set $\sigma = |w^*| \circ \sigma$
 4. back to 2

Comparing methods

Method	Advantages	Disadvantages
Filter 	Independence of the classifier Lower computational cost than wrappers Fast Good generalization ability	No interaction with the classifier
Embedded 	Interaction with the classifier Lower computational cost than wrappers Captures feature dependencies	Classifier-dependent selection
Wrapper 	Interaction with the classifier Captures feature dependencies	Computationally expensive Risk of overfitting Classifier-dependent selection

PCA

Feature selection vs feature reduction

- *Feature Selection* seeks a *subset* of the n original features which retains most of the relevant information
 - Wrappers (e.g. forward selection), Filters (e.g. PMI), Embedded (e.g. Lasso, Regularized SVN)
- *Feature Reduction combines/fuses* the n original features into a smaller set of newly created features which hopefully retains most of the relevant information from *all* the original features - Data fusion (e.g. LDA, PCA, etc.)

PCA: Principal Component Analysis

- PCA is one of the most common feature reduction techniques
- A linear method for dimensionality reduction
- Allows us to combine much of the information contained in n features into p features where $p < n$
- PCA is *unsupervised* in that it does not consider the output class/value of an instance – There are other algorithms which do (e.g. Linear Discriminant Analysis)
- PCA works well in many cases where data have mostly linear correlations

PCA overview

- Seek new set of bases which correspond to the highest variance in the data
- Transform n -dimensional *normalized* data to a new n -dimensional basis
 - The new dimension with the most variance is the first principal component
 - The next is the second principal component, etc.
 - Note z_1 combines/fuses significant information from both x_1 and x_2
- Drop dimensions for which there is little variance

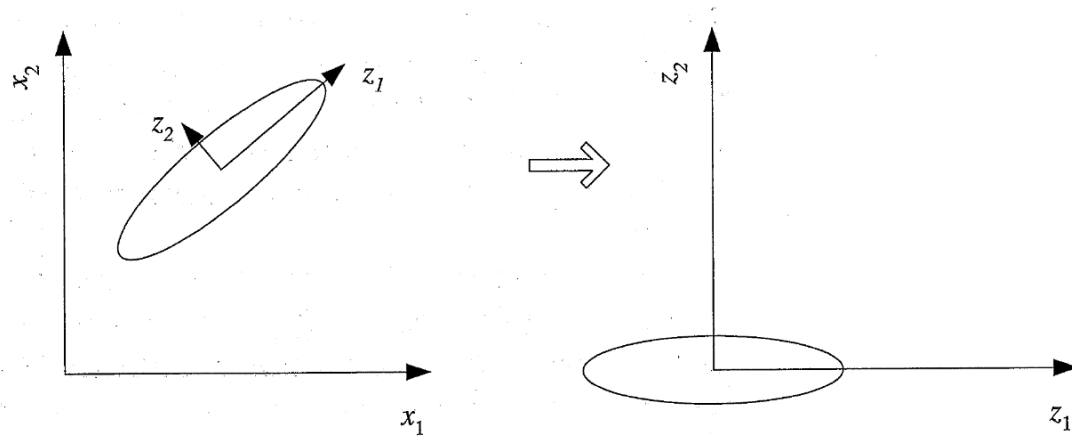
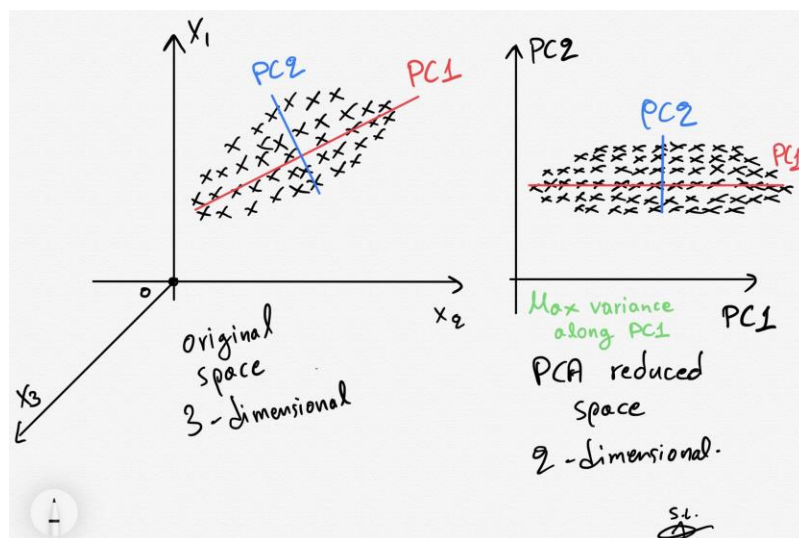


Figure 6.1 Principal components analysis centers the sample and then rotates the axes to line up with the directions of highest variance. If the variance on z_2 is too small, it can be ignored and we have dimensionality reduction from two to one.

PCA overview



<https://towardsdatascience.com/>

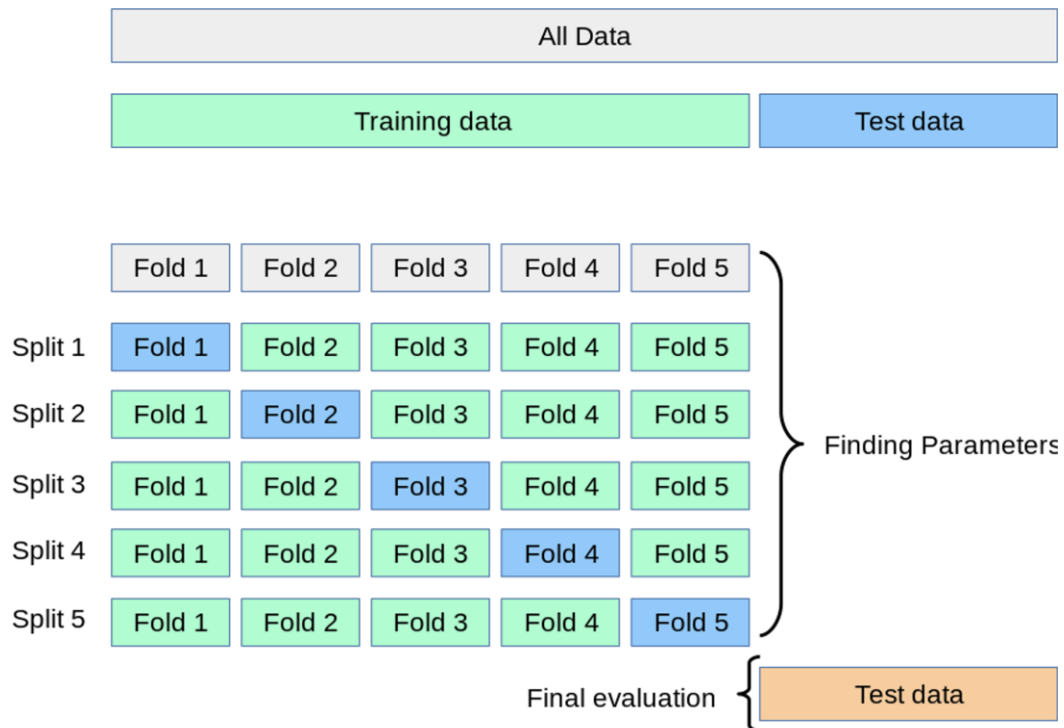
Evaluation | Supervised learning | Which method to use?

Data Splitting

- Training set
 - Validation set (dev set)

- A validation dataset is a dataset of examples used to tune the hyperparameters (i.e. the architecture) of a classifier. It is sometimes also called the development set or the “dev set”.
- Test set

Cross Validation



https://scikit-learn.org/stable/modules/cross_validation.html

Confusion matrix

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Accuracy

- What proportion of instances is correctly classified?
$$\frac{TP + TN}{TP + FP + FN + TN}$$
- Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed.
- Let us say that our target class is very sparse. Do we want accuracy as a metric of our model performance? What if we are predicting if an asteroid will hit the earth? Just say “No” all the time. And you will be 99% accurate. The model can be reasonably accurate, but not at all valuable.

Precision and recall

- Precision: % of selected items that are correct Recall: % of correct items that are selected
- Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.
- Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.

A combined measure: F

A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

The harmonic mean is a very conservative average;

Balanced F1 measure - i.e., with $\beta = 1$ (that is, $\alpha = 1/2$): $F = 2PR/(P + R)$

Summary

Summary

- Feature selection for text
- Different methods
- Can be quite effective!

Time for Practical 3!