Cross-Validation & Feature Selection

Machine Learning

Hamid R Rabiee – Zahra Dehghanian

Spring 2025

Sharif University
of Technology

Outline

Cross-Validation

Dimensionality reduction

Filter univariate methods

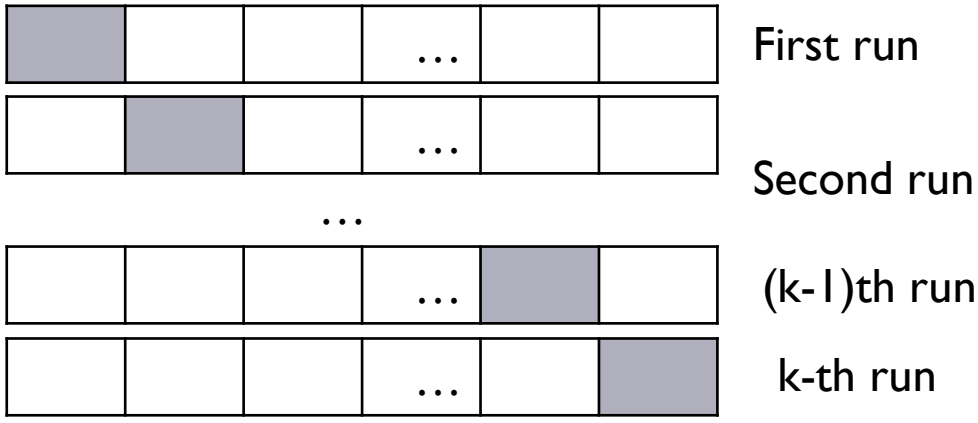
Multi-variate filter & wrapper methods

Search strategies

Embedded approach

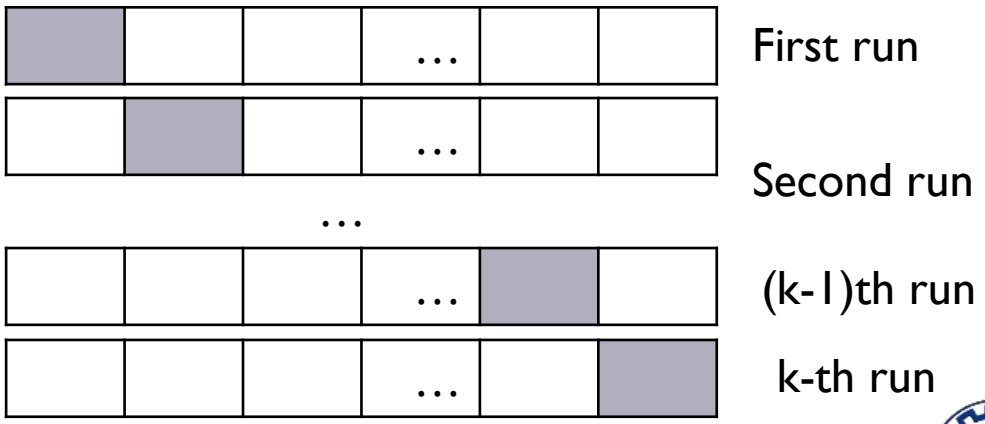
Cross-Validation (CV): Evaluation

- k-fold cross-validation steps:
  - Shuffle the dataset and randomly partition training data into k groups of approximately equal size
  - for $i = 1$ to k
    - Choose the i-th group as the held-out validation group
    - Train the model on all but the i-th group of data
    - Evaluate the model on the held-out group



Cross-Validation (CV): Evaluation

- k-fold cross-validation steps:
  - Shuffle the dataset and randomly partition training data into k groups of approximately equal size
  - for $i = 1$ to k
    - Choose the i-th group as the held-out validation group
    - Train the model on all but the i-th group of data
    - Evaluate the model on the held-out group
  - Performance scores of the model from k runs are **averaged**.
    - The average error rate can be considered as an estimation of the true performance of the model.



Cross-Validation (CV): Model Selection

For each model, we first find the average error by CV.

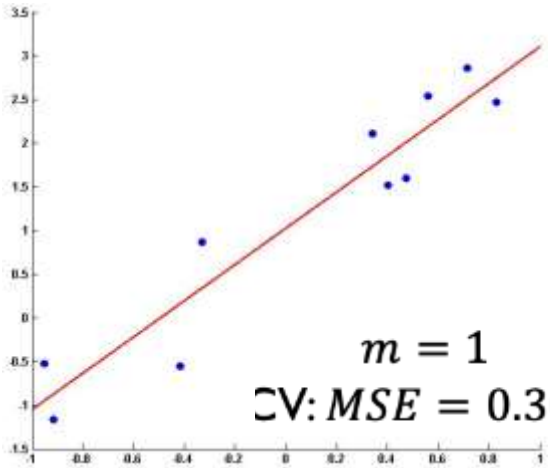
The model with **the best average performance** is selected.

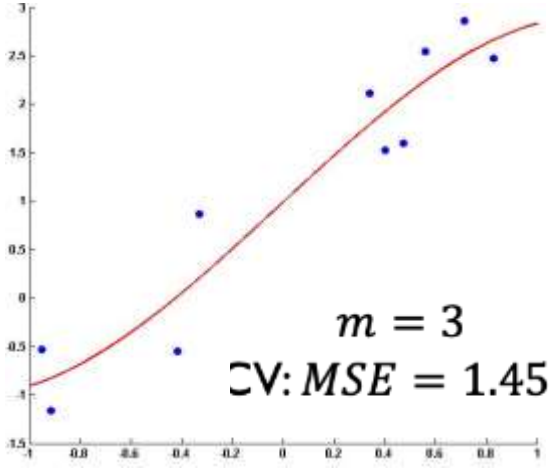
Cross-validation: polynomial regression example

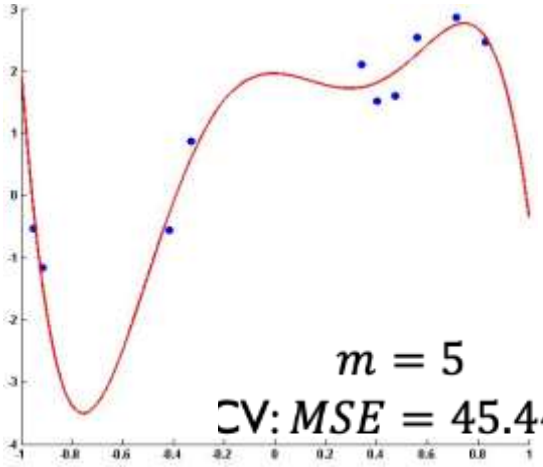
5-fold CV

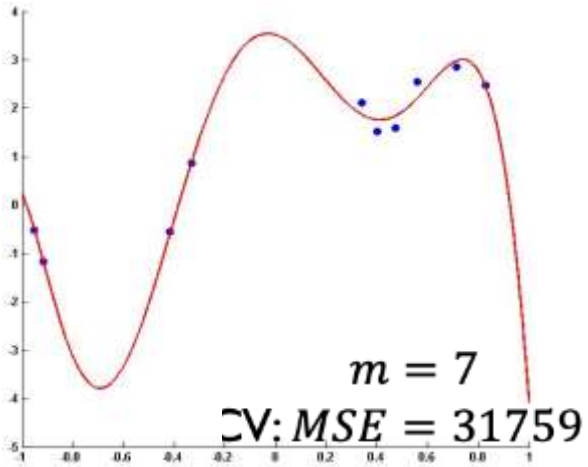
100 runs

average









Leave-One-Out Cross Validation (LOOCV)

- When data is particularly scarce, cross-validation with $k = N$
  - Leave-one-out treats each training sample in turn as a test example and all other samples as the training set.

- Use for small datasets
  - When training data is valuable
  - LOOCV can be time expensive as N training steps are required.

Dimensionality reduction: Feature selection vs. feature extraction

Feature **selection**

Select a subset of a given feature set

Feature **extraction** (e.g., PCA, LDA)

A linear or non-linear transform on the original feature space

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_{d'}} \end{bmatrix}$$

Feature Selection $(d' < d)$

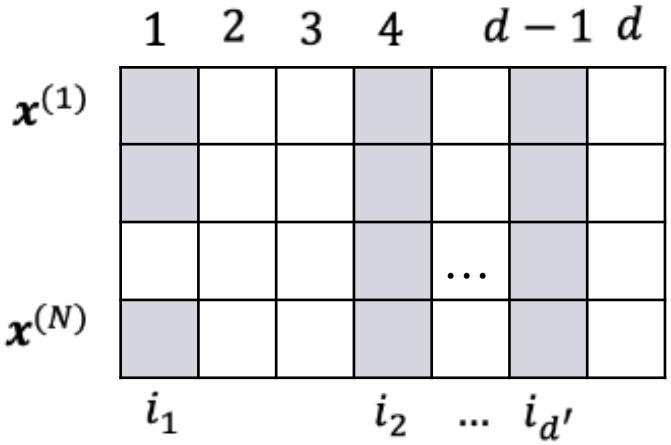
$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ \vdots \\ y_{d'} \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}\right)$$

Feature Extraction

Feature selection

- Data may contain many irrelevant and redundant variables and often comparably few training examples

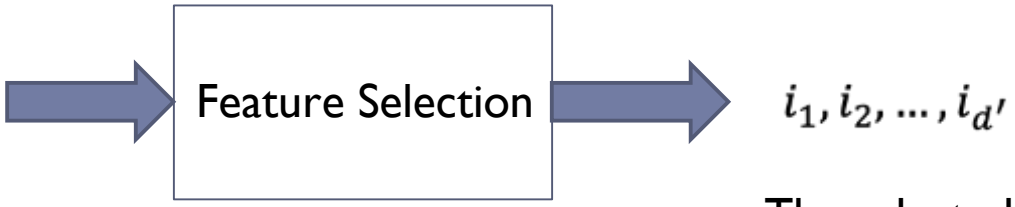
- Consider supervised learning problems where the number of features d is very large (perhaps $d \gg n$)
  - E.g., datasets with tens or hundreds of thousands of features and (much) smaller number of data samples (text or document processing, gene expression array analysis)



Why feature selection?

- FS is a way to find **more accurate, faster, and easier to understand** classifiers.
  - Performance: enhancing generalization ability
    - alleviating the effect of the curse of dimensionality
    - the higher the ratio of the no. of training patterns N to the number of free classifier parameters, the better the generalization of the learned classifier
  - Efficiency: speeding up the learning process
  - Interpretability: resulting a model that is easier to understand by human

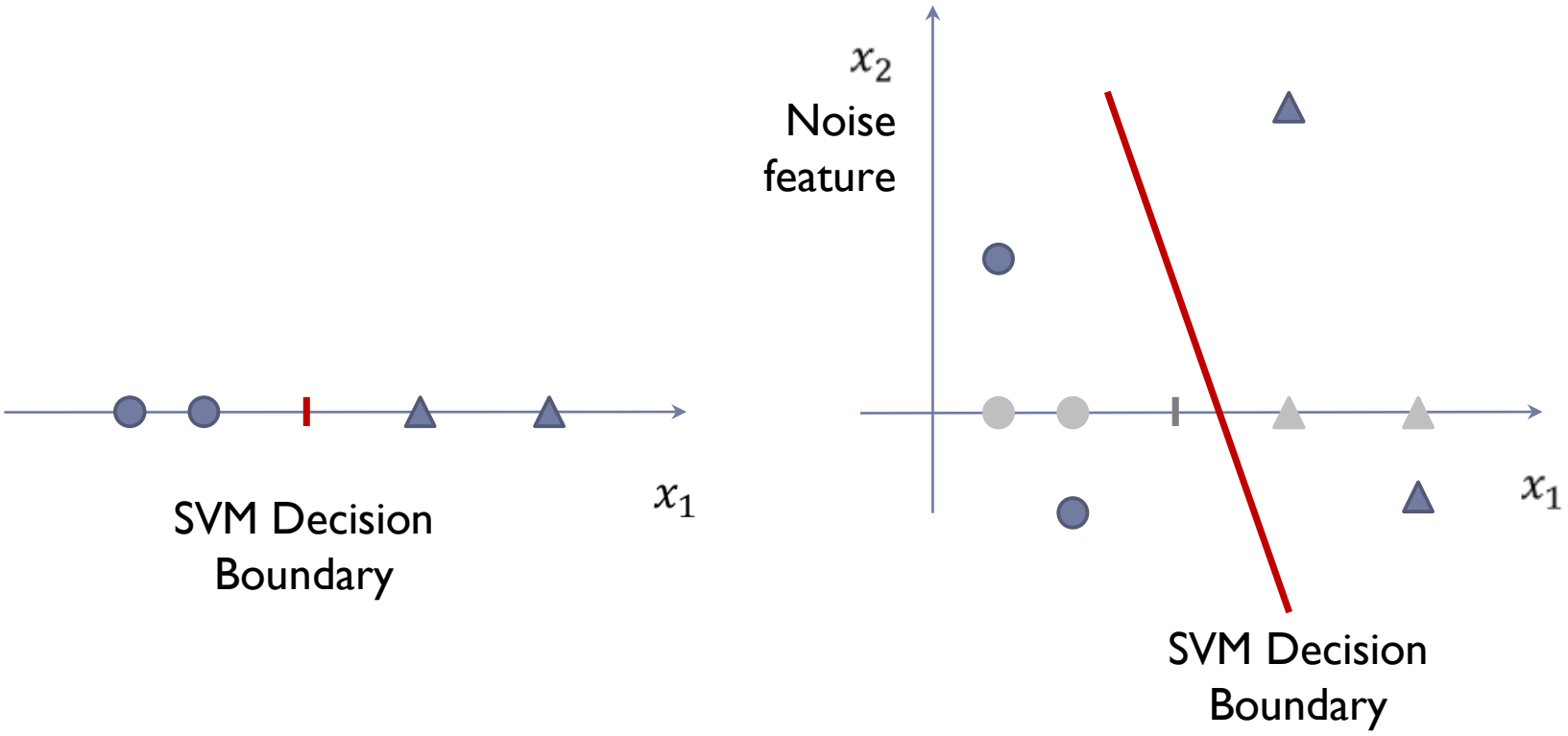
$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}, \; Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$



Supervised feature selection: Given a labeled set of data points, select a subset of features for data representation

Noise (or irrelevant) features

Eliminating irrelevant features can decrease the classification error on test data

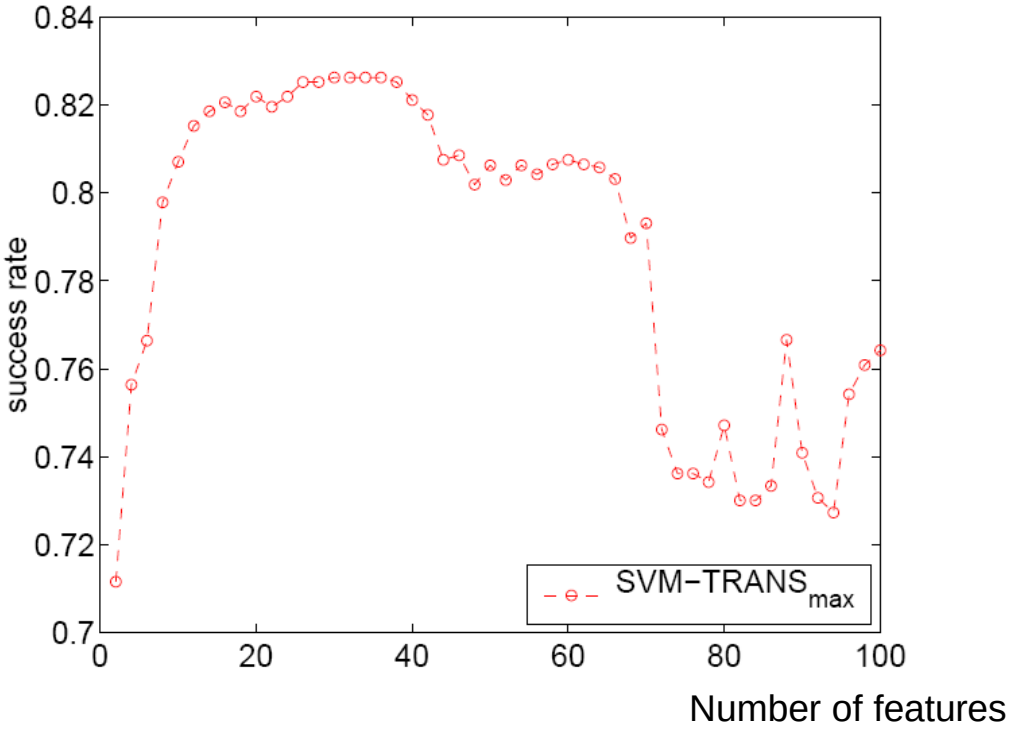


Drug Screening

$N = 1909$ compounds

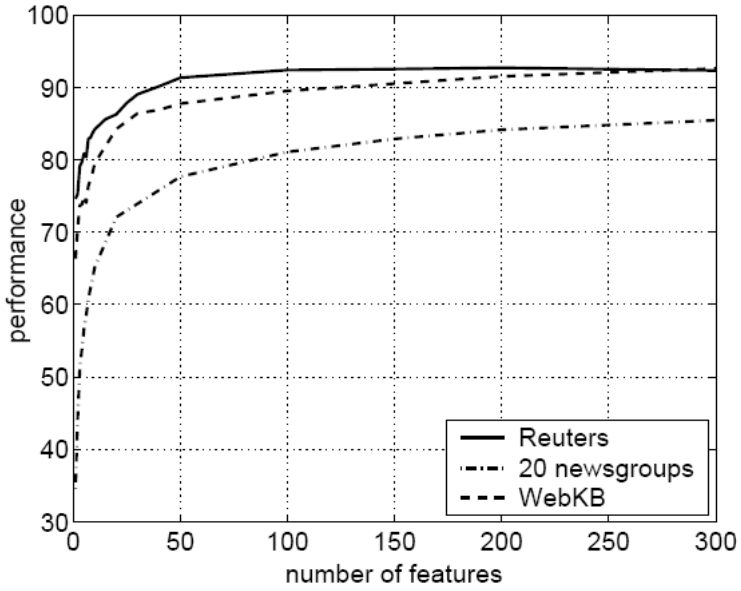
$d = 139{,}351$ binary features

three-dimensional properties of the molecule.



[Weston et al, Bioinformatics, 2002]

Text Filtering



Reuters: 21578 news wire, 114 semantic categories.

20 newsgroups: 19997 articles, 20 categories.

WebKB: 8282 web pages, 7 categories.

Bag-of-words: >100000 features.

Top 3 words of some categories:

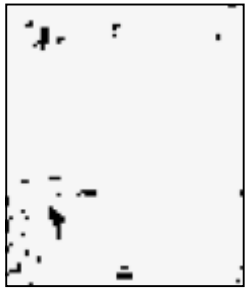
Alt.atheism: atheism, atheists, morality
Comp.graphics: image, jpeg, graphics
Sci.space: space, nasa, orbit
Soc.religion.christian: god, church, sin
Talk.politics.mideast: israel, armenian, turkish

[Bekkerman et al, JMLR, 2003]

Face Male/Female Classification

$N = 1000$ training images

$d = 60 \times 85 = 5100$ features

	100	500	1000
Relief:			
Simba:			

[Navot, Bachrach, and Tishby, ICML, 2004]

Some definitions

- **Filter method**: ranks features or feature subsets independent of the classifier as a preprocessing step.
- **Wrapper method**: uses a classifier to evaluate the score of features or feature subsets.
- **Embedded method**: Feature selection is done during the training of a classifier
  - E.g., Adding a regularization term $\|\boldsymbol{w}\|_1$ in the cost function of linear classifiers

Another categorization

Univariate method (variable ranking): considers one variable (feature) at a time.

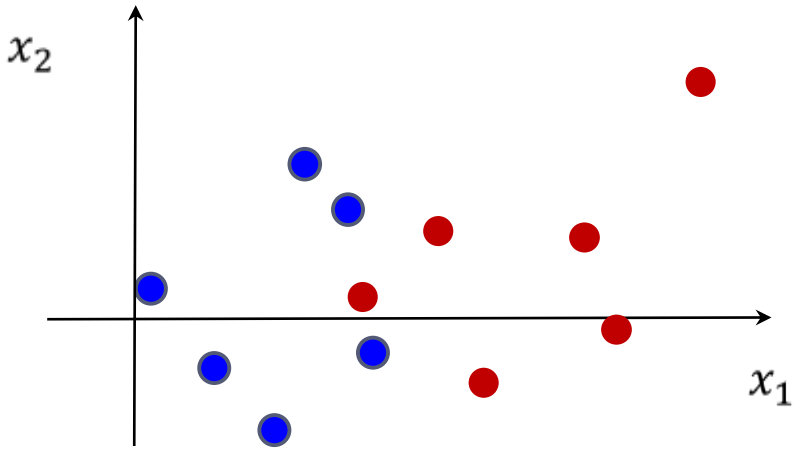
Multivariate method: considers subsets of features together.

Filter: univariate

- Univariate filter method
  - Score each feature k based on the k-th column of the data matrix and the label vector
    - Relevance of the feature to predict labels: Can the feature discriminate the patterns of different classes?
  - Rank features according to their score values and select the ones with the highest scores.
    - How do you decide how many features k to choose? e.g., using cross-validation to select among the possible values of k
- Advantage: computational and statistical scalability

Pearson Correlation Criteria

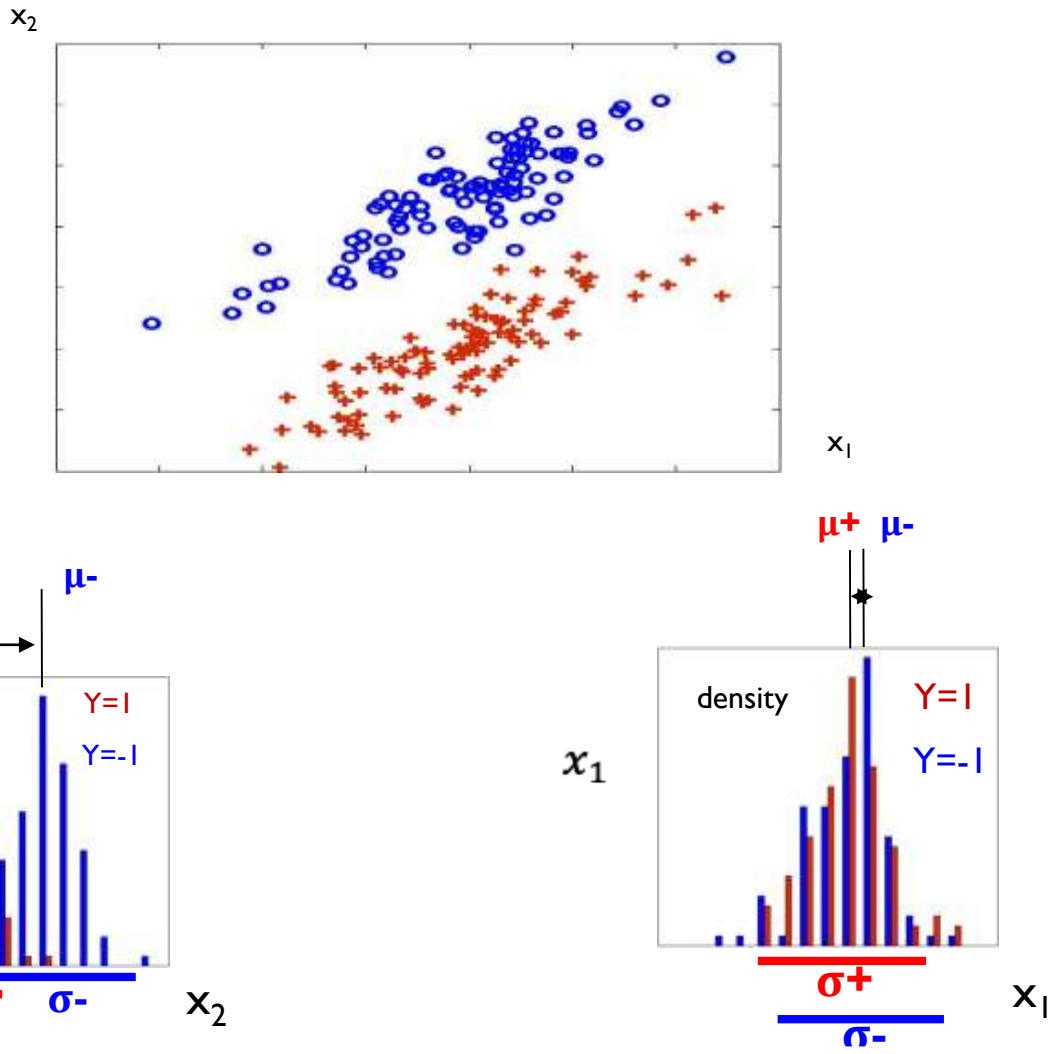
$$R(k) = \frac{cov(X_k, Y)}{\sqrt{var(X_k)}\sqrt{var(Y)}} \approx \frac{\sum_{i=1}^{N}\left(x_k^{(i)} - \overline{x_k}\right)\left(y^{(i)} - \bar{y}\right)}{\sqrt{\sum_{i=1}^{N}\left(x_k^{(i)} - \overline{x_k}\right)^2 \sum_{i=1}^{N}(y^{(i)} - \bar{y})^2}}$$



$$R(1) \gg R(2)$$

- can only detect linear dependencies between feature and target.

Single Variable Classifier



Univariate Mutual Information

- Independence:

$$P(X,Y) = P(X)P(Y)$$

- Mutual information as a measure of dependence:

$$MI(X,Y) = E_{X,Y}\left[\log \frac{P(X,Y)}{P(X)P(Y)}\right]$$

- Score of X_k based on MI with Y:
  - $I(k) = MI(X_k, Y)$

Mutual Information

$$I(X;Y;Z) = I(X;Y) + I(X;Z|Y)$$

Mutual Information

$$I(X;Y;Z) = I(X;Y) + I(X;Z|Y)$$

$$I(X;Y) = H(X) - H(X|Y)$$

$$H(X) = -\sum_{x} P(x) \log P(x)$$

Entropy!

Filter – univariate: Disadvantage

Univariate methods may fail:

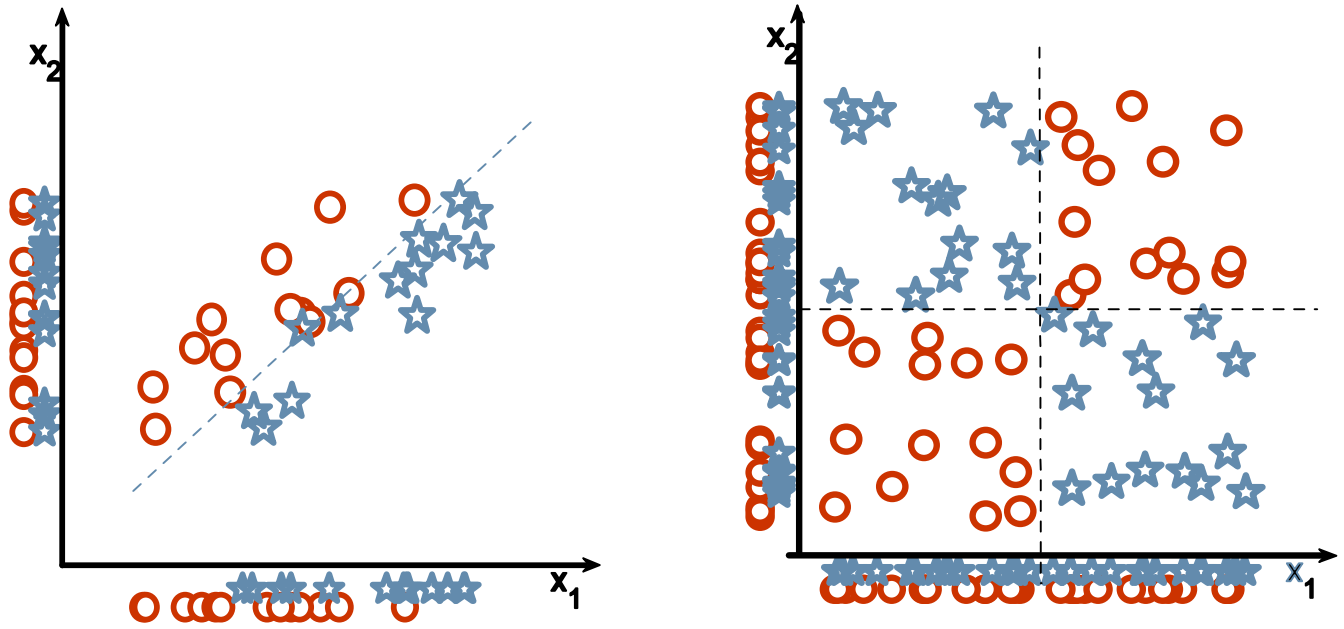
a feature may be important in combination with other features.

Redundant features:

They can select a group of dependent variables that carry similar information about the output, i.e. it is sufficient to use only one (or a few) of these variables.

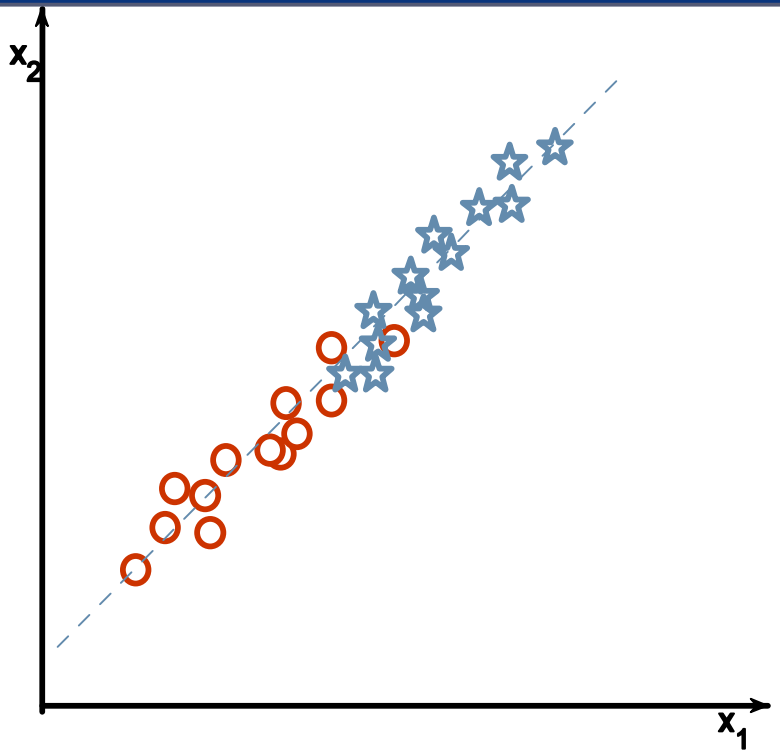
Univariate methods: Failure

Samples on which univariate feature analysis and scoring fails:



[Guyon-Elisseeff, JMLR 2004; Springer 2006]

Redundant features

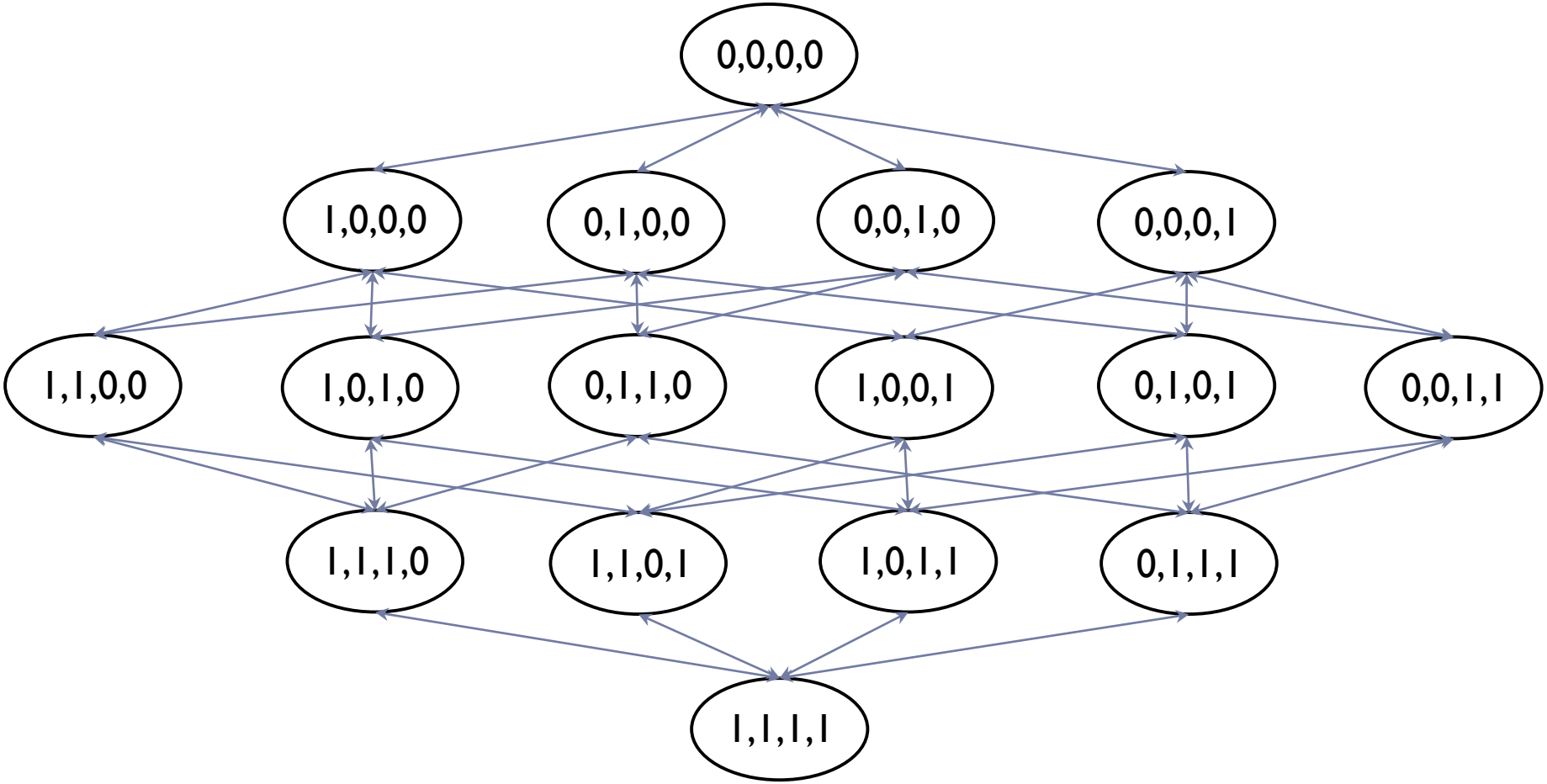


What is the relation between redundancy and correlation:
Are highly correlated features necessarily redundant?
What about completely correlated ones?

Multi-variate feature selection

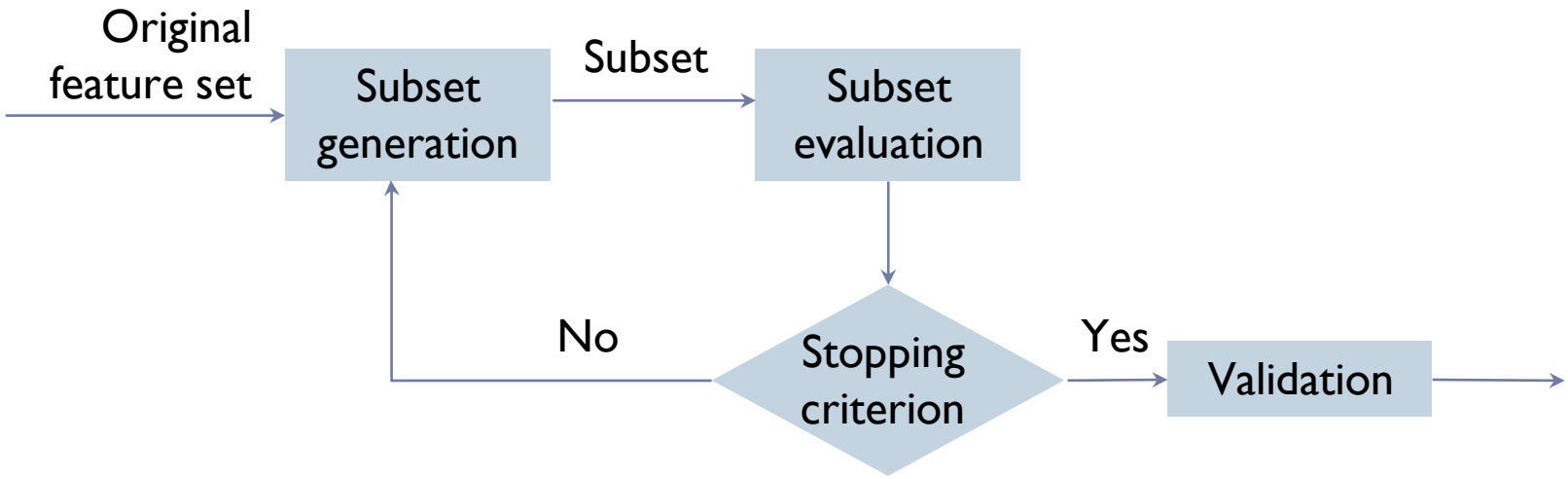
- Search in the space of all possible combinations of features.
  - all feature subsets: For d features, 2^d possible subsets.
  - high computational and statistical complexity.

Search space for feature selection ($d = 4$)



[Kohavi-John,1997]

Multivariate methods: General procedure



Subset Generation: select a candidate feature subset for evaluation

Subset Evaluation: compute the score (relevancy value) of the subset

Stopping criterion: when stopping the search in the space of feature subsets

Validation: verify that the selected subset is valid

Stopping criteria

Predefined number of features is selected

Predefined number of iterations is reached

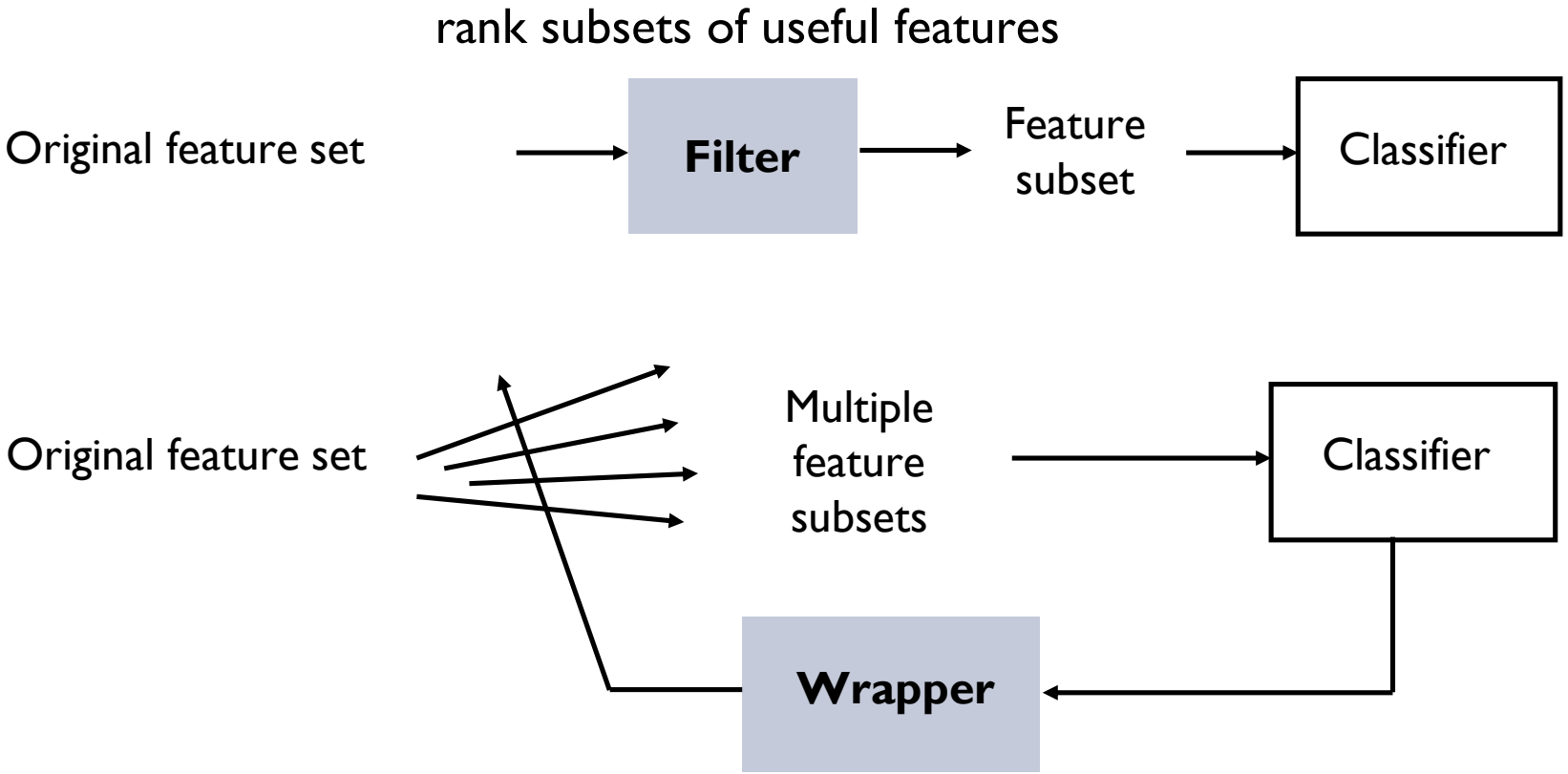
Addition (or deletion) of any feature does not result in a better subset

An optimal subset (according to the evaluation criterion) is obtained.

Filter and wrapper methods

- Wrappers use the classifier performance to evaluate the feature subset utilized in the classifier.
  - Training 2^d classifiers is infeasible for large d.
  - Most wrapper algorithms use a heuristic search.
- Filters use an evaluation function that is cheaper to compute than the performance of the classifier
  - e.g. correlation coefficient

Filters vs. wrappers



take classifier into account to rank feature subsets (e.g., using cross validation to evaluate features)

Wrapper methods: Performance assessment

For each feature subset, train classifier on training data and assess its performance using evaluation techniques like cross-validation

Filter methods: Evaluation criteria

Distance (Euclidean distance)

Class separability: Features supporting instances of the same class to be closer in terms of distance than those from different classes

Dependency (correlation coefficient, mutual information, …)

good feature subsets contain features highly dependent with the class, yet they aren't highly dependent with each other

minimum Redundancy Maximum Relevance (mRMR)

Consistency (min-features bias)

Selects features that guarantee no inconsistency in data

inconsistent instances have the same feature vector but different class labels

Prefers smaller subset with consistency (min-feature)

inconsistent

	f_1	f_2	class
instance 1	a	b	c1
instance 2	a	b	c2

How to search the space of feature subsets?

NP-hard problem.

Complete search is possible only for small number of features.

Greedy search is often used (*forward selection* or *backward elimination*).

Subset selection or generation

Search direction
- Forward
- Backward
- Random

Search strategies
- **Exhaustive - Complete**
  - Branch & Bound
  - Best first search
- **Heuristic or greedy**
  - Sequential forward selection
  - Sequential backward elimination
  - Plus-l Minus-r Selection
  - Bidirectional Search
  - Sequential floating Selection
- **Non-deterministic**
  - Simulated annealing
  - Genetic algorithm

Search strategies

- Complete: Examine all combinations of feature subset
  - Optimal subset is achievable
  - Too expensive if d is large
- Heuristic: Selection is directed under certain guidelines
  - Incremental generation of subsets
  - Smaller search space and thus faster search
  - May miss out feature sets of high importance
- Non-deterministic or random: No predefined way to select feature candidate (i.e., probabilistic approach)
  - Optimal subset depends on the number of trials
  - Need more user-defined parameters

Filters vs. Wrappers

Filters

Fast execution: evaluation function computation is faster than a classifier training

Generality: Evaluate intrinsic properties of the data, rather than their interactions with a particular classifier ("good" for a larger family of classifiers)

Tendency to select large subsets: Their objective functions are generally monotonic (so tending to select the full feature set and a cutoff is required).

Wrappers

Slow execution: must train a classifier for each feature subset (or several trainings if cross-validation is used)

Lack of generality: the solution lacks generality since it is tied to the bias of the classifier used in the evaluation function.

Ability to generalize: Since they typically use cross-validation measures to evaluate classification accuracy, they have a mechanism to avoid overfitting.

Accuracy: Generally achieve better accuracy than filters since they find a proper feature set for the intended classifier.

Examples of embedded methods

Decision trees have a built-in mechanism to perform variable selection

Nested subset methods

(input) node pruning techniques in neural networks are feature selection algorithms.

Direct objective optimization

Combines **goodness-of-fit** and the **number of variables** in the objective function

Direct objective optimization: example

- Performs feature selection as part of learning procedure:

$$f(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

$$J(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N} l\left(f\left(\boldsymbol{x}^{(n)};\boldsymbol{w}\right), y^{(n)}\right) + \lambda\|\boldsymbol{w}\|_p$$

- In the limit as $p \to 0$, the $\|\boldsymbol{w}\|_p$ is just the number of non-zero weights, i.e., the number of selected features.

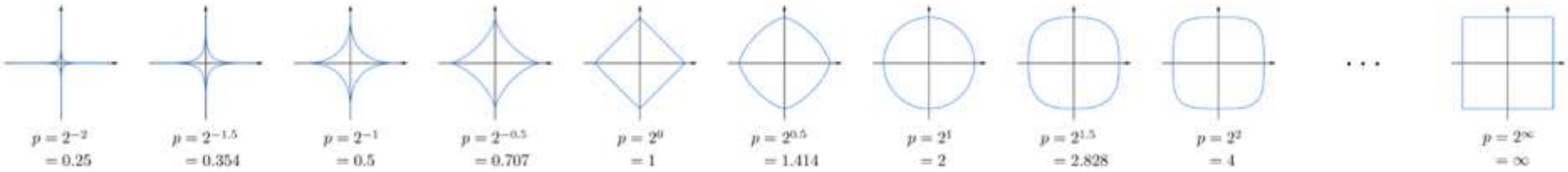
- For some application, the L1-norm minimization suffices to drive enough weights to zero.
  - Lasso: $\|w\|_1$ as regularization term

Lp-Norms

Minkowski distance:

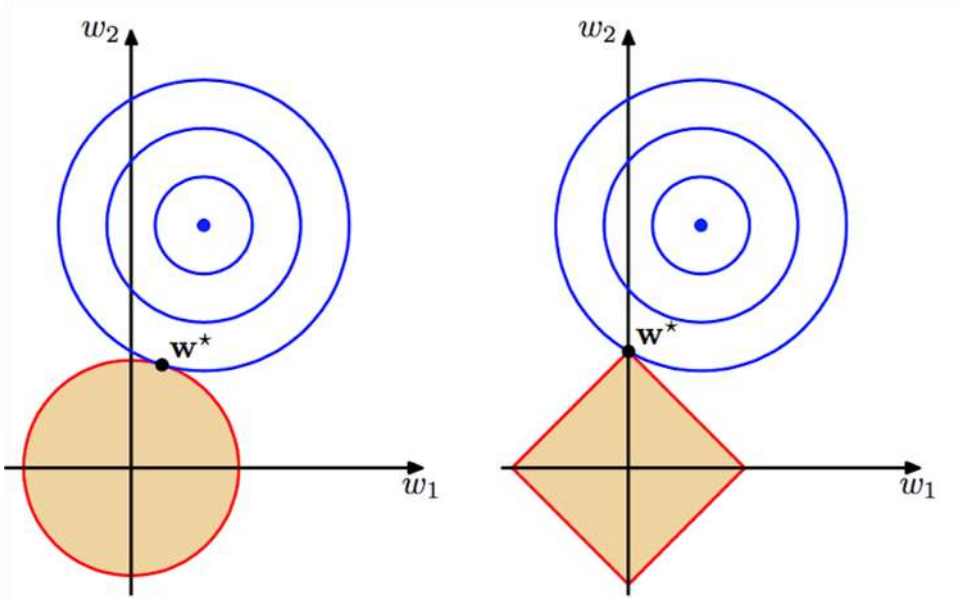
$$\boldsymbol{x} = (x_1, \ldots, x_d)$$
$$\boldsymbol{x}' = (x'_1, \ldots, x'_d)$$

$$D(\boldsymbol{x}, \boldsymbol{x}') = \left(\sum_{i=1}^{d} |x_i - x'_i|^p \right)^{1/p}$$



[wikipedia]

Example



[Bishop]

L1 regularization: the number of zero weights increases and thus shows feature selection property

References

I. Guyon and A. Elisseeff, An Introduction to Variable and Feature Selection, JMLR, vol. 3, pp. 1157-1182, 2003.

S. Theodoridis and K. Koutroumbas, Pattern Recognition, 4th edition, 2008. [Chapter 5]

H. Liu and L. Yu, Feature Selection for Data Mining, 2002.

Course CE-717, Dr. M.Soleymani

Feed back

https://forms.gle/vKRbyVVsWRKcZuqr8