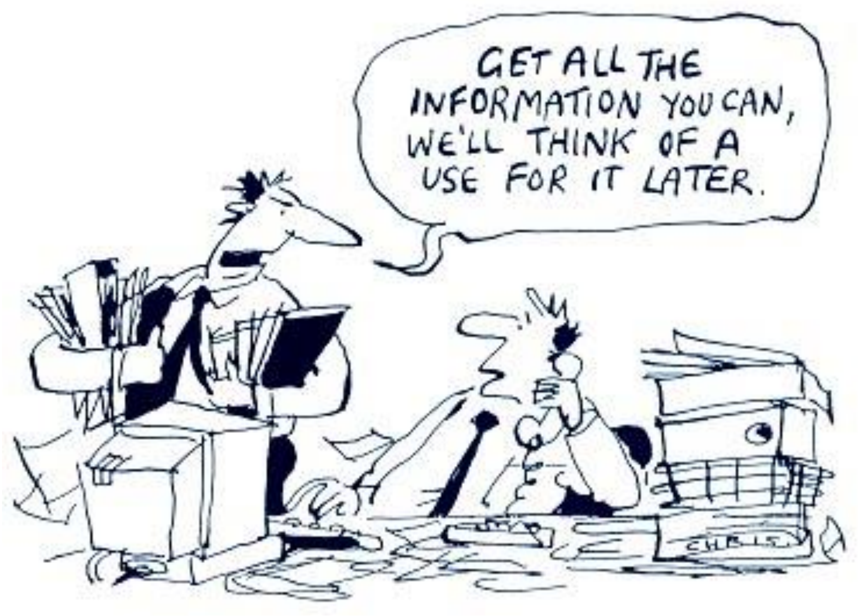
University of Ljubljana, Faculty of Computer and Information Science

Feature evaluation and selection



Prof Dr Marko Robnik-Šikonja

Intelligent Systems, November 2018

Contents

- why feature subset selection?
- filter, wrapper and embedded methods
- in classification: heuristic and optimization based methods
- extensions to supervised learning: multi-task, multi-view, and multi-label learning
- unsupervised and semi-supervised learning
- issues in feature subset selection: stability, redundancy, and higher order interactions.

Supervised learning

- $\mathrm{Y} = f(X)$ $$X = \begin{array}{cc} & \begin{array}{cccc} \boldsymbol{A_1} & \boldsymbol{A_2} & \cdots & \boldsymbol{A_a} \end{array} \\ \begin{array}{c} \boldsymbol{X_1} \\ \boldsymbol{X_2} \\ \vdots \\ \\ \boldsymbol{X_n} \end{array} & \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,a} \\ x_{2,1} & x_{2,2} & \cdots & \\ \vdots & \vdots & \ddots & \\ & & & \\ x_{n,1} & x_{n,2} & & x_{n,a} \end{pmatrix} \end{array}$$
- $Y_i = f(X_i) + \epsilon_i$
- classification: Y is categorical
- regression: Y is numerical

- The goal: prediction - find *f* with minimal error on new data (generalization)
- The goal: understanding – explain relationship between attributes and response

Evaluation of attributes

- numerical evaluation and ranking of the attributes
- the success of the evaluation procedure depends on the role it plays in learning:
  - feature subset selection
  - building of the tree-based models
  - constructive induction
  - discretization
  - attribute weighting
  - comprehension
  - …

Attribute description

color	weight	shape	size	sort
red	12	round	middle	apple
yellow	20	conic	large	pear
red	15	round	tiny	apple
green	8	round	small	pear
yellow	22	conic	large	apple
mixed	12	conic	small	apple
green	15	round	middle	apple
mixed	8	round	tiny	apple
yellow	6	round	small	pear

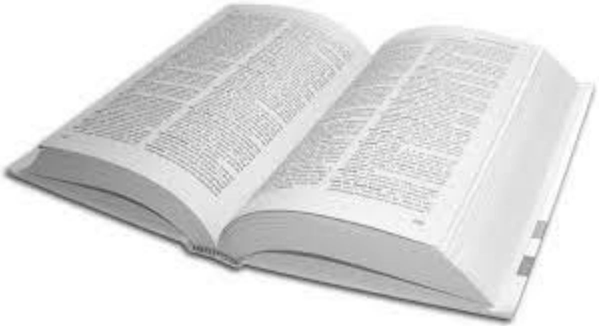
- nominal attributes: ordered and unordered
- numeric attributes

Unsupervised and semisupervised learning

- unsupervised learning: there is no Y, only X
- the goal is to find structure in X (clusters), estimate probability density, detect anomalies, generate new data from the same distribution, etc.

- semi-supervised learning: two sets of data, one with labels (X and Y), the other without Y (only X)
- the goal is to use the unsupervised sample to improve the learning performance of supervised learning

Huge number of features



- text classification, ≈ 50,000 words in a dictionary
- bioinformatics, ≈ 10,000 measurements of gene expression levels

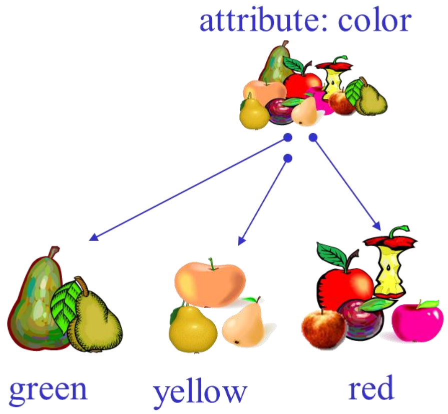
- computer vision, ≈ 1,000,000 pixels

Feature subset selection

- choose a small subset of the relevant features from the original features by removing irrelevant, redundant or noisy features
- the aim: better learning performance, i.e. higher learning accuracy, lower computational cost, or better model interpretability

Feature evaluation

- in order to select attributes we have to evaluate (rank) them
- the success of feature evaluation is measured through the success of learning
- an example: feature evaluation in decision tree building
  - in each interior node of the tree an attribute is selected which determines split of the instances
  - the attributes are evaluated to ensure useful split



Three types of feature selection methods

- filter methods: independent on learning algorithm, select the most discriminative features through a criterion based on the character of data, e.g. information gain and ReliefF
- wrapper methods: use the intended learning algorithm to evaluate the features, e.g., progressively add features to SVM while performance increases
- embedded method select features in the process of learning

Heuristic measures for attribute evaluation

- impurity based
  - information theory based (information gain, gain ratio, distance measure, J-measure)
  - probability based: Gini index, DKM, classification error on the training set
  - MDL
  - statistics G, $\aleph^2$
  - mean squared and mean absolute error (MSE, MAE)
  - assume conditional independence (upon label) between the attributes
- context sensitive measures: Relief, Contextual Merit, random forests based attribute evaluation, affinity graph based

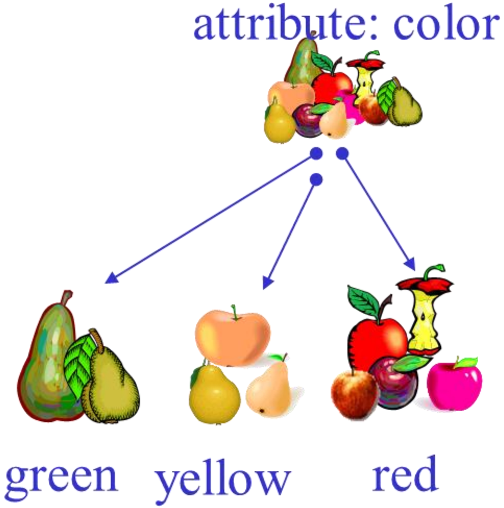
Information gain

- measure purity of labels before and after the split
- impurity = entropy

$$I(\tau) = -\sum_{i=1}^{c} p(\tau_i)\log_2 p(\tau_i)$$

$$I(\tau \mid A) = -\sum_{j=1}^{v_A} p(v_j) \sum_{i=1}^{c} p(\tau_i \mid v_j)\log_2 p(\tau_i \mid v_j)$$

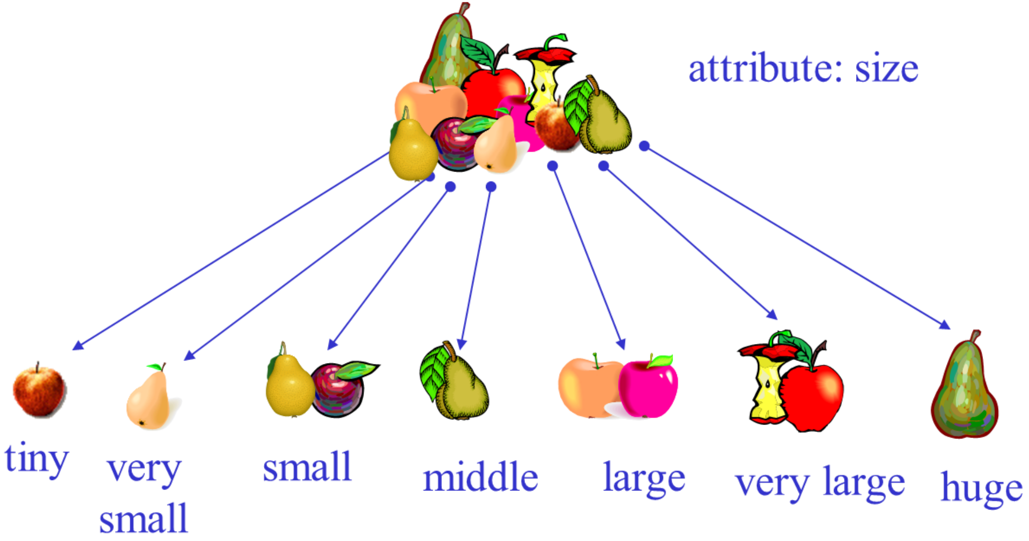
$$IG(A) = I(\tau) - I(\tau \mid A)$$



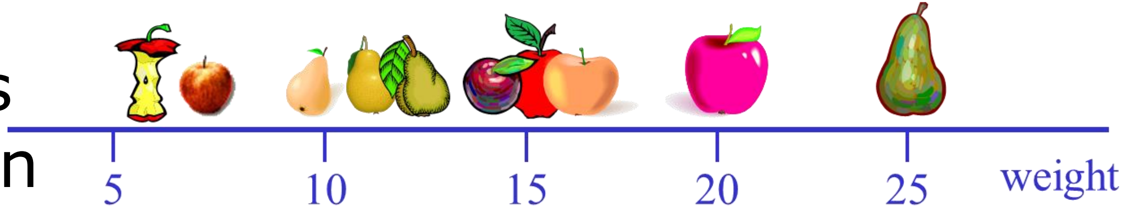
- each attribute is evaluated independently from others

Multivalued and numeric attributes

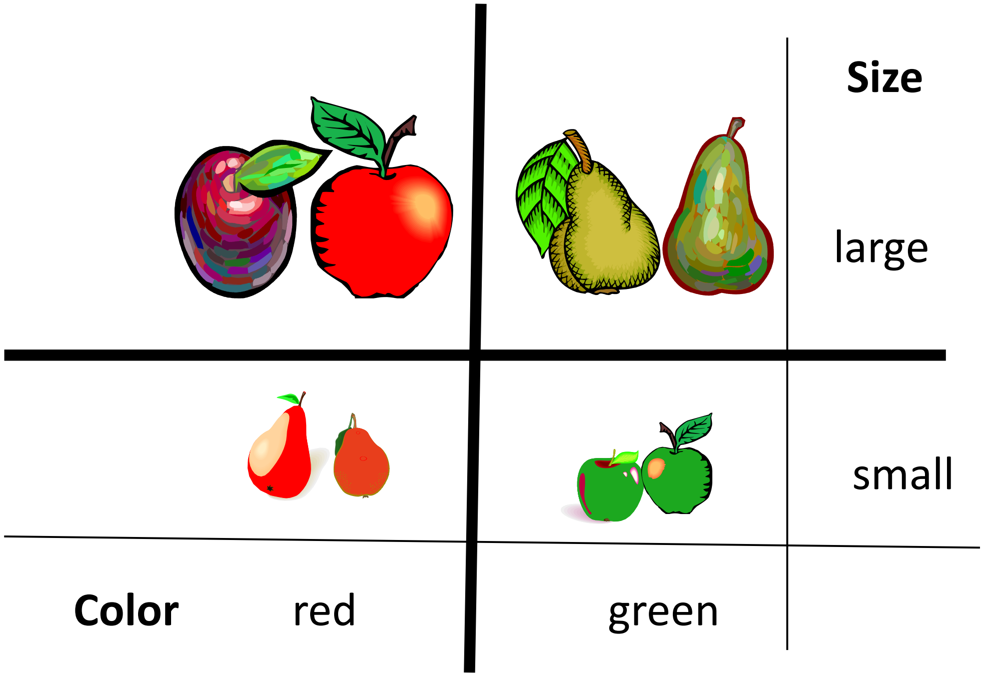
- multivalued: insufficient statistical support in certain splits



- numeric: requires prior discretization

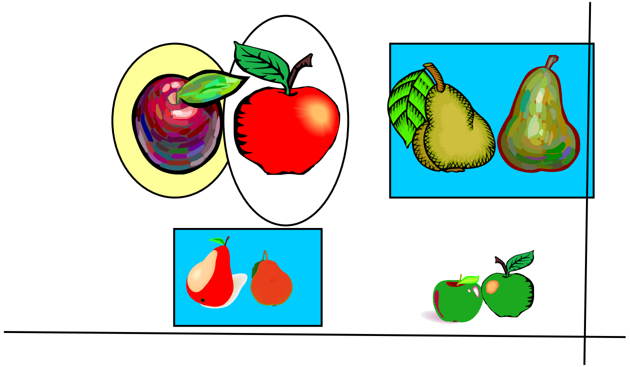


Attribute interactions



Relief algorithms

- criterion: evaluate attribute according to its power of separation between near instances



- values of good attribute should distinguish between near instances from different class and have similar values for near instances from the same class

Relief algorithms

- no assumption of conditional independence
- context sensitive
- reliable also in problems with strong conditional dependencies
- included in several machine learning systems (e.g., Weka, Orange, scikit-learn, R)

  - Relief (Kira in Rendell, 1992): two class classification
  - ReliefF (Kononenko, 1994): multi-class classification
  - RReliefF (Robnik Šikonja in Kononenko, 1997): regression

Marko Robnik-Šikonja, Igor Kononenko: Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning Journal*, 53:23-69, 2003

Algorithm Relief

Input: set of instances <x_i, τ_i>
Output: the vector W of attributes' evaluations

```
set all weights W[A] := 0.0;
for i := 1 to m do begin
  randomly select an instance R;
  find nearest hit H and nearest miss M;
  for A := 1 to #all_attributes do
    W[A] := W[A] - diff(A,R,H)/m + diff(A,R,M)/m;
end;
```

Function diff

- **for nominal attributes**

$$\mathrm{diff}(A, I_1, I_2) = \begin{Bmatrix} 0; \mathrm{value}(A, I_1) = \mathrm{value}(A, I_2) \\ 1; \mathit{otherwise} \end{Bmatrix}$$

- **for numerical attributes**

$$\mathrm{diff}(A, I_1, I_2) = \frac{\left|\mathrm{value}(A, I_1) - \mathrm{value}(A, I_2)\right|}{\max(A) - \min(A)}$$

- **distance between two instances**

$$\delta(\mathrm{I}_1, \mathrm{I}_2) = \sum_{i=1}^{a} \mathrm{diff}(A, I_1, I_2)$$

- **unknown values of attributes**

Extension ReliefF

- multi-class problems
- incomplete and noisy data
- robust
- uses with k nearest instances from all the classes

The algorithm ReliefF

Input: set of instances <x_i, τ_i>
Output: the vector W of attributes' evaluations

for v:=1 to a do W_v := 0.0;
for i := 1 **to** m **do begin**
  randomly select an instance R_i
  find k nearest hits H
  for each class $t \neq R_{i,\tau}$ **do**
    from class t find k nearest misses M(t)
  for v := 1 to a **do**
    update W_v according to update formula
end;

Update formula

$$W_v = W_v - \frac{1}{m}\operatorname{con}(A_v, R_i, H) +$$

$$\frac{1}{m}\sum_{\substack{t=1 \\ t \neq R_{i,\tau}}}^{c} \frac{p(\tau_t)\operatorname{con}(A_v, R_i, M(t))}{1 - p(R_{i,\tau})}$$

$$\operatorname{con}(A_v, R_i, S) = \frac{1}{k}\sum_{j=1}^{k} \operatorname{diff}(A_v, R_i, S_j)$$

In regression: RReliefF

$$W[A] = P(\text{different value of } A \mid \text{nearest instances with different prediction})$$
$$- P(\text{different value of } A \mid \text{nearest instances with same prediction})$$

$$W[A] = \quad = \frac{P_{dC|dA} P_{dA}}{P_{dC}} - \frac{(1 - P_{dC|dA}) P_{dA}}{1 - P_{dC}}$$

- we approximate this formula
- unified view on attribute evaluation in classification and regression

Algorithm RReliefF
Input: for each training instance a vector of attribute values **x** and predicted value $\tau(\mathbf{x})$
Output: vector W of estimations of the qualities of attributes

set all N_{dC}, $N_{dA}[A]$, $N_{dC\&dA}[A]$, $W[A]$ to 0;
for i := 1 **to** m **do begin**
  randomly select instance R_i;
  select **k** instances I_j nearest to R_i;
  for j := 1 **to** k **do begin**
    $N_{dC} := N_{dC} + \mathrm{diff}(\tau(\cdot), R_i, I_j) \cdot d(i, j)$;
    for A := 1 **to** a **do begin**
      $N_{dA}[A] := N_{dA}[A] + \mathrm{diff}(A, R_i, I_j) \cdot d(i, j)$;
      $N_{dC\&dA}[A] := N_{dC\&dA}[A] + \mathrm{diff}(\tau(\cdot), R_i, I_j) \cdot$
        $\mathrm{diff}(A, R_i, I_j) \cdot d(i, j)$;
    end;
  end;
end;
for A := 1 **to** a **do**
  $W[A] := N_{dC\&dA}[A]/N_{dC}$ - $(N_{dA}[A] - N_{dC\&dA}[A])/(m - N_{dC})$;

Relief's interpretations

- probabilistic interpretation

$$
\begin{aligned}
W[A] = {} & P(\text{different value of } A \mid \text{nearest instances with different prediction}) \\
& - P(\text{different value of } A \mid \text{nearest instances with same prediction})
\end{aligned}
$$

- ratio of the explained concept: in the limit attribute is assigned weight interpreted as a ratio between the number of prediction values it helps to determine and the number of examined instances

Regularization for feature selection

- feature selection as part of learning (embedded method)
- loss function is composed of two components: prediction error and number/weight of included features

$$L(X, Y, f) = \sum_{i=1}^{n} I(y_i \neq f(x_i)) + \lambda \sum_{j=1}^{a} I(A_j \in X)$$

- in regression we get similar expressions for ridge regression and lasso

Wrapper approach

start with an empty set of features *S={}* // forward selection

repeat

    add all unused features one by one to *S*

    train a prediction model with each set *S*

    evaluate each prediction model
        keep the best added feature in *S*

until all features are added to *S*

return the best set of features encountered

- high computational load but effective for a given learning model; attention to data overfitting

Model evaluation metrics

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Regression: MSE, MAE
- Classification: accuracy, sensitivity, specificity, AUC, precision, recall
- Comparing classifiers:
    - Confidence intervals
    - Cost-benefit analysis and ROC Curves

Classifier evaluation metrics: confusion matrix

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, $CM_{i,j}$ in a confusion matrix indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

Classification accuracy, error rate

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P’	**N’**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- **Error rate:** *1 – accuracy*, or **Error rate = (FP + FN)/All**

Sensitivity and specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P’	**N’**	**All**

Class Imbalance Problem:

One class may be *rare*, e.g. fraud, or HIV-positive

Significant *majority of the negative class* and minority of the positive class

Sensitivity: True Positive recognition rate

Sensitivity = TP/P

Specificity: True Negative recognition rate

Specificity = TN/N

Precision, recall and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Perfect score is 1.0
- Inverse relationship between precision & recall
- *F* **measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- ***$F_ß$*:** weighted measure of precision and recall
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

Example: precision and recall

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

Error depends on decision threshold

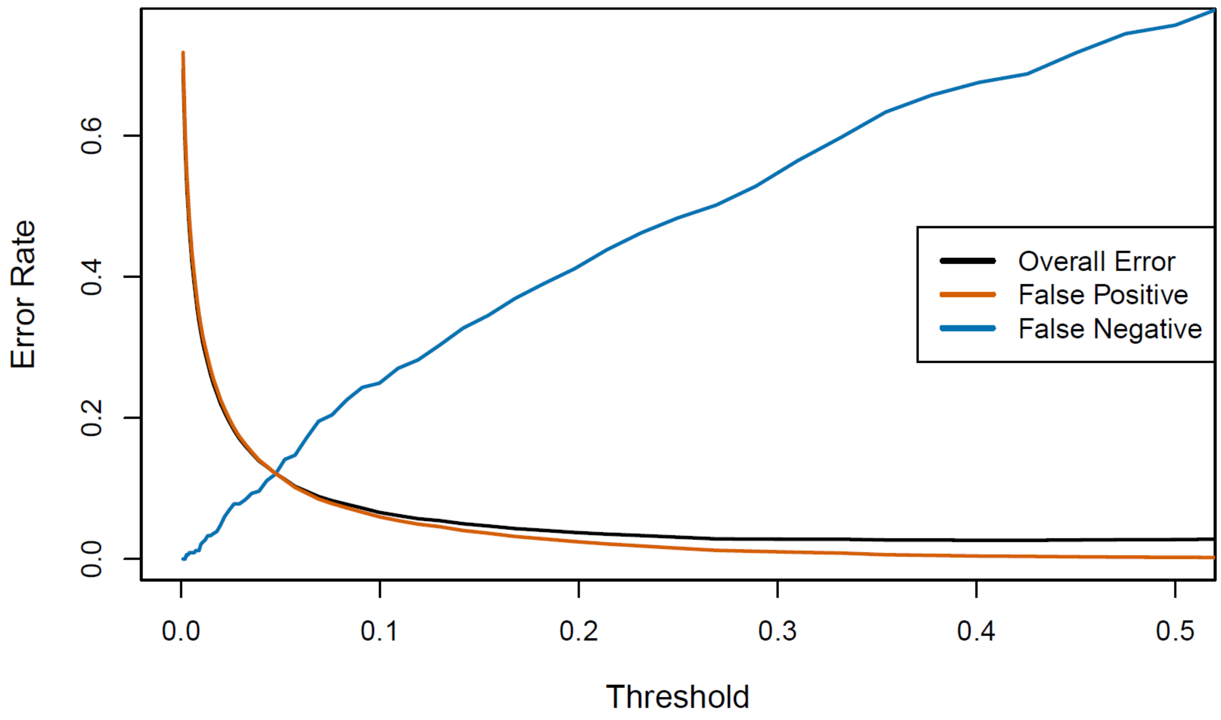
- Example: False positive and false negative rate are computed based on probabilities returned by classifier

$$P(\text{Class=True} \mid X_1, X_2, \ldots) \geq 0.5$$

- We can change the two error rates by changing the threshold from 0.5 to some other value in [0, 1]:

$$P(\text{Class=True} \mid X_1, X_2, \ldots.) \geq \text{threshold}$$

Varying the threshold

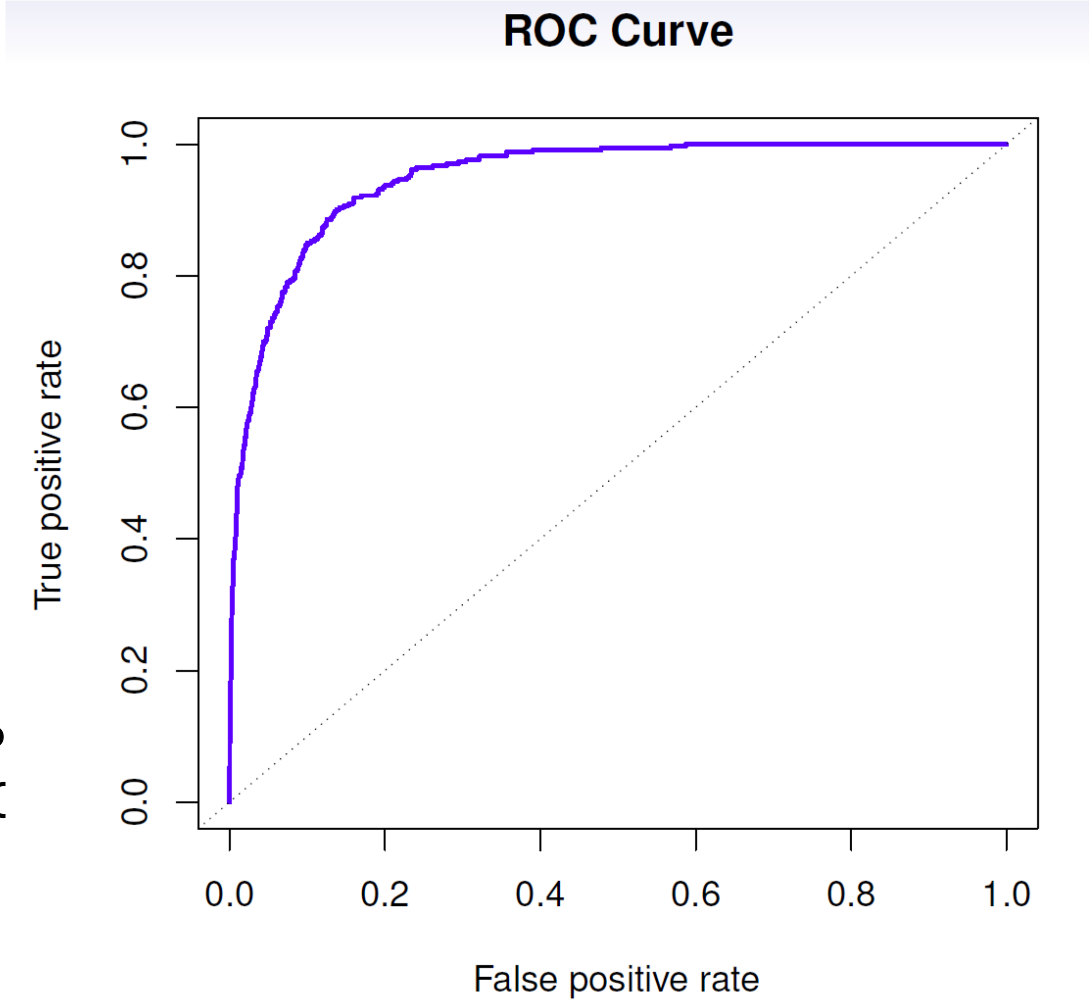


- To reduce false negative rate we would chose threshold other than 0.5, e.g., threshold ≤ 0.1

ROC curve

- ROC curve shows both TP rate and FP rate simultaneously

- To summarize overall performance we also use area under the ROC curve (AUC)
- The larger the AUC the better is the classifier. Why? What would be an ideal ROC curve?



Issues affecting model selection

- **Accuracy**
  - classifier accuracy: predicting class label
  - regression: MSE, MAE
- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency in disk-resident databases
- **Interpretability**
  - understanding and insight provided by the model
  - Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

Unsupervised feature selection

- criterion: preserve similarity between instances

- SPEC: spectral feature selection
- take instance similarity matrix and compute its eigenvectors and eigenvalues of graph Laplacian matrix L
- according to spectral clustering theories, the eigenvalues of *L* measure the separability of the components of the graph and the eigenvectors are the corresponding soft cluster indicators
- rank features according to their consistency with the graph structure
    - a feature that is *consistent* with the graph structure assigns similar values to instances that are near each other in the graph

Zhao Z, Liu H. Spectral feature selection for supervised and unsupervised learning. In Proceedings of ICML 2007, pp. 1151-1157.

Laplacian matrix

- Given a simple graph G with *n* vertices, its Laplacian matrix $L_{n \times n}$ is defined as:

  L = D − A

where D is the degree matrix and A is the adjacency matrix of the graph. A only contains 1s or 0s and its diagonal elements are all 0s. For D, a diagonal matrix, in the case of directed graphs, either the indegree or outdegree might be used, depending on the application.

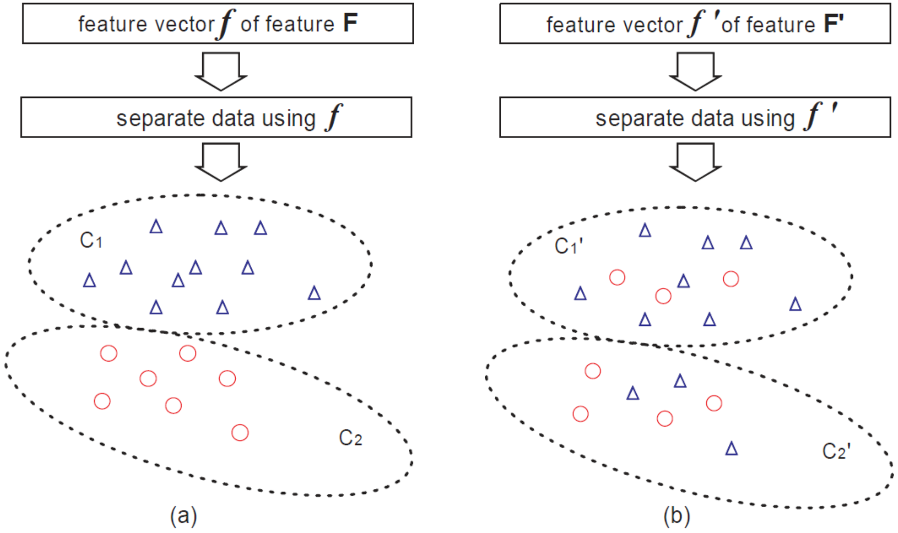
- The elements of L are given by

$$L_{i,j} \begin{Bmatrix} \deg(v_i) & ;\ i = j \\ -1 & ; i \neq j \text{ and } i \text{ is adjacent to } j \\ 0 & \text{otherwise} \end{Bmatrix}$$

- where $\deg(v_i)$ is the degree of the vertex i
- in feature selection,
- adjacency matrix is weighted by the distance between instances (and class membership)
- the degree serves as an estimation of density around instance (vertex) x

SPEC – spectral feature selection

- compute $f^T L f$ to measure how feature f is consistent with graph
- smaller values indicate better consistency
- both f and L have to be normalized in order not to affect the score



Unsupervised FS with clustering

- Nonnegative Discriminative Feature Selection (NDFS)
- perform spectral clustering to learn the cluster labels of the input samples
- simultaneously optimize for cluster labels and feature selection matrix

Li, Z., Yang, Y., Liu, J., Zhou, X. and Lu, H., 2012, Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of AAAI, v*ol. 2, pp. 1026-1032.

Semi-supervised feature selection

- typically a small sample of labelled and a large sample of unlabeled data is available
- principle: use the label information of labeled data and data distribution or local structure of both labeled and unlabeled data to evaluate feature relevance

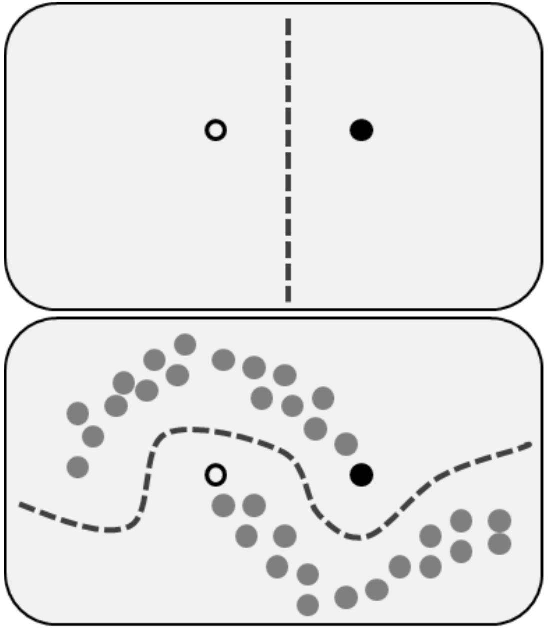


image by Techerin, Wikipedia

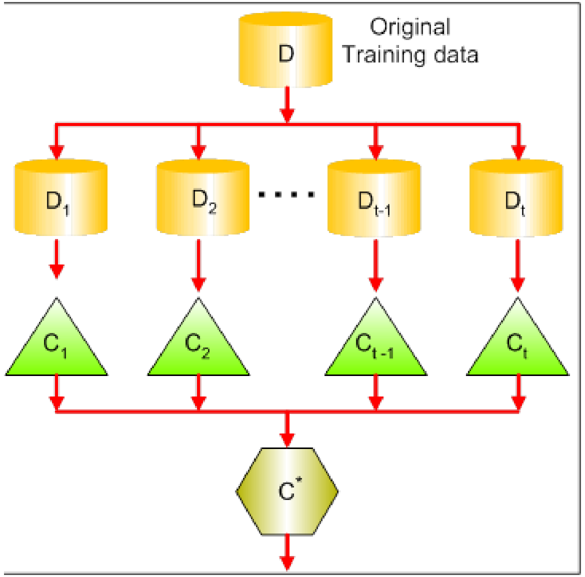
Laplacian score for semi-supervised feature selection

- Build two graphs:
  - W (within class): labeled instances are connected if from the same class, unlabeled instances are connected if near each other
  - B (between): labelled instances are connected if from different class
- proceed similarly as in unsupervised case, compute eigenvectors of Laplacian graph for W and optimize for soft cluster membership, use degree graph of B for normalization

Cheng, H., Deng, W., Fu, C., Wang, Y. and Qin, Z., 2011. Graph-based semi-supervised feature selection with application to automatic spam image identification. In Computer Science for Environmental Engineering and EcoInformatics (pp. 259-264).

Stability of feature selection

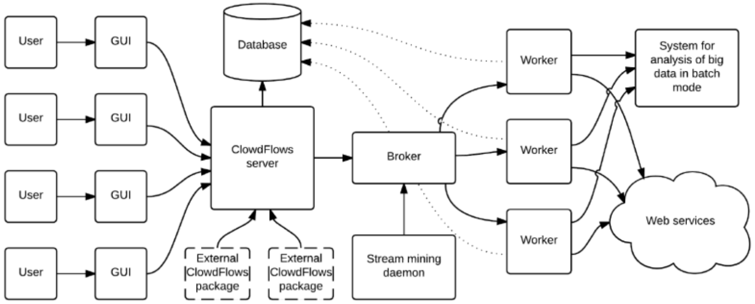
- for high dimensional small sample data stability of feature selection is a pressing issue, e.g., in microarray data we might get similar classification accuracy with different sets of features
- Solution: **ensemble approach:**
  1. produce diverse feature sets
     - different feature selection techniques,
     - instance-level perturbation
     - feature-level perturbation
     - stochasticity in the feature selector,
     - Bayesian model averaging
     - combinations of the above techniques
  2. aggregate them
     - weighted voting
     - counting



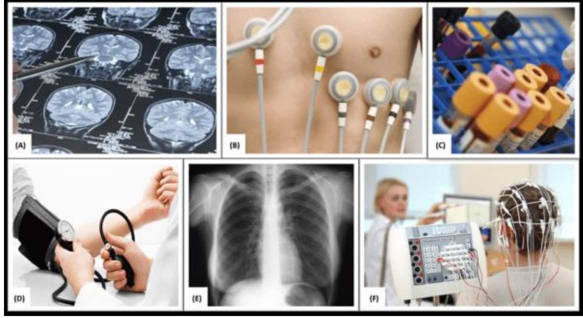
Big data issues

- distributed feature selection, e.g., use Statistical Query model in MapReduce architecture
- procedure
    1. decompose the feature selection process into summation forms over training samples,
    2. divide data and store data partitions on nodes of the cluster,
    3. compute local feature selection results in parallel on nodes of the cluster, and
    4. calculate the final feature selection result by integrating the local results.

Janez Kranjc, Roman Orač, Vid Podpečan, Nada Lavrač, Marko Robnik-Šikonja: ClowdFlows: online workflows for distributed big data mining. *FGCS*, 68:38-58, 2017



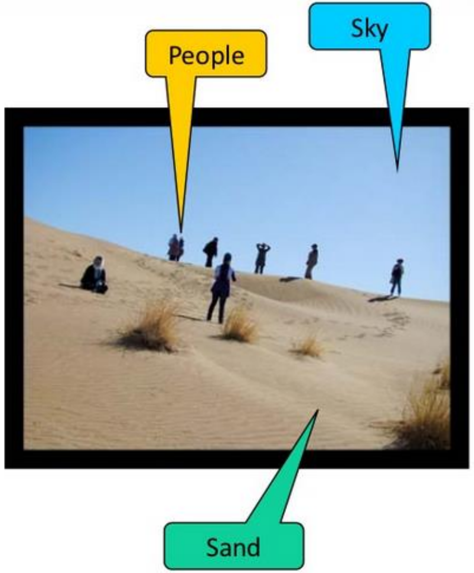
Multi-view learning



- information from different sources, e.g., measurements from a series of medical examinations for each subject, including clinical, imaging, immunologic, serologic, and cognitive measures.
- some measurements are irrelevant, noisy, or conflicting
- different views typically provide complementary information
- approaches:
  - baseline: concatenate all views
  - construct tensor space from views, preserve relations between views, use feature selection in tensor space
  - use ReliefF like approach, where different views contribute to the distances between objects
  - multi-view clustering and feature selection

Magajna T., Robnik-Šikonja M. (2018) Feature Selection for Multi-View Clustering (submitted)

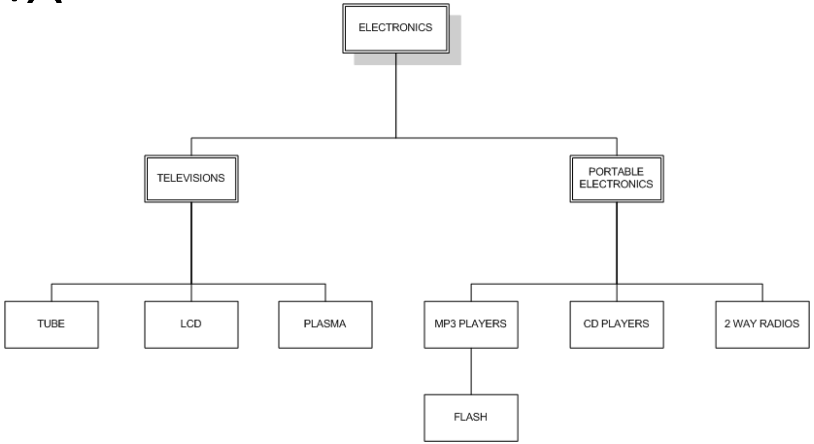
Multi-label learning



- each instance may have more than one label, e.g., purchased items, items on the picture
- approaches
  - transform to single label case (does not take correlations between labels into account)
    - select-max, select-min, select-random
    - copy all, copy weighted
    - label the power set
  - treat multiple labels directly e.g., binary relevance or via graph of correlations between labels
  - Relief like approaches:
    - using multi-label approach to difference of labels (hits, misses) by comparing sets of instance labels (similarly as RReliefF compares different values of response in regression)

Spolaôr, N., Cherman, E.A., Monard, M.C. and Lee, H.D., 2013. A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, *292*, pp.135-151.

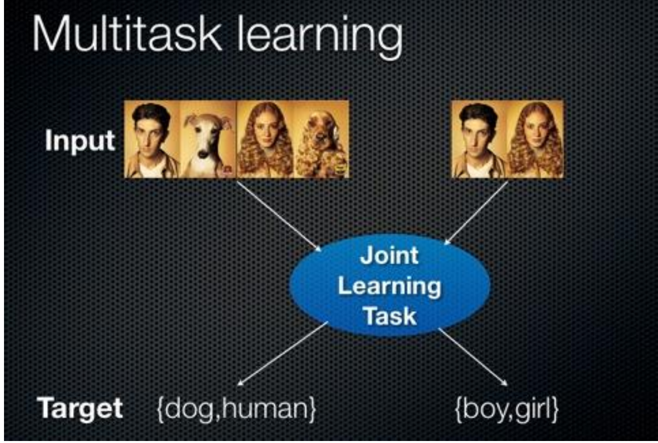
Hierarchical multi-label learning



- labels appear in hierarchies, e.g., biological processes or image labelling
- Relief like approach:
  - compute distances between two label as the distance in the hierarchy
  - using multi-label approach to difference of labels (hits, misses) by comparing sets of instance labels (similarly as RReliefF compares different values of response in regression)

Slavkov, I., Karcheska, J., Kocev, D., Džeroski, S. (2017) HMC-ReliefF: Feature ranking for hierarchical multi-label classification. Computer Science and Information Systems. 15. 43-43.

Multitask learning



- learn several related tasks simultaneously with the same model
- advantages:
  - the tasks share knowledge representation,
  - learning several related tasks prevents overfitting
- feature selection:
  - transform to several single tasks, aggregate the results
  - uses wrapper or embedded methods, e.g., multitask random forests where feature importance is estimated as the degradation of performance if feature values are randomly shuffled

Petković, M., Džeroski, S. and Kocev, D., 2017, October. Feature Ranking for Multi-target Regression with Tree Ensemble Methods. In *International Conference on Discovery Science*, pp. 171-185.

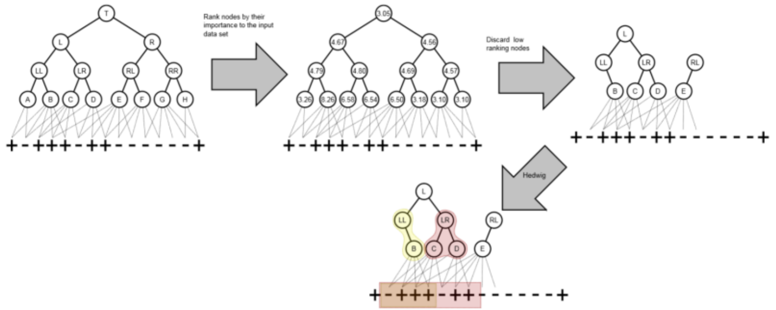
Online feature selection

- in data stream scenario (e.g., financial trading, environmental monitoring, industrial processes)
  - instances arrive sequentially, potentially the learned concept changes (concept-drift problem)
    - detect failure of classifier, reassess features, or
    - continuously asses features, measure their stability
  - new features may appear, e.g., new acronyms or hashtags on Twitter
    - asses new features, potentially replace some of the old chosen ones
  - both the above scenarios appear simultaneously

Wang, J., Zhao, P., Hoi, S.C. and Jin, R., 2014. Online feature selection and its applications. IEEE Transactions on Knowledge and Data Engineering, 26(3), pp.698-710.

Feature selection for graphs

- graphs are useful to represent relations, e.g., in biology
- Linked Open Data: huge graphs of relations for different areas, e.g. Bio2RDF
- GeneOntology – a hierarchical description of knowledge about genes
- the graphs are often embedded into a vector space to enable learning
- graph reduction techniques enable relational learning



J. Kralj, N. Lavrač and M. Robnik-Šikonja (2018)
NetSDM: Network pruning for semantic data mining (submitted)

Things we did not cover

- cost-sensitive feature evaluation
- privacy preserving feature selection
- adaptations of feature selection approaches for specific important domains: bionformatics, image analysis, NLP, graphs

Conclusions

- well researched area with many successful approaches
- but: abundance of data, new types of structured data, and many new learning scenarios offer lots of opportunities for new developments
- combinations of scenarios, e.g., multi-label feature selection in online scenario
- from feature rankings to feature subsets
- how to assure combinations of properties, e.g., scalability, stability, and security