Computer Vision
(CSE 40535 / 60535)

Feature selection

[Part III: Pattern recognition]

Adam Czajka
Department of Computer Science and Engineering
University of Notre Dame, USA
Fall 2019

Version: Oct/29/2019

Why we need feature selection?

1. Features have different discrimination power
    - some of them may poorly contribute to class separation
2. Features are correlated
    - we do not need redundant information
3. Smaller feature sets are practical
    - dimensionality reduction
    - simpler/faster classification
    - low memory usage

Two approaches

1. Feature subset selection
    - Select the subset of *M* features (among *N*) maximizing some aim function (e.g., the classification accuracy)
    - Classification based only on the selected *M* features

original
N features

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \mapsto \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iM} \end{bmatrix}$$

selected
M features

Two approaches

2. Feature space transformation

- Find a new coordinate system in a feature space (typically with lower number of dimensions)
- Make projection of all samples onto new coordinate system
- Classify in this new coordinate system

original
N features

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \mapsto \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_M \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}\right)$$

new (transformed) *M* features

transformation

Two approaches

Note: if transformation *f* is a sparse *M* x *N* projection matrix, feature space transformation becomes feature subset selection:

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iM} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}$$

new (transformed)
M features

transformation

original
N features

1. Feature subset selection

Advantages

1. Features may be expensive to calculate
   - select only small subset for final classification

2. Good interpretation of the results
   - design of classification rules
   - optimization of feature calculation methodology
     (e.g., selection of optimal parameters in Gabor filtering)

3. Useful for non-numeric features

Requirements

Search strategy to select candidates

1. Exponential algorithms
   - the number of evaluations grows exponentially with number of input space dimensions
   - examples: exhaustive search, branch & bound
2. Sequential algorithms
   - add/remove features sequentially
   - prone to be stalled in local minima
   - examples: sequential forward/backward selection, bidirectional search
3. Randomized algorithms
   - better exploration of the feature space in limited time
   - may "escape" from local minima
   - examples: simulated annealing, genetic algorithms

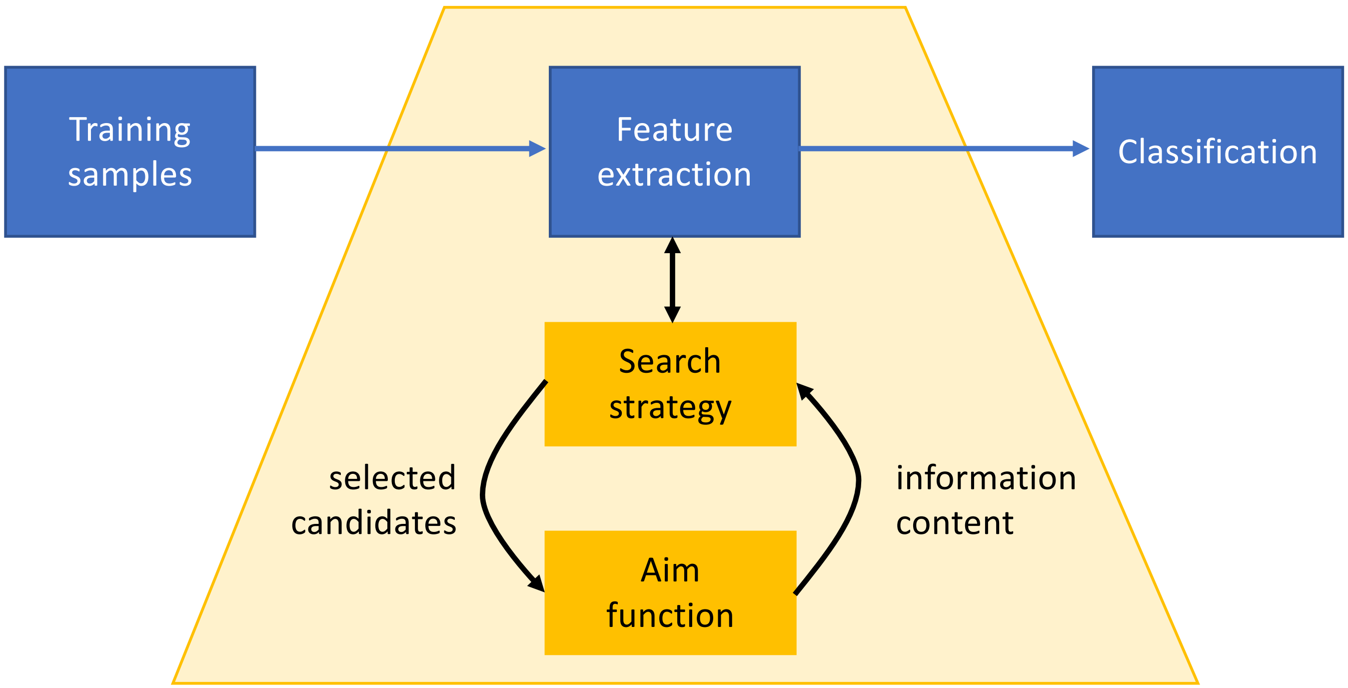
Requirements

Aim function to evaluate ”goodness” of candidates

1. Filters
   - evaluate candidates without engaging the classifier
   - use different measures of information content (statistical dependence, statistical correlation, mutual information, class separation, etc.)

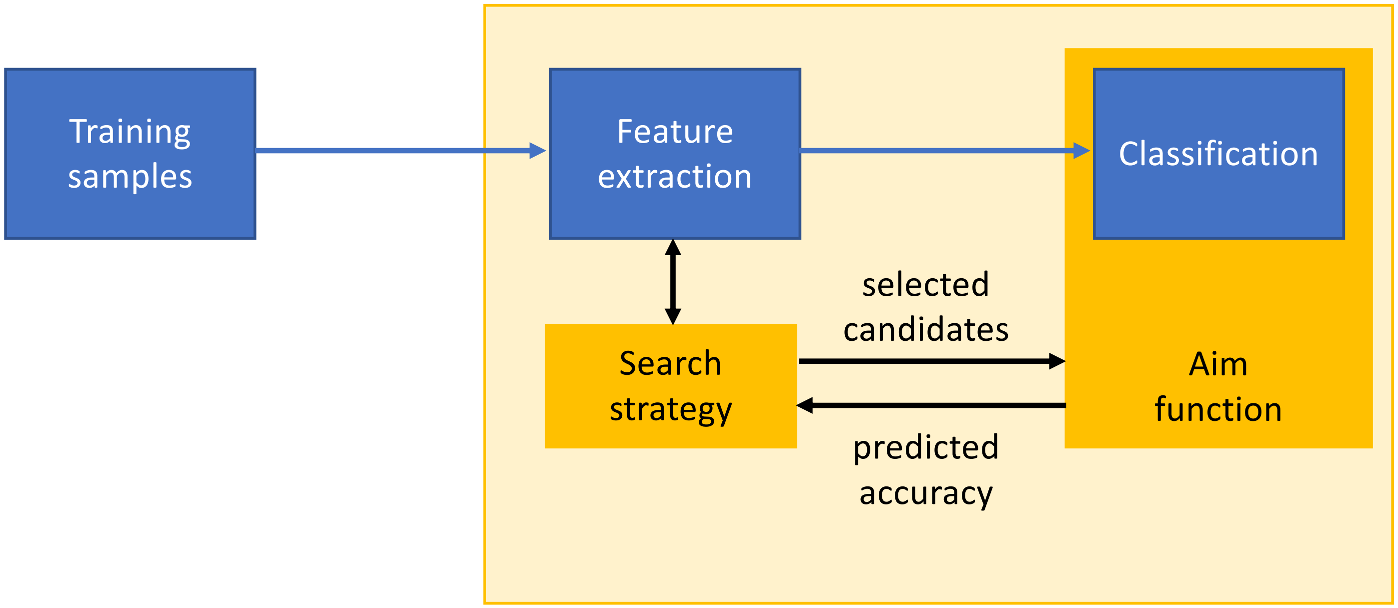
2. Wrappers
   - use classifier to predict a classification accuracy for selected candidates

Filters ...



Training samples
Feature extraction
Classification
Search strategy
selected candidates
information content
Aim function

… vs Wrappers


Training
samples
Feature
extraction
Classification
Search
strategy
selected
candidates
predicted
accuracy
Aim
function

Filter types

1. Separability
   - we want relevant features
   - distance between classes
     (different classes should be far enough to each other)
   - Fisher ratio: within-to-between variability
     (different classes should be far enough to each other
     AND each class should be as concise as possible)
2. Correlation
   - we do not need redundancies
3. Mutual information / statistical dependence
   - nonlinear estimate of mutual information between random variables

Filter types

Mutual information

- Mutual information between two random variables *x* and *y:*

joint *x* and *y*
probability distributions

$$MI(x,y) = \int_x \int_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy$$

marginal *x* and *y*
probability distributions

Filter types

Mutual information

- Feature relevance:

class assignment

$$f_i^{(rel)} = MI(f_i, c)$$

- Feature redundancy:

$$f_i^{(red)} = \frac{1}{K} \sum_{k=1}^{K} MI(f_i, f_k)$$

features already selected

Filter types

Mutual information

- Simultaneous selection of relevant and not redundant features (mRMR: **m**inimum **R**edundancy, **M**aximum **R**elevance)

$$min_i\left(f_i^{(red)} - f_i^{(rel)}\right)$$

Further reading on mRMR method:
Peng, *et al.*: "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min- redundancy," *IEEE TPAMI,* 27(8), 1226–1238, 2005

Search strategies

Sequential Forward Selection (SFS)

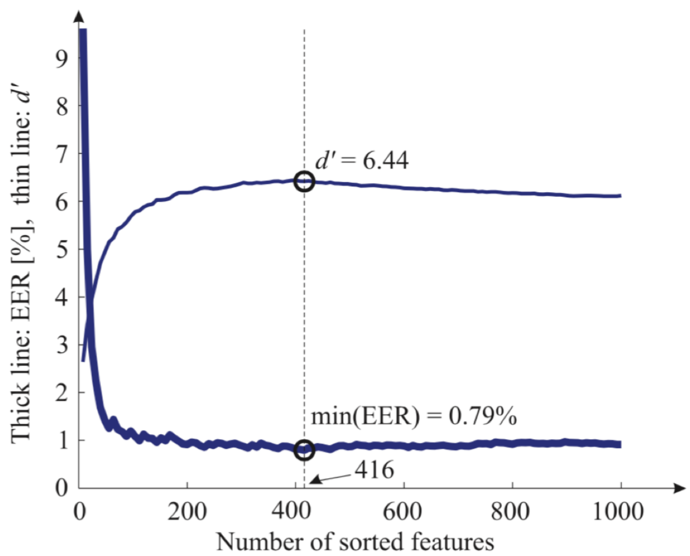
1. Start with an empty set of features
2. Sequentially add features f_i that increase collectively with already selected features the discrimination strength (or classification accuracy)

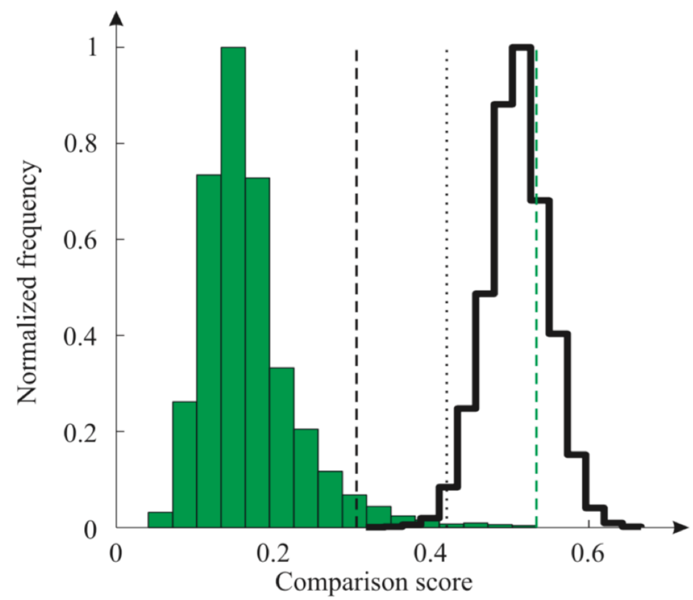
3. Notes
   - SFS performs best when the optimal subset is small
   - Disadvantage: unable to remove features that become obsolete after addition of other features

Search strategies

Sequential Forward Selection (SFS)

Example: the best iris features among ~100k candidates





Source: Adam Czajka, Krzysztof Piech, "Secure Biometric Verification Station Based on Iris Recognition", Journal of Telecommunications and Information Technology (JTIT), Vol. 3, pp. 40-49, 2012

Search strategies

Sequential Backward Selection (SBS)

1. Start with a full set of features
2. Sequentially remove features f_i that do not deteriorate the discrimination strength (or classification accuracy)

3. Notes
   - SBS works best when the optimal feature subset is large
   - Disadvantage: inability to reevaluate the usefulness of a feature after it has been discarded

Search strategies

Plus-L minus-R Selection (LRS)

1. Generalization SFS and SBS
2. For L > R, LRS starts from the empty set
   and iteratively adds L / removes R features
3. For L < R, LRS starts from the full set
   and iteratively removes R / adds L features
4. Notes
   - LRS tries to compensate for lack of feature reevaluation in SFS and SBS
   - Optimal values of L and R are hard to be calculated; floating L and R values based on increase in classification accuracy (SFFS and SFBS methods)

Search strategies
Bidirectional Search (BDS)

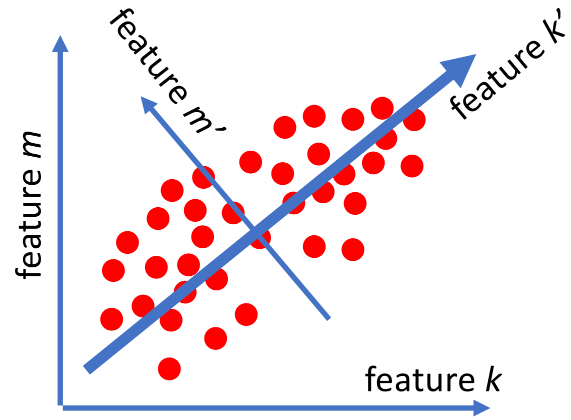
1. SFS and SBS are run alternately
    - SFS starts from an empty set
    - SBF starts from a full set

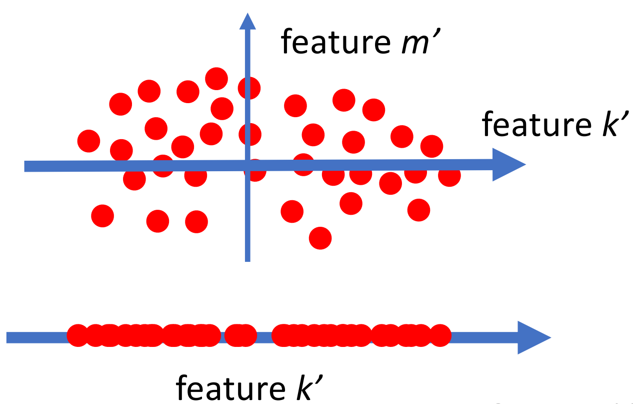
2. SFS and SBF converge to the same solution if:
    - features removed by SBS are not added by SFS
    - features added by SFS are not removed by SBS

2. Feature space transformation

Principal Component Analysis (PCA)

1. Find a new coordinate system in which features will be linearly uncorrelated
2. Principal component: eigenvector of the covariance matrix with the largest eigenvalue

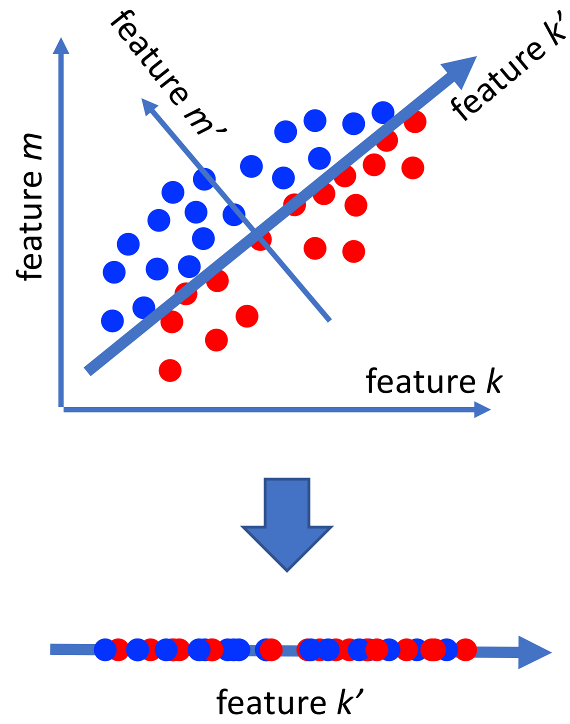




3. Use only *p* first principal components in classification:
   - calculate all features
   - make a projection on new coordinate system

Principal Component Analysis (PCA)

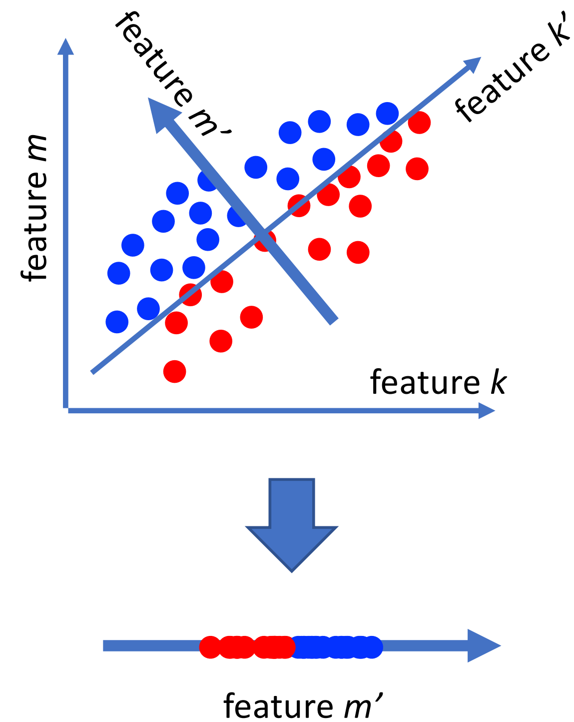
PCA ignores class labels and finds directions with the largest variance



Linear Discriminant Analysis (LDA)

LDA considers both within-class and between-class variance (Fisher ratio) to maximize class separation

LDA is often preceded by PCA to reduce feature space dimensionality



Lecture wrap-up

1. We do not need features that are not relevant (do not predict our class) and are redundant (other features bring the same information)
2. Fewer features = simpler classification
3. Two approaches to feature selection
   - subset of features (no feature space transformation)
   - transformations of the feature space