Classification

Anders Brun
Centre for Image Analysis

Uppsala University
Outline

- An overview on classification
- Basics of classification
- How to choose appropriate features (feature set)
- How to perform classification
- Classification methods
Where are we right now?

**Figure:** Classification in image processing.
What is classification?

- Distinguish different types of objects, situations, events
- We would like to create an intelligent system that can draw conclusions from our image data
- Determine which parts (pixels) of the image belong to the object of interest
- No classification (pattern recognition) is possible without some kind of knowledge
- Often used as the last stage in automated image analysis task
Some important concepts

- Descriptors are called features (feature vectors) $\mathbf{x} = (x_1, \ldots, x_d)$
- Classification is grouping of patterns (samples) into classes
- Patterns are placed in classes of objects that share common properties
Classification example 1

- How to separate apples from pears?
- Separate pedestrians from cars in a surveillance camera?
- Any other examples?
Classification example 2

Figure: Histological bone implant images.
What are good features?

- Features are the individual measurable heuristic properties of the phenomena
- Discriminating (effective) features
- Ideally independent features
- Features:
  - area, perimeter
  - texture
  - color
  - ...
What are good features?

- Each pattern is represented in terms of $d$ features $\mathbf{x} = (x_1, \ldots, x_d)$
- The goal is to choose those features that allow pattern vectors belonging to different classes to occupy compact and disjoint regions
- It is application dependent
- You might try many, many features, until you find the right ones
- Often, people compute 100s of features, and put them all in a classifier
  - ”The classifier will figure out which ones are good”
  - This is typically wrong!
Peaking phenomenon (Curse of dimensionality)

- Additional features may actually degrade the performance of a classifier.
- This paradox is called peaking phenomenon (curse of dimensionality).
- For a finite set of training data, the error will increase when we add too many features.

**Figure**: Curse of dimensionality.
Feature set 1

- Colour?
- Area?
- Perimeter?
- ...

[Images of plane, sun, ship, octopus, two silhouettes]
Feature set 2

- Colour?
- Area?
- Perimeter?
- ...

[Image of a set of coins of varying sizes and colors]
Feature set 3

- Colour?
- Area?
- Perimeter?
- ...

![Feature set 3 diagram](image-url)
Dimensionality reduction

- Keep the number of features as small as possible
  - Measurements cost
  - Accuracy
- Simplify pattern representation
- The resulting classifier will be faster and will use less memory
- On the other hand, a reduction in the number of features may lead to a loss in the discriminatory power and thereby lower the accuracy
Feature extraction

- Feature extraction is the process of generating features to be used in the selection and classification tasks.
- Reduce dimensionality by (linear or non-linear) projection of $D$–dimensional vector onto $d$–dimensional vector ($d < D$).
- Linear case: $y = Ax$, where $A$ is $m \times d$ and $m \leq d$.
- May have not clear physical meaning.

Figure: Feature extraction.
PCA?

Figure: Feature extraction.
Independent Component Analysis (ICA)
Feature selection

- Feature selection methods choose features from the original set based on some criteria
- Reduce dimensionality by selecting subset of original features
- Have clear physical meaning

**Figure**: Feature selection.
Choice of a criterion function

- The main issues in dimensionality reduction is the choice of a criterion function
- A commonly used is the classification error of a feature subset
Feature selection methods

- Exhaustive search
- Best individual features
- Sequential forward selection
- Sequential backward selection
Exhaustive search

- Evaluate all $\binom{d}{m}$ possible subsets
- Guaranteed to find the optimal subset
- Very expensive!
- Example: If $d = 50$, $m = 5$ then there are 2 118 760 possible subset (classifications) to evaluate
Best individual features

- Evaluate and rank all $d$ features individually
- Select the best $l \leq d$ individual features
- Simple and not likely to lead to an optimal subset
- Example:
  - Feature 1 is best
  - Feature 2 is second best
  - Maybe features 3 and 4 in combination outperform features 1 and 2!
Sequential forward selection

- Selects the best feature and then add one feature at a time
- Once a feature is retained, it cannot be discarded
- Computationally fast
Sequential backward selection

- Starts with all $d$ features and successive delete one feature at a time
- Once a feature is deleted, it cannot be brought back into the optimal subset
Search algorithms

- The presented algorithms are suboptimal
- When should we use forward selection?
- When should we use backward selection?
- Are there any better algorithms?
  - Branch and bound (divide and conquer method)
  - Genetic algorithms
  - Simulated annealing
Supervised vs. Unsupervised classification

- **Supervised**
  - First apply knowledge, then classify
- **Unsupervised**
  - First classify, then apply knowledge

**Figure:** Supervised and unsupervised classification.
Supervised classification

- Training mode - find the appropriate features for representing the patterns, and a classifier is trained to partition the feature space.
- Classification mode - the trained classifier assigns the input pattern to one of the pattern classes under consideration based on measured features.
- A given pattern $x$ is to be assigned to one of the $k$ classes $C_1, C_2, ..., C_k$ based on the vector of $d$ feature values $x = (x_1, ..., x_d)$.
- $p(x|\omega_i)$ pattern vector $x$ belong to class $\omega_i$ (conditional probability distribution).
- Decision rules: Bayes decision, maximum-likelihood, ...
Decision boundary

Figure: Decision boundary.
Linear and quadratic classifier

Figure: Linear (left) and quadratic (right) classifier.
Training data

- Training data can be obtained from available training samples
- Later, we classify patterns which are not used during the training stage
- The performance of classifier depends on the number of available training samples as well as the specific values of the samples
- Number of training samples should not be too small!
Training data

- Conditional densities, $P(\mathbf{x}|C_i)$ are unknown and they have to be learned from available training data.
- Maybe the class of the conditional density is known (for example, multivariate Gaussian).
- A common strategy is to replace unknown parameters by their estimated values (e.g., mean and covariance matrix).
- Less available information - the difficulty of classification problem increases.
How to choose appropriate training set?

Figure: Original image and training image.
How to choose appropriate training set?

**Figure:** Original image and training image.
The performance of classifier depends on the interrelationship between sample size, number of features and classifier complexity.

The choice of a classifier is a difficult problem!

Unfortunately, it is often based on which classifier(s) happen to be be available. Or best known to the user.
Some commonly used classifiers

- The user can modify several associated parameters and criterion function
- There exist some classification problem for which they are the best choice
- Three examples of classifiers (read more about them in the book!):
  - K-nearest neighbour classifier
  - Support vector machines
  - Linear and quadratic discriminant analysis
K-Nearest neighbour

- Stores all training samples
- Assigns pattern to majority class among $k$ nearest neighbour
$k$—nearest neighbour

- Assigns pattern to majority class among $k$ nearest neighbour
$k$—nearest neighbour

- Metric dependent (scale a feature $\times 10$ you may get a different result)
- Sometimes called “Lazy learning”, because we just keep training data
- The function is approximated only locally
- The best choice of $k$ depends on data
- Larger values of $k$ reduce effects of noise, but make boundaries between the classes less distinct
Support vector machine

- Linear classifier
- Based only on samples close to boundary (the support vectors)
- In some sense, the samples closes to the decision boundary are the samples that are hardes to classify.
Support vector machine

- It is primarily a two class classifier
- Maximize the width of the margin (empty area) between the classes
Support vector machine

- Only the “support vectors” define the classification function.
- This is a simplified explanation of SVMs, often a “soft margin” is used, i.e. allowing some samples also inside the margin.
- Often SVMs are used in combination with so called kernels, which allow them to produce non-linear decision boundaries.
Discriminant analysis

- Classes are normally distributed
- Covariance matrices $\Sigma_1, \Sigma_2$ for classes $C_1$ and $C_2$, respectively
- Mean values $m_1, m_2$ for classes $C_1$ and $C_2$, respectively
- Prediction that point $x$ belong to the second class $C_2$

$$
(x - m_1)^T \Sigma_1^{-1} (x - m_1) + \ln |\Sigma_1| - (x - m_2)^T \Sigma_2^{-1} (x - m_2) - \ln |\Sigma_2| < T
$$

- Threshold $T$?
Discriminant analysis

- Simple and fast
- Linear Discriminant Analysis (LDA)
  - Covariance matrices of the considered classes are equal, i.e., $\Sigma_1 = \Sigma_2$
- Quadratic Discriminant Analysis (QDA)
  - Covariance matrices of the considered classes are not equal, i.e., $\Sigma_1 \neq \Sigma_2$
Linear and Quadratic discriminant analysis

- LDA can only separate the classes with straight lines (in two dimensions) or, more generally, (hyper)-planes in higher dimensions.
- QDA is more sophisticated since it can separate classes with circles, lines, ellipses, parabolas or hyperbolas.

Figure: LDA (left) and QDA (right).
Bayes Classifiers

- LDA is a special case of a Bayes classifier
- In a Bayes classifier, the probability of a sample belonging to class $C_k$ is $p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}$ and the decision is the class with maximal probability.
Bayes Classifiers

- The difficulty is to estimate the probability functions
- LDA and LQA are special cases where the probability functions are Gaussian, estimated using the mean and covariances.
- In the Naive Bayes classifier, the features are assumed to be independent, which leads to a simplified formulation,

\[
p(C_k|x) = \frac{p(C_k) \prod_i p(x_i|C_k)}{p(x)}.
\]
Combine classifiers!

- Each of the classifiers can be developed in a different context and for entirely different representation/description (identification of persons by their voice, face as well as hand writing)
- More than a single training set is available
Combine classifiers!

- Different classifiers trained on the same data may only differ in their global performances.
- May show strong local differences.
- Combination is very useful if the individual classifiers are independent.
Combination

- Parallel (ensemble)
  - All classifiers are evaluated independently
  - Their results are combined (first weight then combine!)

- Serial - linear sequence
  - The number of possible classes for a given pattern are gradually reduced as more classifiers in the sequence has be involved
  - First consider simple classifiers (low computational and measurements cost)
  - More accurate and expensive classifiers apply after

- Hierarchical (tree-like)
  - Individual classifiers are combined into a structure (a tree-like structure)
  - Advantage high efficiently and flexibility
Unsupervised classification

- Difficult, expensive or even impossible to rely label training sample with its true category (It is not possible to obtain ground truth)
- Patterns within a cluster are more similar to each other than are patterns belonging to different clusters
Unsupervised classification

- Difficult
- How to determine the number of clusters $K$?
- The number of clusters often depends on resolution (fine vs. coarse)
$K$—means

- Usually, the number of clusters $K$ is predetermined
- Often starts with a random partition of data into classes
- Tries to minimize some type of error criterion
  $C = \{C_1, \ldots, C_k\}$ classes, $d$ distance measure

\[
\arg \min_{C,m_i} \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j - m_i)
\]

- We can iterate the minimization of the partition $C$ and the cluster centers $m_i$ in a “ping pong way”
- Stops when some stopping criteria is satisfied (the cluster are stabilized)
$K-$means

- **Step 1.** Select an initial partition with $K$ clusters. Repeat steps 2 through 4 until the cluster membership stabilizes.
- **Step 2.** Generate a new partition by assigning each pattern to its closest cluster center.
- **Step 3.** Compute new cluster centres as the centroids of the clusters.
- **Step 4.** Repeat steps 2 and 3 until an optimum value of the criterion function is found.
$K \text{ - means}$
$K$—means
$K$—means - disadvantages

- Different initialization can result in different final clusters
- Difficult to predict what $K$ should be
- It is helpful to rerun the classification using the same as well as different $K$ values, to compare the achieved results
- 10 different initializations for 2D data
- For $N$-dimensional data 10 different initializations is often not enough!
How to determine $K$?

- Davies-Bouldin index (DB index)
- It works for spherical clusters
- Intra-cluster and inter-cluster measure

![Graph showing DB index vs. Number of clusters](attachment:graph.png)
Question 1

The following eight points $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$ should be classify into three clusters using $K$-means clustering. Initial cluster centres are: $A_1(2,10)$, $A_4(5,8)$ and $A_7(1,2)$.

Find the three cluster centres after the first iteration. The distance function between two points $A(x_a, y_a)$ and $B = (x_b, y_b)$ is defined as $d(A, B) = |x_a - x_b| + |y_a - y_b|$.
Fuzzy $K$–means (fuzzy $C$–means)

- In the hard clustering process, each data sample is assigned to only one cluster.
- In practise there are many cases in which the clusters are not completely disjointed.
- Membership values are assign for all points for each of the clusters.
Hierarchical clustering

- Hierarchical clustering - construct clustering tree (dendrogram)
- Start with each object/pixel as its own class
- Merge the classes that are closest according to some distance measure
- Continue until only one class is achieved
- Decide the number of classes based on the distances in the tree
Simple dendrogram
Distances measure and linking rules

- Distance measures:
  - Euclidean
  - City block
  - Mahalonobis

- Linking rules - how distances are measured when a cluster contains more than one object
  - Single linkage (nearest neighbours)
  - Complete linkage (furthest neighbours)
  - Mean linkage (distance between cluster centres)
Linking rules

- Dependent on metric
- $d_{complete}$ - red
- $d_{single}$ - blue
The following seven points are given: \( A_1(1, 8), A_2(1, 6), A_3(6, 8), A_4(5, 3) \\ A_5(7, 1) A_6(13, 4) A_7(13, 3) \). Apply hierarchical clustering assuming single linkage and city block distance. Construct the dendrogram.
Good features = Easy classification

- “Given the right representation, classification is easy”
- The more feature dimensions we have, the more likely it is that our classes are separable by a simple linear classifier
- It even helps adding feature dimensions as non-linear combination of existing features
Semi-Supervised classification

- Semi-Supervised
  - If you have only a few labeled training data points, and a lot of unlabeled training data points. These algorithms can take advantage of the geometry of the unlabeled data in order to create a less naive classifier to separate the labeled data. Below, the intrinsic 1-D parameterization of the spiral is perhaps the most natural space to create a classifier in, and not the 2-D embedding space.

Figure: Semi-supervised classification.
Semi-Supervised classification

- From Robust and Scalable Graph-Based Semisupervised Learning. Liu, Wei and Wang, Jun and Chang, Shih-Fu In Proceedings of the IEEE, vol 100, no 9
Summary and conclusions

- Classification is needed in image processing
- It is highly application dependent
- Features and classifiers
- Using more features does not guarantee a better result!