1. Overview

- complete data mining process:
  - (planning)
  - (preparation)
  - preprocessing
  - learning
  - evaluation
  - (analysis)
- MIR (= music information retrieval)
- basic question: what is similarity?
- genre classification based on spectral similarity:
  - database of songs
  - MFCC (= mel frequency ceptstrum coefficient)
  - GMM (= gaussian mixture model) per song
  - log-likelihood of songs given GMMs
  - distance matrix
  - nearest neighbour classification

2. The very basics of Statistical Pattern Recognition

- rote learning, lookup-table:
  - store every possible image with class label
  - problem: 256x256 pixel, 8bit/pixel -> 10^158000 images
- generalization: classifiers must classify previously unseen image vectors
- preprocessing: combine large number of input variables to create features
- use thresholds for minimization of misclassification
- classification:
  - vector x represents image
  - variable y represents result
  - mathematical function with adjustable parameter w: $y_k = y_k(x, w)$
  - concept space: all possible distinct functions
  - language bias: restrict the set of all concepts
  - search bias: prefer some functions to others
- regression: continuous variables represent output
- classification and regression are particular cases of function approximation
- polynomial curve fitting (way of function approximation):
  - high bias:
    - little flexibility
    - low complexity
    - under-fitting
  - high variance:
    - too much flexibility
    - high complexity
    - over-fitting
- use different training and test sets

3. Bayes and all that!

- posterior probability $P(C_k|X^1)$
- likelihood (class conditional probability) $P(X^1|C_k)$
- prior probability $P(C_k)$
- joint probability $P(C_k, X^1)$
- $P(C_k, X^1) = P(X^1|C_k)\,P(C_k)$
- $P(C_k|X^1) = \frac{P(X^1|C_k)\,P(C_k)}{P(X^1)}$
- assign image to class with largest posterior (minimize misclassification)
- sometimes priors in training data are not representative (e.g. x-ray images + cancer)
- non-bayesian approaches are often just approximations (easier to estimate but not optimal)

4. Probability Density Estimation

- parametric methods:
  - ◇ assume specific form of density model (gaussian or normal distribution)
  - ◇ parameters optimized using best fit to data
  - ◇ estimate mean and covariance matrix by using maximum likelihood
  - ◇ minimizing negative log likelihood is more convenient
  - ◇ pro: easy to evaluate
  - ◇ con: tied to specific functional form
- non-parametric methods:
  - ◇ form of density determined entirely by data
  - ◇ kernel based methods: data point x, fixed volume V, estimate points falling in V
  - ◇ nearest neighbour methods: data point x, number of nearest neighbours K, estimate volume V which contains K nearest neighbours
  - ◇ pro: general form (driven by data)
  - ◇ con: number of variables grows with size of data
- semi-parametric methods:
  - ◇ “best of both worlds”
  - ◇ GMM iterative procedure:
    - □ make initial guess of GMM parameters
    - □ use old values -> evaluate right sides -> new values -> smaller error on E ->
    - □ replace old by new values
    - □ E-step: evaluate posteriors for all components
    - □ M-step: evaluate means, variances and mixing posteriors
- preprocessing to calculate MFCCs:
  - ◇ convert to frames
  - ◇ discrete fourier transform
  - ◇ log of amplitude spectrum
  - ◇ mel-scaling and smoothing
  - ◇ discrete cosine transform
- modelling of songs using GMMs
  - ◇ for each song train a GMM using EM (input: vectors of MFCCs)
  - ◇ computation of similarity: for every combination of songs and models compute negative log-likelihood
  - ◇ K-nearest neighbour classification:
    - □ for new data x find K nearest neighbours

        - assign majority class of K to x
        - draw hypersphere which contains K points around x
        - nearest neighbour rule directly estimates posteriors
        - one nearest neighbour error is maximal twice that of the bayes optimal classifier
- artist vs. genre classification: use artist filter to make sure that all songs of an artist are either in the training or test set

5. Advanced Classification

- novelty/outlier detection: identification of new/unknown data that a machine learning system is not aware of during training
- ratio reject: $reject\ X\ if\ \rho(X) > E[\rho(X^{tr})] + s * std(\rho(X^{tr}))$
- doubt levels: if max(posterior) > doubt_level => classify, else don't classify
- KNN-reject:
    - all K neighbours must agree
    - qualified majority of neighbours must agree

6. Statistical Evaluation of Machine Learning Experiments

- K-fold cross-validation:
    - divide into K equal sized parts
    - each part used as a test for classifier trained on all other parts
    - each part is used for testing exactly once
    - computationally costly
    - less random influences
    - stratified CV:
        - ensure the same class distribution in each fold as in the full training data
        - reduces variance across folds
    - leave-one-out CV:
        - each fold contains only a single example
        - unbiased error estimate
        - possibly high variance
        - computationally very costly

7. Unsupervised Learning I: Visualisation

- goals of unsupervised learning:
    - find useful representation of data
    - find clusters
    - dimensionality reduction
    - find hidden causes/sources of data
    - model data density
    - find patterns
- goal of visualisation: find low dimensional projection of high dimensional data that captures most correlation
- PCA (= principal component analysis):
    - transform vectors x linearly to uncorrelated (= orthogonal) vectors by finding a new basis of the input space
    - first new vector should explain most of the variance in data
    - second should explain remaining, etc.

- multi dimensional scaling:
  - ◇ lower dimensional projection in which points that are close to each other in the high-dimensional input space are also close in the low-dimensional output space
  - ◇ Sammon mapping (“chain link”)
- ICA (= independent component analysis):
  - ◇ finds hidden causes/sources in data
  - ◇ blind separation of sources
  - ◇ cocktail party problem
  - ◇ 2 variables are uncorrelated if their covariance is 0
  - ◇ if 2 variables are independent they are also uncorrelated, but not vice versa
  - ◇ not more than 1 gaussian source allowed
  - ◇ ICA used for artefact removal in EEGs

8. Unsupervised Learning II: Clustering

- divide observations into groups so that members of a group are alike
- mapping that assigns each input vector a reproduction (codebook) vector (out of a finite alphabet)
- K-means clustering:
  - ◇ start with initial codebook
  - ◇ partition data according to codebook
  - ◇ update codebook
- partitioning methods divide data into number of groups
- hierarchical methods produce trees of clusters:
  - ◇ agglomerative algorithms: start with each point as cluster and merge clusters
  - ◇ divisive algorithms: start with one big cluster and divide it iteratively
- HMM (= hidden markov model)
  - ◇ models locally stable probability densities (using GMMs) and the according transition probabilities between these states
  - ◇ markov chain property: probability of next state only depends on previous state
  - ◇ evaluation problem: given HMM M, observation sequence O, calculate probability that M has generated O
  - ◇ decoding problem: given HMM M, observation sequence O, calculate most likely sequence of hidden states that produced O
  - ◇ learning problem: given observation sequences O, general structure of HMM, determine parameters M of HMM that fit training data best
  - ◇ better describe spectral similarity of songs
  - ◇ advantage not visible when doing genre classification based on spectral similarity

9. Supervised Learning

- entropy: amount of impurity/randomness
- infogain: expected reduction of entropy
- constructing decision tree:
  - ◇ constructed in a top-down recursive divide-and-conquer manner
  - ◇ at the beginning all training samples are at the root
  - ◇ attributes are categorical (if continuous, discretise them)
  - ◇ test attributes are selected on the basis of a statistical measure
  - ◇ examples partitioned recursively based on selected attributes
- stopping construction of decision tree:

- all samples of a node belong to the same class
- no remaining attributes
- no samples left

- overfitting decision tree:
  - too many branches
  - may reflect anomalies due to noise or outliers
  - prepruning: stop tree construction early
  - postpruning: remove branches (use holdout data to chose best pruned tree)
- SVM (= support vector machine):
  - points on separating hyperplane are the support vectors
  - SVMs pick best separating hyperplane according to the maximum margin criterion
  - Lagrange formulation:
    - constraints replaced by lagrangian multipliers
    - training data will only occur as dot products
  - kernel trick: if boundary is not linear use function to map data into another space where it can be linearly separated
  - only few support vectors needed for training
  - no overfitting despite high dimensionality
- Multi-Layer Perceptron:
  - learns non-linear mapping from input to output
  - learns non-linear decision boundaries