Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

CAS Practical Machine Learning
Introduction

Unsupervised Learning

Prof. Dr. Jürgen Vogel (juergen.vogel@bfh.ch)

Unsupervised Learning

an algorithm learns from experience `E` to solve some tasks `T` with performance `P` if `P` improves with `E`

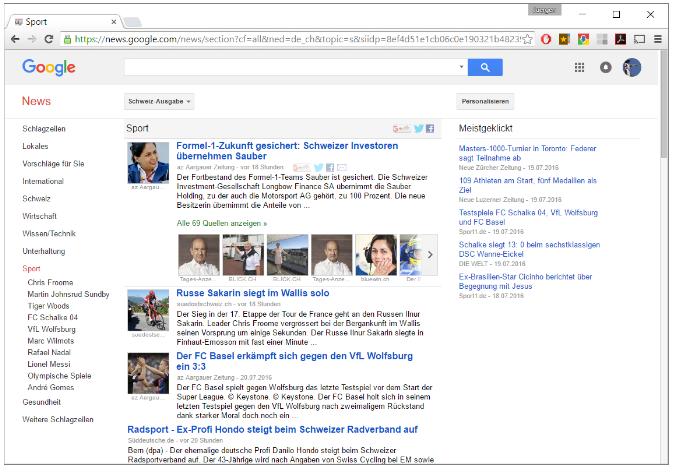
Unsupervised Learning

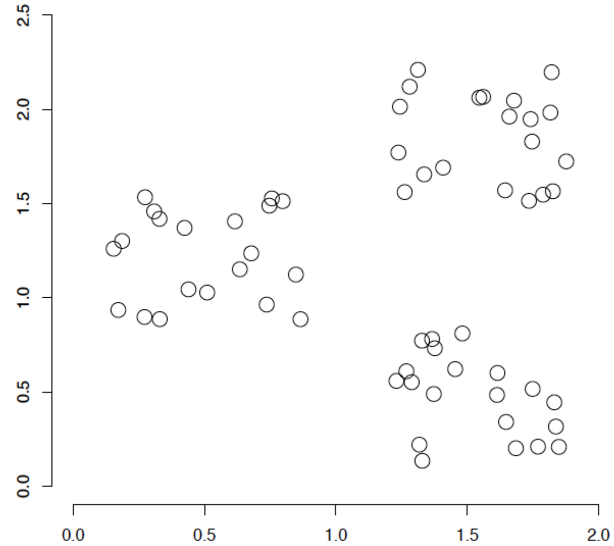
- tasks `T` that are solved
  - identifying similar items
    - e.g., recommender systems
      - e.g., collaborative filtering
  - identifying correlated features
    - e.g., dimensionality reduction
      - e.g., PCA
  - mapping a sample (based on its features) to some output
    - e.g., clustering = map to a group
      - e.g., K-Means
- the model to solve T is inferred from data `E` based on some distinctive features of `E`
- the algorithm does not have access to `P`
- the algorithm learns in the sense that more data `E` should improve `P`

Clustering (1)

Clustering

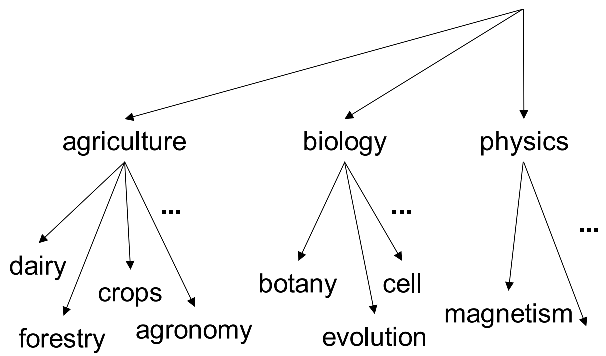
- aim to organize a dataset into groups, i.e., clusters
  - Iris dataset into distinct species
  - customers into target groups
  - news articles into topic groups (e.g., Google News)
  - …
- ~ unsupervised classification
- based on some similarity measure
  - all instances within a cluster should be similar
  - and instances in different clusters should be dissimilar
- may also produce a description for each cluster discovered
  - i.e., a representative instance, a label, several labels, ….





Clustering Documents (2)

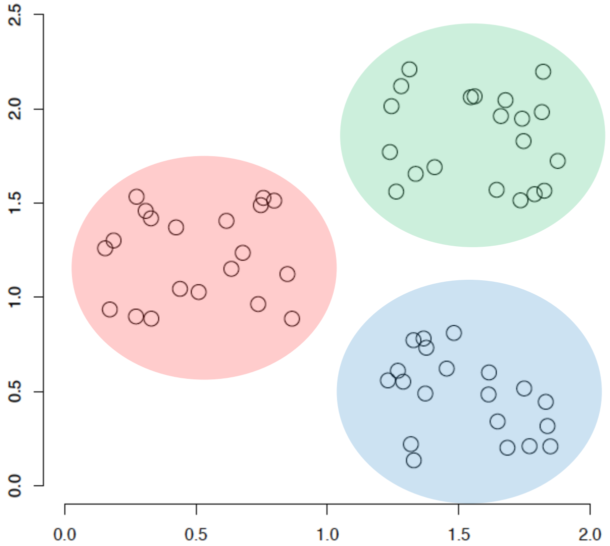
- may also built a hierarchy (relations) between clusters
- often based on unsupervised machine learning
  - i.e., can run fully automated w/o training
- may need to be fine-tuned via parameters
  - e.g., number of clusters
- tend to be computationally expensive



Clustering via Partitioning

Partitioning Approach

- construct a partition of n instances into a set of K clusters
    - given: a set of instance and the number K
    - find: a partition of K clusters that optimizes the partitioning criterion
        - optimal?
            - intractable for many objective functions
            - in many cases would require full enumeration
        - more practical: heuristic solution



K-Means

Idea

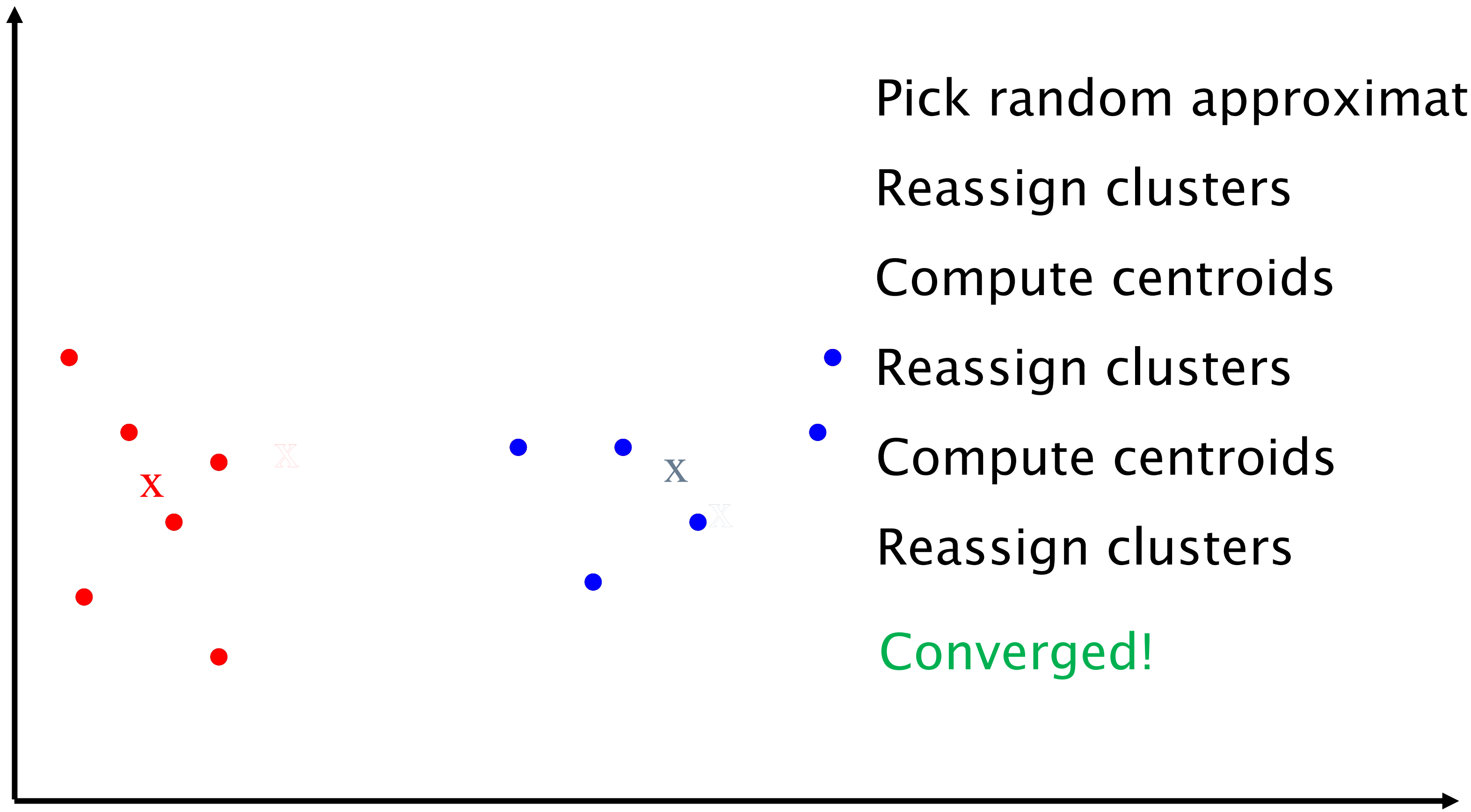
- creates K clusters
- interpret samples x as real-valued vectors $\vec{x}$
  - data preparation: numeric data only
- assignment of x to a cluster is based on its distance to the cluster centroids
- centroid of a cluster C_i: $\mu(C_i) = \frac{1}{|C_i|}\sum_{\vec{x} \in C_i} \vec{x}$

Algorithm

```
select K random samples {c1, c2,… ,cK} as approximation of
centroids
until termination condition
    for each sample xi
        assign xi to the cluster Cj such that dist(xi,cj) is
        minimal
    for each cluster Cj update the approximations of centroids
        cj = μ(Cj)
```

- termination conditions
  - clusters converge (= do not change)
  - fixed number of iterations
  - centroid positions unchanged

K-Means Example for K=2



Pick random approximations

Reassign clusters

Compute centroids

Reassign clusters

Compute centroids

Reassign clusters

Converged!

How Many Clusters K?

1. Number of clusters *K* is given
   - partition *n* samples into predetermined number of clusters
2. Finding the “right” number of clusters is part of the problem
   - partition *n* samples into appropriate number of clusters
   - often “try and error”
3. Use an algorithm to determine *K* automatically
   - define a function to assess the “quality” of all clusters
     - e.g., pairwise distance of all samples within a cluster to measure how homogenous the cluster is
   - increase *K* until no further quality improvement

Discussion K-Means

Advantages

- easy to implement and understand (“white box”)

Disadvantages

- assumes that clusters are sphere-shaped
- number of iterations and resulting clusters results depend on seed choice
    - use heuristic rather than random picks
- algorithm may converge on local minima
    - re-run with different seeds
    - post-process resulting clusters
        - split the n “worst” clusters into 2 (or more) sub-clusters
        - merge 2 close clusters (=centroid are close) into one
- relatively slow
    - updating centroid after each new sample assignment may speed up the process

Cluster Evaluation Metrics (1)

- in case we have a classified data set (gold standard)
  - homogeneity score
    - $\in [0; 1]$ where 1 means that each computed cluster contains only samples of one gold standard cluster
  - completeness score
    - $\in [0; 1]$ where 1 means that all samples from a gold standard cluster are assigned to the same computed cluster
  - adjusted rand index (ARI)
    - overlap between the sets of clusters (computed vs. gold standard)
      - overlap = number of common items
    - $ARI \in [-1; 1]$ where 1 means equality

Cluster Evaluation Metrics (2)

- in case the gold standard is not known
  - sum of squared error (SSE)
    - sum of squared distance of each sample to the centroid of its assigned cluster
    - squared: penalty for samples that are far from the cluster centroid
  - silhouette coefficient
    - per sample: (normalized) average distance of sample to all other points in the same cluster – average distance of sample to all other points in the next nearest cluster
    - overall silhouette coefficient as average
    - $\in [-1; 1]$ where 1 means dense clusters