CS-E3210 Machine Learning: Basic Principles

Lecture 9: Clustering

slides by Alex Jung, 2017

Department of Computer Science
Aalto University, School of Science

Autumn (Period I) 2017

Today's Motto

Your Friends Probably Look Like You

Outline

What you should learn today...

- difference between supervised and unsupervised learning
- organize datasets into clusters
- difference between hard and soft clustering
- one algorithm for hard clustering
- one algorithm for soft clustering

Applications of Clustering

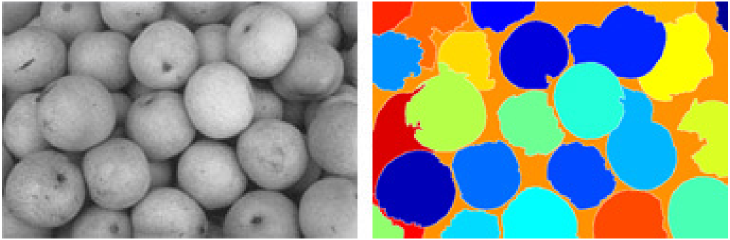
- biology: group homologous sequences into gene families
- marketing: partition consumers into market segments
- sociology: find communities in social networks
- natural language processing: identify topics in a corpus
- computer vision: group pixels to segments
- climatology: find weather regimes
- ...

Applications of Clustering: Market Segmentation

Applications of Clustering: Social Network Analysis

Applications of Clustering: Text Document Analysis

Applications of Clustering: Image Segmentation



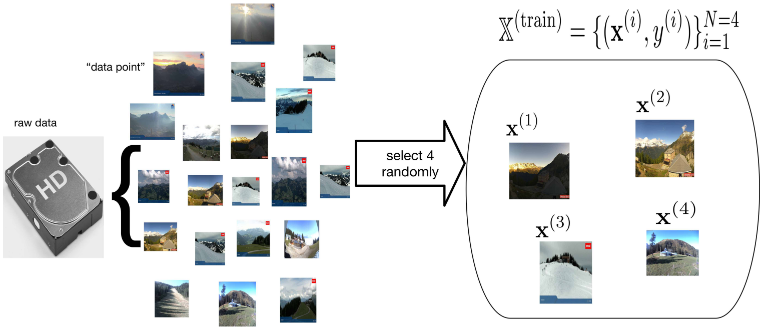
Outline

Ski Resort Marketing

- you still did not find another job
- thus, you still work as marketing of a ski resort
- hard disk full of webcam snapshots (gigabytes of data)
- want to group them into “winter” and ”summer” images
- you have only a few hours for this task ...

The Dataset

$$\mathbb{X}^{(\mathrm{train})} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N=4}$$



ML workflow so far...

- create $\mathbb{X}^{(\text{train})} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N_t}$ by manual labeling
- features $\mathbf{x}^{(i)} \in \mathcal{X}$ and label $y^{(i)} \in \mathcal{Y}$ of ith data point
- define loss $L((\mathbf{x}, y), h(\cdot))$ (e.g., $L((\mathbf{x}, y), h(\cdot)) = (y - h(\mathbf{x}))^2$)
- define hypothesis space $\mathcal{H}$ (e.g., linear maps $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$)
- learn predictor $h(\cdot) : \mathcal{X} \to \mathcal{Y}$ by empirical risk minimization

$$\min_{h(\cdot) \in \mathcal{H}} \mathcal{E}\{h(\cdot) | \mathbb{X}^{(\text{train})}\} = \min_{h(\cdot) \in \mathcal{H}} \frac{1}{|\mathbb{X}^{(\text{train})}|} \sum_{(\mathbf{x}, y) \in \mathbb{X}^{(\text{train})}} L((\mathbf{x}, y), h(\cdot))$$

NO Time For Labeling

- already spent 3 weeks on grouping into winter/summer
- we have time for manual labelling only one picture
- can we cluster/group the snapshots directly into two groups ?
- if clustering works, need to look at ONE SINGLE snapshot

How To Group or Cluster into Two Clusters?

Definition of Clustering?

- informal description according to Wikipedia:

  "Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters."

- no single best formal definition of clustering!

Look At Data in Feature Space

Clustering vs. Classification

- common: feature vector $\mathbf{x} \in \mathbb{R}^d$ and label $y \in \{0, 1\}$
- classification is supervised learning method
  - need labeled training data $\mathbb{X}^{(\text{train})} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{t=1}^{N_{\text{train}}}$
  - learn classifier (LogReg, SVM,...) via ERM using $\mathbb{X}^{(\text{train})}$
  - predict label y for (classify) new snapshot with features $\mathbf{x}$
- clustering is unsupervised learning method
  - unlabeled data $\mathbb{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$
  - find cluster index $y^{(i)}$ for each data vector $\mathbf{x}^{(i)}$
  - clustering based on intrinsic geometry of data points $\mathbb{X}$

Hard vs. Soft-Clustering

- hard clustering:
  - data points belong to one and only one cluster, $y^{(i)} \in \{0, 1\}$
  - data points partitioned into non-overlapping clusters
  - hard-clustering method: K-means
- soft clustering:
  - datapoint may belong to several clusters
  - clusters are overlapping
  - strength of association/degree of belonging $y^{(i)} \in [0, 1]$
  - soft-clustering using Gaussian mixture models (covered in APM course CS-E4820)

Outline

K-means Clustering
The Basic Idea

- partition $\mathbb{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ into clusters $\mathcal{C}_y$, $y \in \{0, \ldots, K-1\}$
- hard clustering: each $\mathbf{x}^{(i)}$ belongs exactly to one $\mathcal{C}_{y^{(i)}}$
- cluster $\mathcal{C}_y$ represented by cluster mean $\mathbf{m}_y$
- popular clustering method: K-means

K-means Clustering

The Algorithm (also called Lloyd's Algorithm)

1. input: $\mathbb{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, number K, cluster means $\{\mathbf{m}_y\}_{y=0}^{K-1}$
2. repeat until convergence
   - cluster assignment: for each $\mathbf{x}^{(i)}$ find nearest cluster mean

     $$y^{(i)} = \underset{y \in \{0,\ldots,K-1\}}{\operatorname{argmin}} \|\mathbf{x}^{(i)} - \mathbf{m}_y\|_2$$

   - mean update: for each cluster $\mathcal{C}_y = \{\mathbf{x}^{(i)} : y^{(i)} = y\}$, compute

     $$\mathbf{m}_y = (1/|\mathcal{C}_y|) \sum_{i:y^{(i)}=y} \mathbf{x}^{(i)}$$

- output: cluster means $\{\mathbf{m}_y\}_{y=0}^{K-1}$ and assignments $\{y^{(i)}\}_{i=1}^{N}$

K-means Clustering

The Algorithm in Action

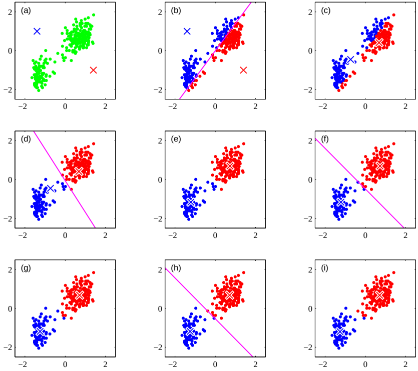


Figure 9.1 of Bishop (2006)

K-means Clustering
Initialization

- K-means needs initial choice for cluster means $\mathbf{m}_y$
- no optimal choice in general; some heuristics:
  - use K randomly selected data points
  - use K random perturbations of sample mean
  - divide range of principal component into K grid points

K-means Clustering
The Optimization Problem

- K-means can be interpreted as optimization method
- for $\{\mathbf{m}_y\}_{y=0}^{K-1}$, $\{y^{(i)}\}_{i=1}^{N}$, define cost/distortion function:

$$\mathcal{E}\left(\{\mathbf{m}_y\}_{y=0}^{K-1}, \{y^{(i)}\}_{i=1}^{N}\right) = \sum_{i=1}^{N} \left\| \mathbf{x}^{(i)} - \mathbf{m}_{y^{(i)}} \right\|^2$$

- $\mathcal{E}\left(\{\mathbf{m}_y\}_{y=0}^{K-1}, \{y^{(i)}\}_{i=1}^{N}\right)$ is non-convex and non-smooth!
- K-means=coordinate descent for $\mathcal{E}\left(\{\mathbf{m}_y\}_{y=0}^{K-1}, \{y^{(i)}\}_{i=1}^{N}\right)$
- allows for convergence diagnosis

Coordinate Descent

- consider function $f(x, y)$ of two variables x, y
- we aim for x_0, y_0 such that $f(x_0, y_0) = \min_{x,y} f(x, y)$
- this minimization problem is often difficult
- sometimes, the minimization of $f(x, y)$ either w.r.t. to x and fixed y and vice-versa is easy
- coordinate descent:
  - given current guess x_k, y_k obtain new x_{k+1} by
    $$x_{k+1} = \operatorname*{argmin}_x f(x, y_k)$$
  - obtain new y_{k+1} by
    $$y_k = \operatorname*{argmin}_y f(x_{k+1}, y)$$

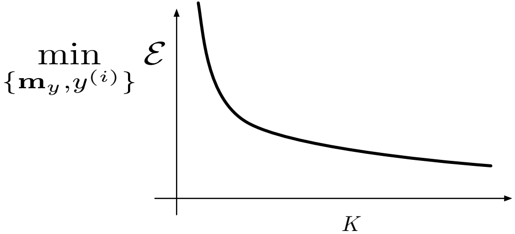
K-means Clustering
Convergence

- K-means is coord. descent for $\mathcal{E}\left(\{\mathbf{m}_y\}_{y=0}^{K-1}, \{y^{(i)}\}_{i=1}^{N}\right)$
- objective $\mathcal{E}$ monotonically decreasing throughout iterations
- however, K-means can get stuck in local minimum of $\mathcal{E}$
- workaround: run K-means several times with random initial.
- pick solution yielding smallest cost $\mathcal{E}$

K-means Clustering

How To Choose K

- e.g., by finding "elbow" in distortion curve



- often clustering used as pre-processing for learning method
- choose K by cross-validation of overall method
- use complexity penalization (favouring smaller K)

K-means Clustering

Properties

- conceptually and algorithmically simple
- typically only small number of iterations required
- K-means sensitive to initialization
- iterations might get stuck in local optimum
- workaround: run K-means several times with random init.
- select solution with smallest cost $\mathcal{E}$

Outline

"Lets Put on the Probabilistic Glasses"

- lets consider $K = 2$ for simplicity (extended easily to other K)
- lets interpret $y^{(i)}$ as probability of $\mathbf{x}^{(i)} \in \mathcal{C}_1$
- $y^{(i)} = P(\mathbf{x}^{(i)} \in \mathcal{C}_1 | \mathbb{X})$ "degree of $\mathbf{x}^{(i)}$ belonging to $\mathcal{C}_1$"
- what is degree of $\mathbf{x}^{(i)}$ belonging to $\mathcal{C}_0$?
- K-means enforces $P(\mathbf{x}^{(i)} \in \mathcal{C}_1 | \mathbb{X}) \in \{0, 1\}$

Gaussian Mixture Model (GMM)

- cluster $\mathcal{C}_y$ represented by Gaussian distribution $\mathcal{N}(\mathbf{x}; \mathbf{m}_y, \mathbf{C}_y)$[1]
- cluster $\mathcal{C}_0$ has mean $\mathbf{m}_0 \in \mathbb{R}^d$ and covariance $\mathbf{C}_0 \in \mathbb{R}^{d \times d}$
- cluster $\mathcal{C}_1$ has mean $\mathbf{m}_1 \in \mathbb{R}^d$ and covariance $\mathbf{C}_1 \in \mathbb{R}^{d \times d}$
- probability of data point $\mathbf{x}^{(i)}$ belonging to $\mathcal{C}_1$ is

$$y^{(i)} = \frac{\mathcal{N}(\mathbf{x}^{(i)}; \mathbf{m}_1, \mathbf{C}_1)}{\mathcal{N}(\mathbf{x}^{(i)}; \mathbf{m}_0, \mathbf{C}_0) + \mathcal{N}(\mathbf{x}^{(i)}; \mathbf{m}_1, \mathbf{C}_1)} \in [0, 1]$$

[1] $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{C}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})\right)}{\sqrt{\det\{2\pi\mathbf{C}\}}}$

(Approximate) Maximum Likelihood

- consider current guess for $y^{(i)}$ (degree of $\mathbf{x}^{(i)} \in \mathcal{C}_1$)
- "effective size" of $\mathcal{C}_1$ is $N_1 = \sum_{i=1}^{N} y^{(i)}$
- "effective size" of $\mathcal{C}_0$ is $N_0 = \sum_{i=1}^{N} (1 - y^{(i)}) = N - N_1$
- approximate $\mathbf{m}_1$ by $(1/N_1) \sum_{i=1}^{N} y^{(i)} \mathbf{x}^{(i)}$
- approx. $\mathbf{C}_1$ by $(1/N_1) \sum_{i=1}^{N} y^{(i)} \left(\mathbf{x}^{(i)} - \mathbf{m}_1\right) \left(\mathbf{x}^{(i)} - \mathbf{m}_1\right)^T$
- similarly for $\mathbf{m}_0$ and $\mathbf{C}_0$

A Soft-Clustering Algorithm

- 1: use initial guess for GMM parameters $\mathbf{m}_0, \mathbf{m}_1, \mathbf{C}_0, \mathbf{C}_1$
- 2: update degrees of belonging

$$y^{(i)} = \mathcal{N}(\mathbf{x}^{(i)}; \mathbf{m}_1, \mathbf{C}_1)/(\mathcal{N}(\mathbf{x}^{(i)}; \mathbf{m}_0, \mathbf{C}_0) + \mathcal{N}(\mathbf{x}^{(i)}; \mathbf{m}_1, \mathbf{C}_1))$$

- 3: update GMM parameters $N_1 = \sum_{i=1}^{N} y^{(i)}, N_0 = N - N_1,$

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{i=1}^{N} y^{(i)}\mathbf{x}^{(i)}, \ \mathbf{C}_1 = \frac{1}{N_1}\sum_{i=1}^{N} y^{(i)}\left(\mathbf{x}^{(i)} - \mathbf{m}_1\right)\left(\mathbf{x}^{(i)} - \mathbf{m}_1\right)^T$$

$$\mathbf{m}_0 = \frac{1}{N_0}\sum_{i=1}^{N}(1 - y^{(i)})\mathbf{x}^{(i)}, \ \mathbf{C}_0 = \frac{1}{N_0}\sum_{i=1}^{N}(1 - y^{(i)})\left(\mathbf{x}^{(i)} - \mathbf{m}_0\right)\left(\mathbf{x}^{(i)} - \mathbf{m}_0\right)^T$$

- 4: if not converged go to step 2

Soft Clustering

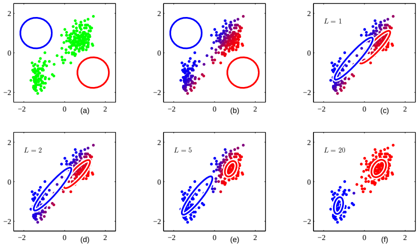


Figure 9.8 of Bishop (2006)

A Soft Clustering Algorithm

Properties

- based on generative GMM for data
- implicitly estimates GMM parameters $(\mathbf{m}_0, \mathbf{C}_0, \ldots)$
- problem of local optima (use several random initializations)
- soft cluster assignments (degree of belonging) $y^{(i)} \in [0, 1]$
- reduces to K-means for $\mathbf{C}_0 = \mathbf{C}_1 = \sigma^2 \mathbf{I}$ with small σ^2

Outline

Summary

what we learned today ...

- difference between soft- and hard clustering
- one hard-clustering algorithm, i.e., K-means
- one soft clustering algorithm (Gaussian mixture models)

What happens next?

- next lecture feature learning
- recommended preparation: read Chap 5.8. [DLBook]