Outlier Detection: Probabilistic and Density-Based Methods

Mining Massive Datasets

Prof. Carlos Castillo

Topic 20

Universitat Pompeu Fabra Barcelona

Sources

- Data Mining, The Textbook (2015) by Charu Aggarwal (chapter 8) – slides by Lijun Zhang

Probabilistic methods

Related to probabilistic model-based clustering

- Assume data is generated from a mixture-based generative model
- Learn the parameters of the model from data
  - EM algorithm
- Evaluate the probability of each data point being generated by the model
  - Points with low values are outliers

Mixture-based generative model

- Data is generated by a mixture of k distributions with probability distributions $G_1, \ldots, G_k$
- Each point $\overline{X}$ is generated as follows:
  1) Select a mixture component with probability α_i
     - Suppose it's component r
  2) Sample a data point from distribution G_r

Learning parameters from data

- Probability of generating a point

$$f^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right) = \sum_{i=1}^{k} P\left(\mathcal{G}_i, \overline{X_j}\right)$$

$$= \sum_{i=1}^{k} P(\mathcal{G}_i)P(\overline{X_j}|\mathcal{G}_i)$$

$$= \sum_{i=1}^{k} \alpha_i f^i(\overline{X_j})$$

Learning parameters from data

- Probability of generating a point

$$\mathrm{f}^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right) = \sum_{i=1}^{k} \alpha_i f^i(\overline{X_j})$$

- Probability of generating a dataset

$$f^{\text{data}}(\mathcal{D}|\mathcal{M}) = \prod_{j=1}^{n} f^{\text{point}}(\overline{X_j}|\mathcal{M})$$

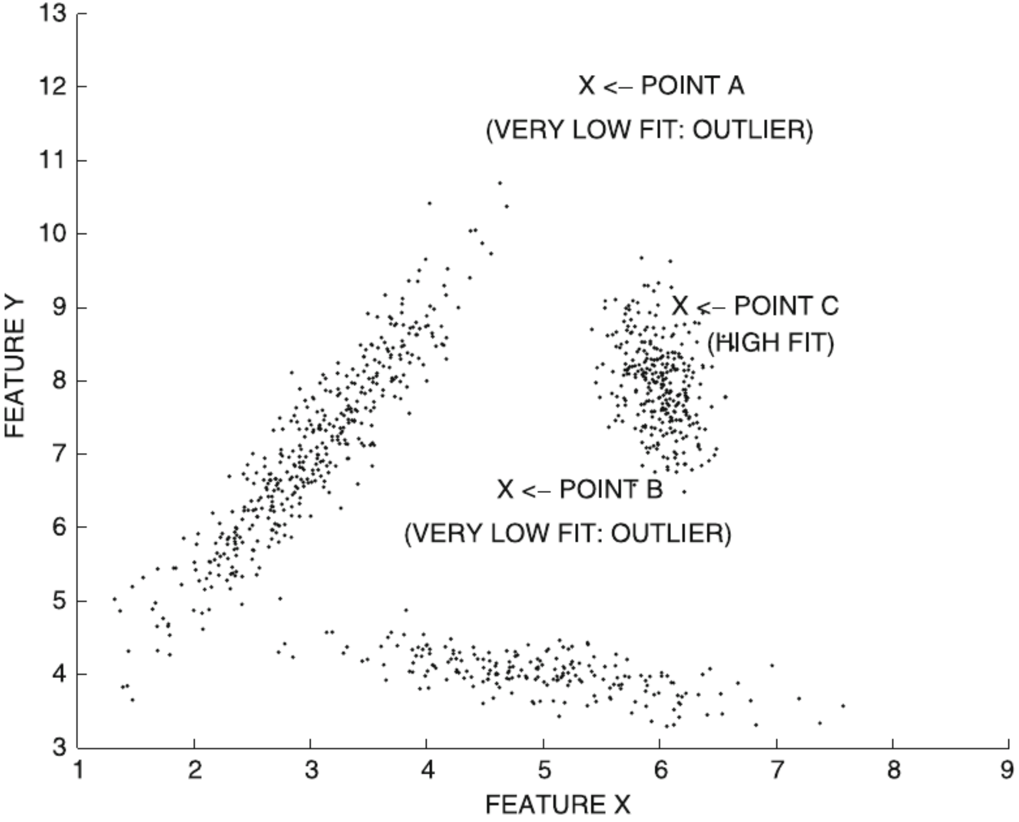
- Learning: min log loss

$$\min \mathcal{L}\left(\mathcal{D}|\mathcal{M}\right) = \log\left(\prod_{j=1}^{n} f^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right)\right) = \sum_{j=1}^{n} \log\left(\sum_{i=1}^{k} \alpha_i f^i\left(\overline{X_j}\right)\right)$$

Identifying an outlier

Outlier score:

$$f^{\text{point}}\left(\overline{X_j}|\mathcal{M}\right) = \sum_{i=1}^{k} \alpha_i f^i(\overline{X_j})$$



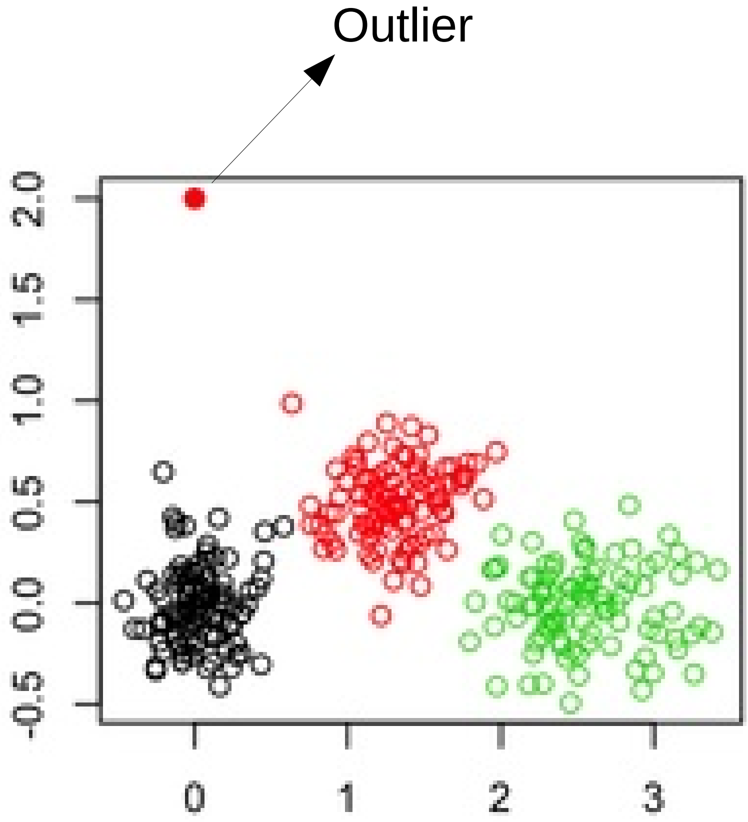
Clustering-based methods

Clustering for outlier analysis

- Clustering associate points to similar points
- Points either clearly belong to a cluster or are outliers
- Some clustering algorithms also detect outliers
  - Examples: DBSCAN, DENCLUE

Simple method

- Cluster data, associating each point to a centroid, e.g., using k-means
- Outlier score = distance of point to its centroid



Exercise

Spreadsheet does k-means to cluster the electric scooter database

1) Re-run with a new initial clustering
2) Do you see any interesting pattern in the final clustering assignment?
3) Find outliers according to the method from the previous slide

Answer in
Google Spreadsheet

Improved method

- Cluster data
- Outlier score = local Mahalanobis distance with respect to center of cluster r

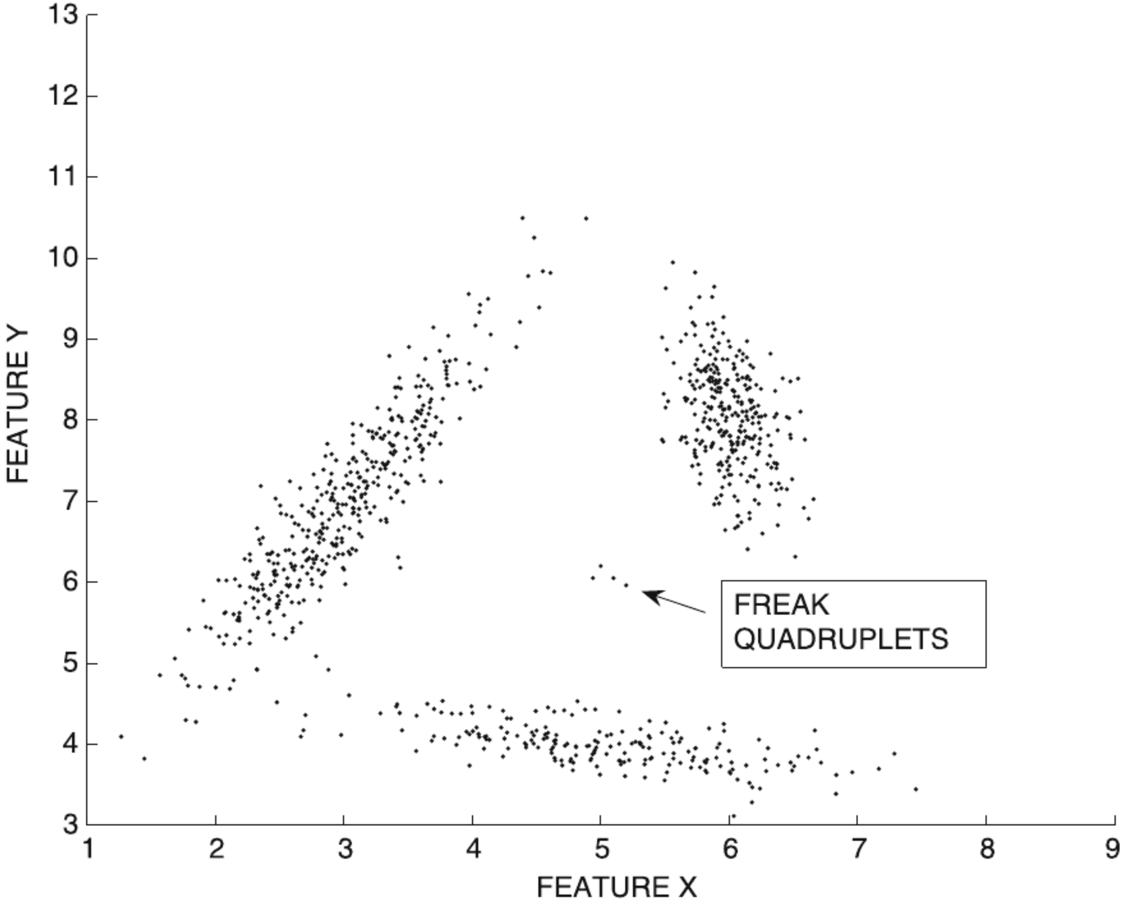
$$\mathrm{Maha}(\overline{X}, \overline{\mu_r}, \Sigma_r) = \sqrt{(\overline{X} - \overline{\mu_r})\Sigma_r^{-1}(\overline{X} - \overline{\mu_r})^T}$$

$\overline{\mu_r}$ is the mean of the cluster r

Σ_r is the covariance matrix of cluster r

Improved method (cont.)

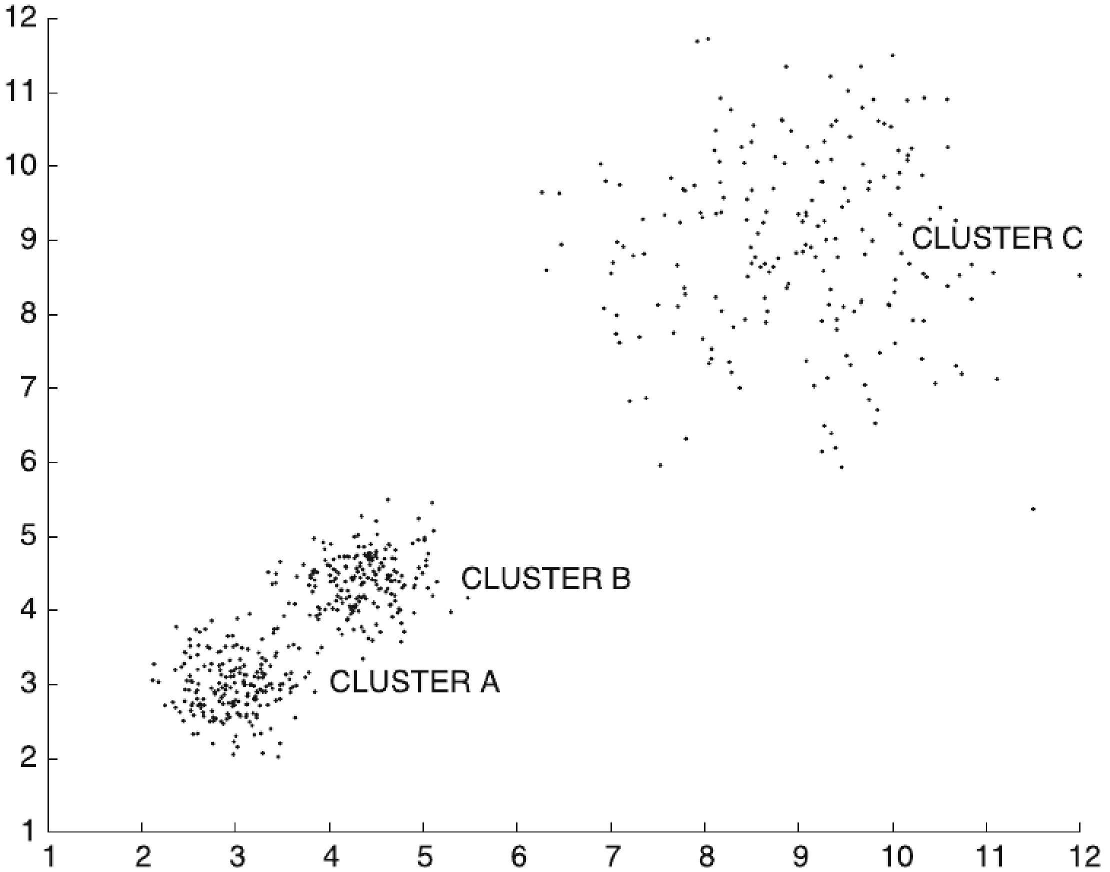
- Remove tiny clusters



Density-based methods

Density-based methods

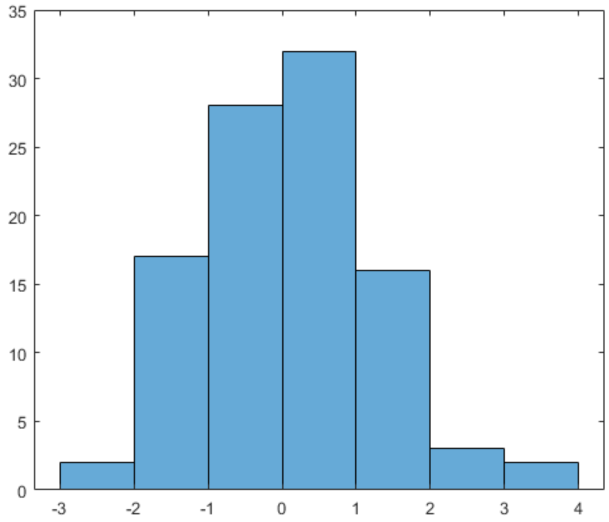
- Key idea:
  find sparse regions
  in the data
- Limitation:
  cannot handle
  variations of
  density



Histogram- and grid-based methods

Histogram-based method:

1) Put data into **bins**
2) Outlier score: *num* – 1, where *num* is the number of items in the same **bin**

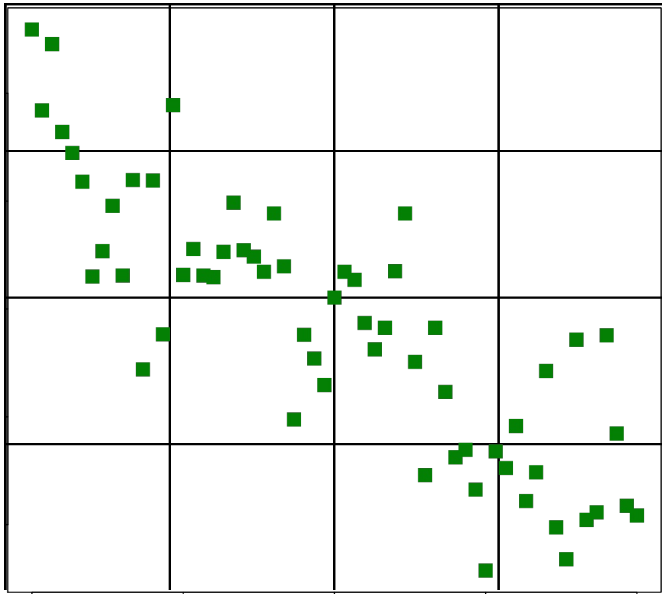


Clear outliers are alone or almost alone in a **bin**

Histogram- and grid-based methods

Grid-based method

1) Put data into a **grid**
2) Outlier score: *num* – 1, where *num* is the number of items in the same **cell**



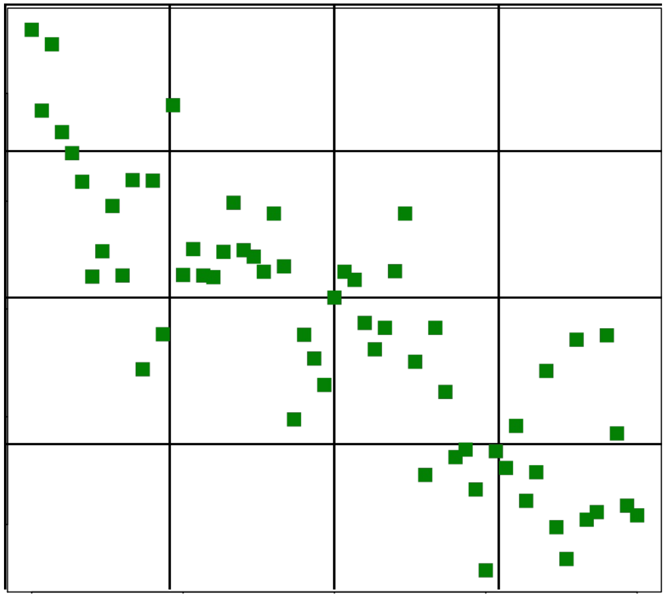
Clear outliers are alone or almost alone in a **cell**

Problems with grid-based methods

How to choose the grid size?

Grid size should be chosen considering data density, but density might vary across regions

If dimensionality is high, then most cells will be empty



Kernel-based methods

- Given n points $\overline{X_1}, \overline{X_2}, \ldots, \overline{X_n}$

$$f(\overline{X}) = \frac{1}{n} \sum_{i=1}^{n} K_h(\overline{X} - \overline{X_i})$$

- K_h is a function peaking at $\overline{X_i}$ with *bandwidth* h
- For instance, a Gaussian kernel:

$$K_h(\overline{X} - \overline{X_i}) = \left(\frac{1}{\sqrt{2\pi} \cdot h} \right)^d \cdot e^{-\|\overline{X} - \overline{X_i}\|^2 / (2h^2)}$$

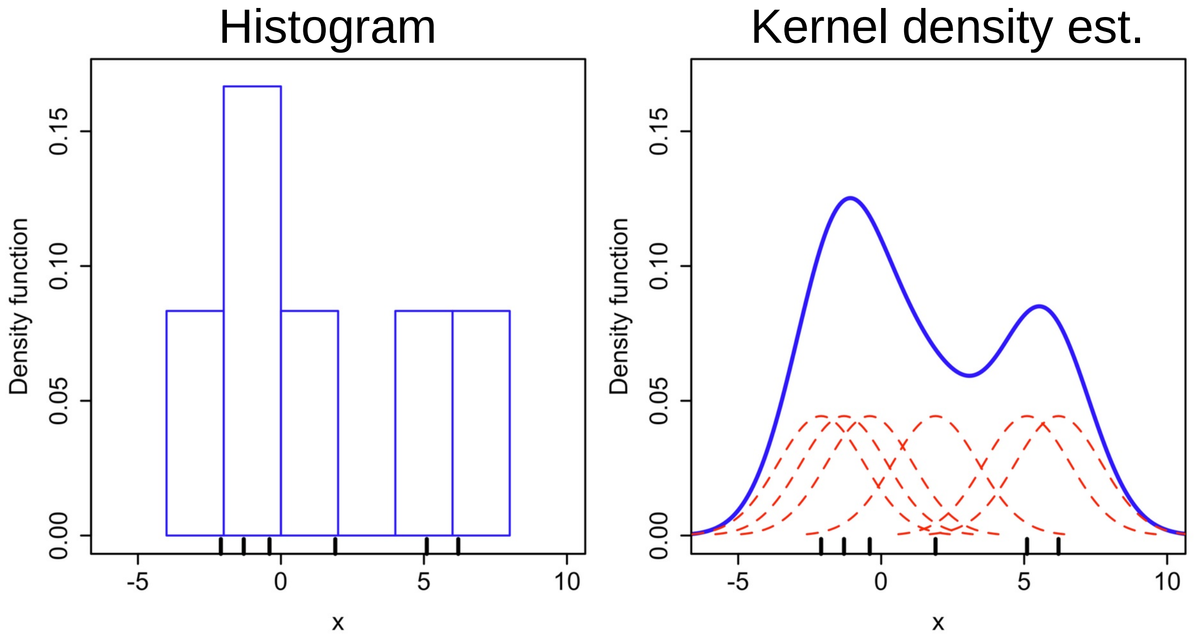
Kernel-based methods (cont.)

- Example with a Gaussian kernel

$$\overline{X} = < -2.1, -1.3, -0.4, 1.9, 5.1, 6.2 >$$

- Each K_h in **red**
- f = sum of K_h in **blue**

$$f(\overline{X}) = \frac{1}{n}\sum_{i=1}^{n} K_h(\overline{X} - \overline{X_i})$$



[Wikipedia: Kernel density estimation]

Summary

Things to remember

- Probabilistic methods
- Clustering-based methods
- Density-based methods

Exercises for TT19-TT21

- Data Mining, The Textbook (2015) by Charu Aggarwal
  - Exercises 8.11 → all except 10, 15, 16, 17