Robust Information-theoretic Clustering

Simon Lackerbauer

Institut für Informatik
Ludwig-Maximilians-Universität München
lackerbauer@lrz.mwn.de

Seminar *Information Theoretic Data Mining* im WS 2015/16

Outline

Clustering Problems

Solution: The iterative approach

- VAC – Volume After Compression
- RF – Robust Fitting
- CM – Cluster Merging

Example: Cat Retina Images

Summary

References

Outline

Clustering Problems

- There exist a wide range of possible clustering algorithms.
- Many need user input or assume only Gaussian clusters
- We want an algorithm without user input that automatically selects appropriate cluster functions

Clustering Problems

Example: How not to do it

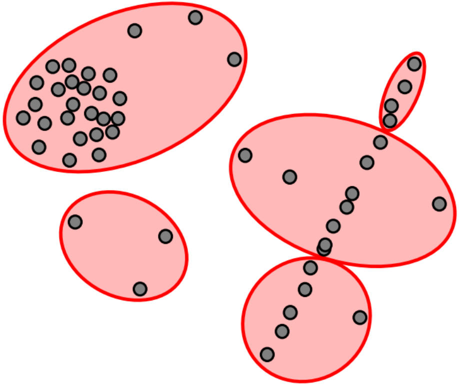


Figure 1: Example of “Bad” Clustering[1]

Clustering Problems

Example: Reasonable reduction

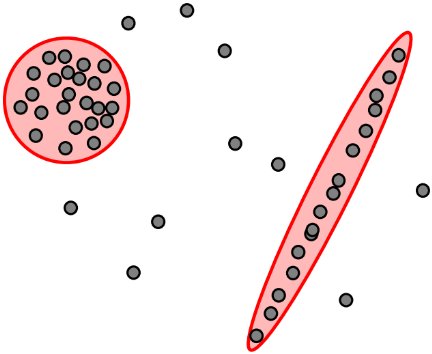


Figure 2: Example of “Good” Clustering[1]

Comparison of examples

What makes the second pattern better than the first?

- It's more descriptive of the interesting patterns in the data, because outliers have been “omitted”.
- The clusters in the “good” example are those a human would immediately recognize as points being associated somehow.

Measuring Success

This human intuition must be translated into a dependable clustering algorithm, for which two measures for success can be defined:

- **Goodness of fit**
- **Efficiency**

Measuring Success

This human intuition must be translated into a dependable clustering algorithm, for which two measures for success can be defined:

- **Goodness of fit**
- **Efficiency**

Measuring Success

This human intuition must be translated into a dependable clustering algorithm, for which two measures for success can be defined:

- **Goodness of fit**
- **Efficiency**

Outline

Clustering Problems

Solution: The iterative approach

- VAC – Volume After Compression
- RF – Robust Fitting
- CM – Cluster Merging

Example: Cat Retina Images

Summary

References

Outline

Proposition for a solution: VAC

VAC - Volume after Compression

- does not specify good grouping
- specifies for two groupings x, y which one is better (e.g., because $VAC(x) < VAC(y) \rightarrow$ x is a better grouping)
- size of total, **lossless** compression

Proposition for a solution: VAC

Integer Encoding

- point coordinates are always integers
- self-delimiting encoding of integers: Elias (gamma) codes
- smaller integers require fewer bytes

Proposition for a solution: VAC

Cluster Encoding

- uses Huffman encoding for positioning points with probability distribution according to assumed cluster pdf
- such, if we assume the correct distribution for the cluster, core points will be more efficiently encoded

Proposition for a solution: VAC

Cluster Encoding

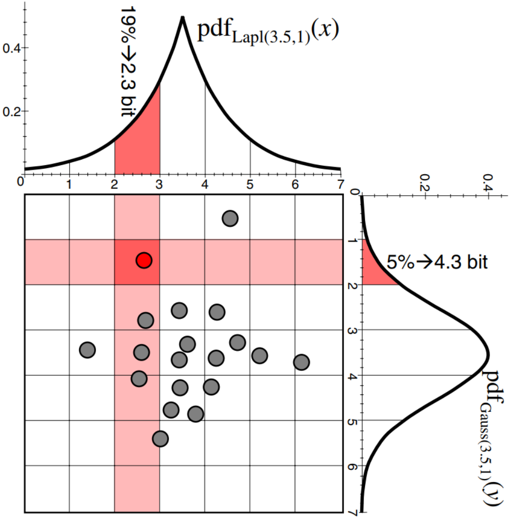


Figure 3: Example of VAC[1]

Proposition for a solution: VAC

Cluster encoding

Definition (VAC of point $\vec{x}$)

Let $\vec{x} \in \mathbb{R}^d$ be a point of a cluster C and $\overrightarrow{pdf}(\vec{x})$ be a d-dimensional vector of probability density functions which are associated to C. Each $pdf_i(x_i)$ is selected from a set of predefined probability density functions with corresponding parameters, i.e.
$PDF = \{pdf_{Gauss(\mu_i,\sigma_i)}, pdf_{uniform(lb_i,ub_i)}, pdf_{Lapl(a_i,b_i)}, \ldots\}$,
$\mu_i, lb_i, ub_i, a_i \in \mathbb{R}, \sigma_i, b_i \in \mathbb{R}^+$. Let γ be the grid constant (distance between grid cells). The VAC_i of coordinate i of point $\vec{x}$ corresponds to

$$VAC_i(x) = \log_2 \frac{1}{pdf_i(x_i) \cdot \gamma}$$

The VAC of point $\vec{x}$ corresponds to

$$VAC(x) = \left(\log_2 \frac{n}{|C|}\right) + \sum_{0 \le i < d} VAC_i(x)$$

Proposition for a solution: VAC

Cluster Encoding and Decorrelation

- γ is a measure of granularity of grid cells
- absolute VAC changes with grid resolution, but relative VAC stays the same
- to choose optimal parameter settings for clusters, we use the statistical parameter of the dataset
- if data is correlated amongst itself, define a decorrelation matrix iff the VAC savings at least compensate for saving the decorrelation matrix

Outline

Two helper algorithms

Robust Fitting

- Start: get as input a set of clusters $\mathcal{C} = \{C_1, ..., C_k\}$ by an arbitrary method
- for every C_i in $\mathcal{C}$ define a similarity measure (decorrelation matrix $==$ ellipsoid)
- use the VAC score to try out decorrelation matrices until the one with the lowest VAC is found

Two helper algorithms

Robust Fitting

Decorrelation Matrix:

- contains the vectors that define the space in which points in the cluster reside
- to improve robustness of cluster center estimation use coordination-wise median instead of arithmetic means
- of the several matrices generated during this step, again partition into core points and noise by choosing the one with best VAC

Two helper algorithms

Robust Fitting

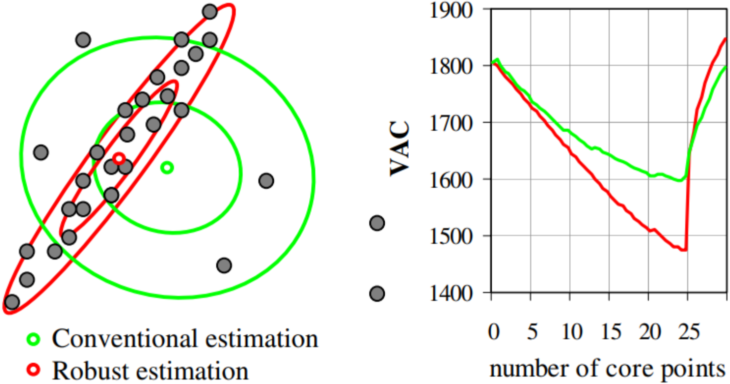


Figure 4: Conventional and robust estimation[1]

Outline

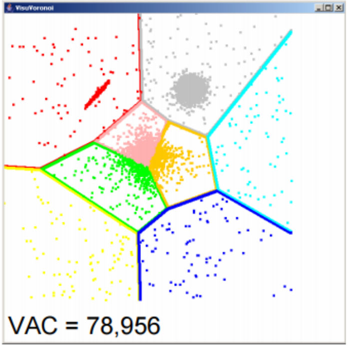
Two helper algorithms

Cluster Merging

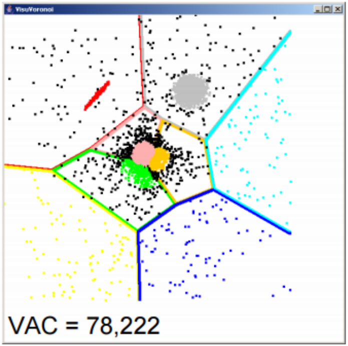
- Start: get as input a set of clusters $\mathcal{C} = \{C_1, ..., C_k\}$ by an arbitrary method
- Purify (of noise) each cluster individually

Two helper algorithms

Cluster Merging



(a) K-means.

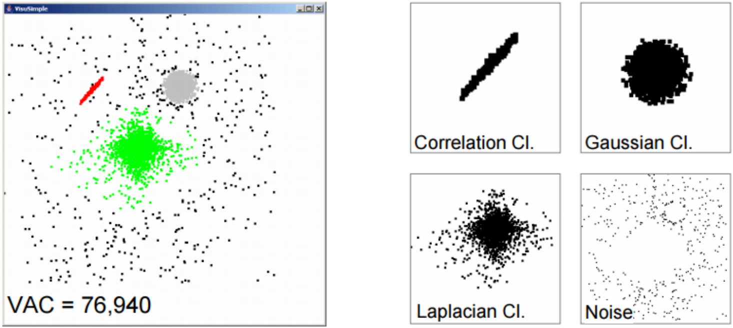


(b) After purifying.

Figure 5: Clustering by using K-means and then purifying[1]

Two helper algorithms

Cluster Merging



(c) RIC result. (d) Detailed view.

Figure 6: After merging[1]

Outline

Example: Cat Retina Images

- 219 blocks of retinal images, 96 tiles per image $\rightarrow$ 22,024 tiles in total (example tiles in figure 7)
- each tile is represented as vector of 7 features (figure 8(a))
- RIC finds 13 clusters, color coded in figure 8(b)
- Example clusters in figures 9(a)-(f)

Example: Cat Retina Images

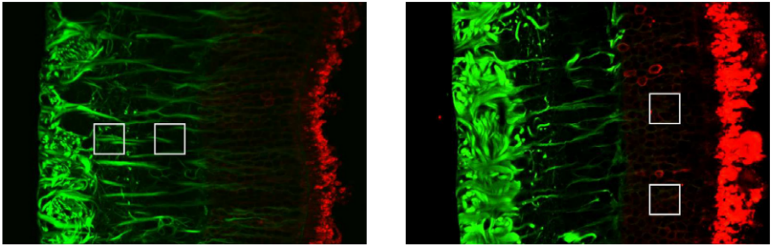

(a) (b)

Figure 7: Examples of tiles[1]

Example: Cat Retina Images

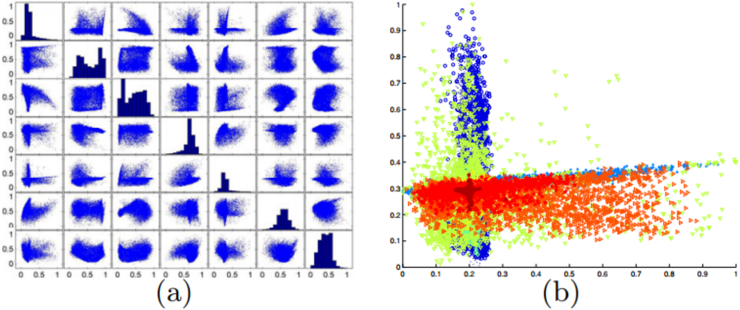


Figure 8: Visualization of cat retina data[1]

Example: Cat Retina Images

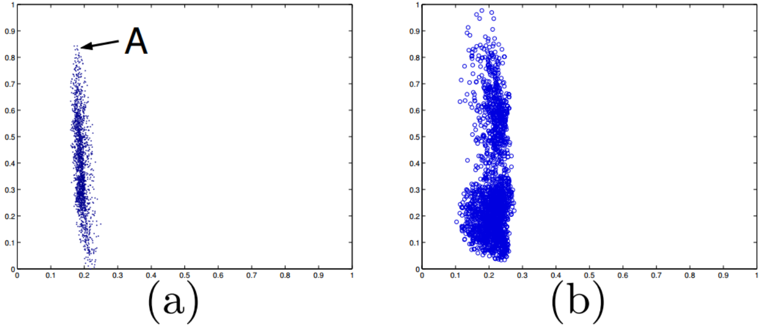


Figure 9: Example clusters[1]

Example: Cat Retina Images

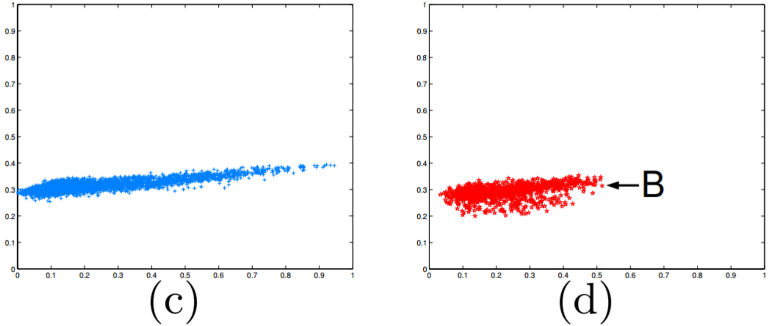


Figure 8: Example clusters[1]

Example: Cat Retina Images

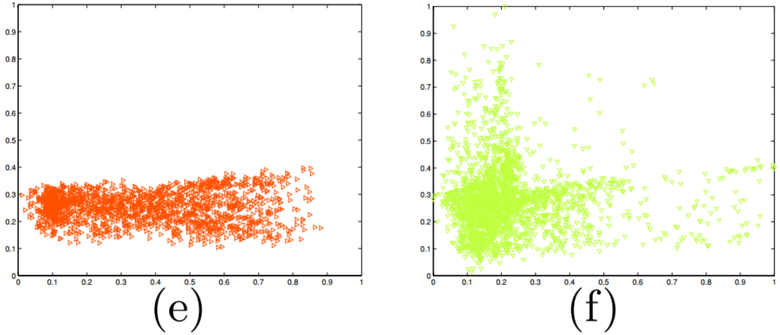


Figure 8: Example clusters[1]

Outline

Summary

- The VAC criterion provides a stable measure of goodness of fit.
- The RIC framework is very flexible, does not rely on user input and can handle any distribution that can be described by a pdf
- Anytime a new, better clustering algorithm is introduced, RIC can improve on it by running its parts (CM and RF) with the better algorithm as a starting point

Outline

References

[1] Christian Böhm, Christos Faloutsos, Jia-Yu Pan, and Claudia Plant. Robust information-theoretic clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 65–75. ACM, 20 August 2006.