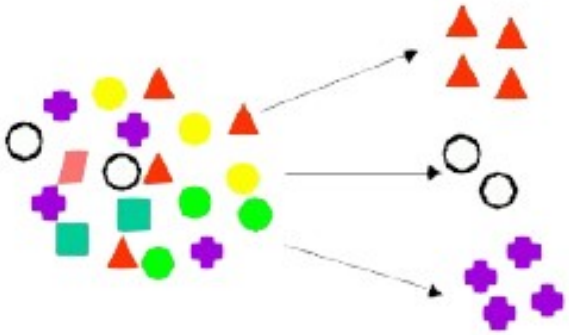


Efficient Evidence Accumulation Clustering for large datasets/big data

Diogo Alexandre Oliveira Silva

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Ana Fred 1 and Helena Aidos 2

Examination Committee

Chairperson:	Professor Full Name
Supervisor:	Professor Full Name 1 (or 2)
Member of the Committee:	Professor Full Name 3

Month 2015

Dedicated to someone special...

Acknowledgments

A few words about the university, financial support, research advisor, dissertation readers, faculty or other professors, lab mates, other friends and family...

Resumo

Inserir o resumo em Português aqui com o máximo de 250 palavras e acompanhado de 4 a 6 palavras-chave...

Palavras-chave: palavra-chave1, palavra-chave2,...

Abstract

Insert your abstract here with a maximum of 250 words, followed by 4 to 6 keywords...

Keywords: keyword1, keyword2,...

Contents

List of Tables

List of Figures

Glossary

API	Application Programming Interface.
CPU	Central Processing Unit.
EAC	Evidence Accumulation Clustering.
GPGPU	General Purpose computing in Graphics Processing Units.
GPU	Graphics Processing Unit.
HAC	Hierarchical Agglomeration Clustering.
PCA	Principal Component Analysis.
PC	Principal Component.
QK-Means	Quantum K-Means.
Qubit	Quantum bit.
SL-HAC	Single-Linkage Hierarchical Agglomeration Clustering.
SVD	Singular Value Decomposition.

Chapter 1

Clustering

1.1 The problem of clustering

Hundreds of methods for data analysis exist. Many of these methods fall into the realm of machine learning, which is usually divided into 2 major groups: *supervised* and *unsupervised* learning. Supervised learning deals with labeled data, i.e. data for which ground truth is known, and tries to solve the problem of classification. Examples of supervised learning algorithms are Neural Networks, Decision Trees, Linear Regression and Support Vector Machines. Unsupervised learning deals with unlabeled data for which no extra information is known Clustering algorithms, expectation-maximization and Principal Component Analysis are examples of unsupervised algorithms.

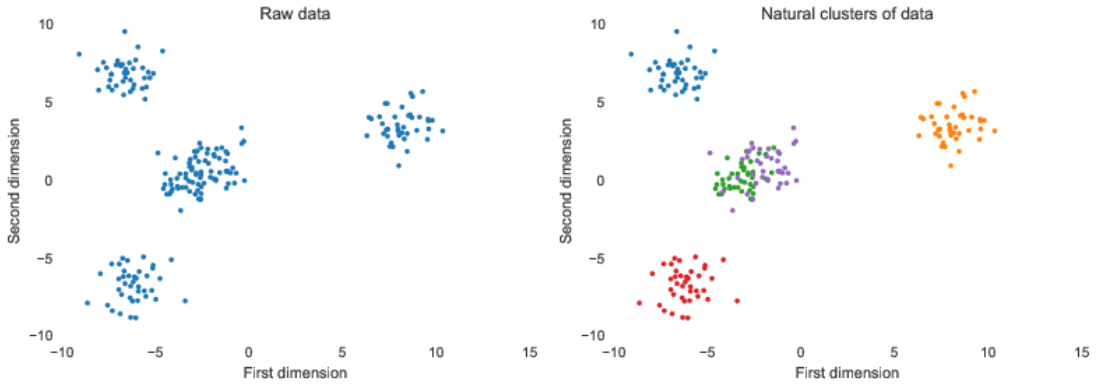
Cluster analysis methods are unsupervised and the backbone of the present work. The goal of data clustering, as defined by [1], is the discovery of the *natural grouping(s)* of a set of patterns, points or objects. In other words, the goal of data clustering is to discover structure on data. The methodology used is to group patterns (usually represented as a vector of measurements or a point in space [2]) based on some similarity, such that patterns belonging to the same cluster are typically more similar to each other than to patterns of other clusters. Clustering is a strictly data-driven method, in contrast with classification techniques which have a training set with the desired labels for a limited collection of patterns. Because there is very little information, as few assumptions as possible should be made about the structure of the data (e.g. number of clusters). And, because clustering typically makes as few assumptions on the data as possible, it is appropriate to use it on exploratory structural analysis of the data. The process of clustering data has three main stages [2]:

- **Pattern representation** refers to the choice of representation of the input data in terms of size, scale and type of features. The input patterns may be fed directly to the algorithms or undergo *feature selection* and/or *feature extraction*. The former is simply the selection of which features of the originally available should be used. The later is deals with the transformation of the original features such that the resulting features will produce more accurate and insightful clusterings, e.g. Principal Component Analysis. It should be noted that
- **Pattern similarity** refers to the definition of a measure for computing the similarity between two

patterns.

- **Grouping** refers to the algorithm that will perform the actual clustering on the dataset with the defined pattern representation, using the appropriate similarity measure.

As an example, Figure 1.1a shows the plot of a simple synthetic dataset - a Gaussian mixture of 5 distributions. No extra information other than the position of the points is given, since clustering algorithms are unsupervised methods. Figure 1.1b presents the desired clustering for this given dataset.



(a) Input data, unlabeled.

(b) Desired labels.

Figure 1.1: Gaussian mixture of 5 distributions. Fig. 1.1a shows the raw input data, i.e. how the algorithms ”see” the data. Fig. 1.1b shows the desired labels for each point, which here means their corresponding Gaussian.

1.2 Definitions and Notation

This section will introduce relevant definitions and notation within the clustering context that will be used throughout the rest of this document.

A *pattern* $\mathbf{x}$ is a single data item and, without loss of generalization, it consists of a set of d *features* x_i that characterize that data item, $\mathbf{x} = (x_1, \ldots, x_d)$, where d is referred to as the dimensionality of the pattern. A *pattern set* (or data set) $\mathcal{X}$ is then the collection of all n patterns $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. The number of features is usually the same for all patterns in a given pattern set.

In cluster analysis, the desired clustering, typically, is one that reflects the natural structure of the data, i.e. the original true clustering. In other words, one wants to group the patterns that came from the same state of nature when they were generated, the same *class*. A class, then, can be viewed as a source of patterns and the effort of the clustering algorithm is to group patterns from the same source. Throughout this work, these classes will be also be referred to as the ”natural” or ”true” clusterings. *Hard* clustering (or partitional) techniques assign a class label l_i to each pattern $\mathbf{x}_i$. The whole set of labels of a pattern set $\mathcal{X}$ is given by $\mathcal{L} = l_1, \ldots, l_n$. Closely related to the whole set of labels is the concept of a *partition*, which completely describes a clustering in a different way. A partition P is a collection of k *clusters*. A cluster C is a subset of nc patterns $\mathbf{x}_i$ taken from the pattern set, where the patterns belonging to one subset don’t belong to any other in the same partition. A clustering *ensemble* $\mathbb{P}$ is a

set of partitions from a given pattern set. The relationship between the above concepts is given by the following expressions:

$$\mathbb{P} = \left\{P^1, P^2, \dots P^N\right\} \tag{1.1}$$

$$P^j = \left\{C_1^j, C_2^j, \dots C_{k_j}^j\right\} \tag{1.2}$$

$$C_i^j = \left\{x_1, x_2, \dots x_{nc_i^j}\right\} \tag{1.3}$$

The last concept to be introduced

Formulation

Some notation and nomenclature, adopted from [3], should be defined since it will be used throughout the remainder of the present work. The term *data* refers to a set X of n objects or patterns $X = \{\mathsf{x}_1, ..., \mathsf{x}_n\}$, and may be represented by $\chi = \{x_1, ..., x_n\}$, such that $x_i \in \mathbb{R}^d$. A clustering algorithm takes χ as input and returns k groups or *clusters* C of some part of the data, which form a *partition* P. A clustering *ensemble* $\mathbb{P}$ is group of N partitions. This means that:

$$\mathbb{P} = \left\{P^1, P^2, ...P^N\right\} \tag{1.4}$$

$$P^i = \left\{C_1^i, C_2^i, ...C_{k_i}^i\right\} \tag{1.5}$$

$$C_k^i = \{x_a, x_b, ..., x_z\} \tag{1.6}$$

where C_j^i is the $j-th$ cluster of the $i-th$ partition, which contains k_i clusters and n_j^i is the number of samples that constitutes that cluster, with

$$\sum_{j=1}^{k} n_j^i = n, \qquad i = 1, ..., N \tag{1.7}$$

1.3 Characteristics of clustering techniques

There are two big main types of clustering: hierarchical and partitional.

usually some metric (e.g. euclidean distance, Pearson correlation).

Cluster analysis is a relevant technique across several domains ([4]):

- grouping users with similar behaviour or preferences in **customer segmentation**;
- image segmentation in the field of **image processing**;
- clustering gene expression data, among other application, in the domain of **biological data analysis**;
- generation of hierarchical structure for easy access and retrieval of **information systems**;

1.4 K-Means

K-Means is one of the earliest clustering algorithms to offer a partitional solution to the clustering problem

It's also widely used due to it's simplicity and efficiency. Because of this, it's also often used as a foundational step of more complex and robust algorithms, such as EAC. the algorithm works as follows

explain K-means algorithm

As an example, the output of the K-means algorithm to the data presented in Fig. 1.1 is represented in Fig. 1.2. The algorithm executed with 4 random centroids.

Figure 1.2: The output labels of the K-Means algorithm with the number of clusters (input parameter) set to 4.

The number of clusters was purposefully set to an "incorrect" number to demonstrate that it is not trivial to discover, even in such a simple example. In this synthetic dataset, the number of clusters is not clear due to the two superimposed Gaussians. When no prior information about the dataset is given, the number of clusters can be hard to discover. This is why, when available, a domain expert may provide valuable insight on tuning the initialization parameters.

1.5 Single-Link

Do the same thing as K-Means but for SL

Bibliography

[1] Anil K Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. ISSN 01678655. doi: 10.1016/j.patrec.2009.09.011. URL `http://dx.doi.org/10.1016/j.patrec.2009.09.011`.

[2] a. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3): 264–323, 1999. ISSN 03600300. doi: 10.1145/331499.331504.

[3] Ana N L Fred and Anil K Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[4] Charu C Aggarwal and Chandan K Reddy. *Data clustering algorithms and applications*. ISBN 9781466558229.

Appendix A

Vector calculus

In case an appendix if deemed necessary, the document cannot exceed a total of 100 pages...

Some definitions and vector identities are listed in the section below.

A.1 Vector identities

$$\nabla \times (\nabla \phi) = 0 \tag{A.1}$$

$$\nabla \cdot (\nabla \times \mathbf{u}) = 0 \tag{A.2}$$