KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering

DRAFT

To remove, add 'final' to class options

Clustering evolving data using kernel-based methods

Rocco Langone

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

Month Year

Clustering evolving data using kernel-based methods

Rocco LANGONE

Supervisory Committee:
Mr. De Voorzitter, chair
Prof. dr. ir. Johan A. K. Suykens, supervisor
Jurylid 1
Jurylid 2
External jury member
(External affiliation)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

Month Year

© 2013 KU Leuven – Faculty of Engineering
Uitgegeven in eigen beheer, Rocco Langone, Kasteelpark Arenberg 10, B-3001 Heverlee, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

ISBN XXX-XX-XXXX-XXX-X

D/XXXX/XXXX/XX

Preface



Abstract

Beknopte samenvatting

Contents

Abstract iii

Contents vii

List of Figures ix

List of Tables xi

1 Introduction 3

1.1 Background 3

1.2 Challenges 4

1.3 Objectives 5

1.4 Chapter by Chapter Overview 6

1.5 Main Contributions 8

2 Conclusions and Future Challenges 11

A Appendix 13

A.1 Cluster Quality Evaluation 13

A.1.1 Overview 13

A.1.2 Data-Points 13

A.1.3 Network Data 13
A.1.4 Agreement between Partitions 13

Bibliography **15**

Curriculum **17**

List of Figures

1.1 **Thesis overview**. 8

List of Tables

Todo list

Instructies van de faculteit:

De hoofdstukken: Elk hoofdstuk is ingelast met een bepaald doel voor ogen. Dit doel wordt vermeld in de eerste paragraaf van elk hoofdstuk. Naargelang de aard van de tekst (experiment, uitvoering, theoretische ontwikkeling, ...) volgen de paragrafen elkaar op. Beweringen worden altijd gestaafd, hetzij door eigen experimenten, hetzij door een theoretische afleiding, hetzij door verwijzingen naar de literatuur. Elk hoofdstuk eindigt met een kort samenvattend besluit waarbij nagegaan wordt in hoeverre de doelstelling van het betrokken hoofdstuk verwezenlijkt is. De deelbesluiten moeten de lezer automatisch leiden naar het algemeen besluit aan het einde van het werk.

Chapter 1

Introduction

{ch:introduction}

1.1 Background

We live in the Information Age. The recent development of information technology (computers, internet, smart phones, any kind of sensor etc.) has a big impact on science and society. In principle, the large amount of available data can help to grasp the complexity of many phenomena of interest, in order to make new scientific discoveries rather than designing optimal business strategies or optimizing industrial processes.

Recognition of complex patterns in the data is of crucial importance to extract useful knowledge. In this context, clustering is a fundamental mode of understanding and learning [**?**, **?**]. It refers to the task of organizing the data in meaningful groupings based only on the similarity between the data elements, and therefore is exploratory in its essence. Since no target or desired patterns are known a priori, it belongs to the family of unsupervised learning techniques [**?**].

Unveiling the underlying structure of the data through the cluster analysis is just one side of the coin. The rest of the story is related to the dynamic version of the problem, i.e. monitoring the evolution of the clusters. Understanding how the behaviour of the system under study changes in time represents a key issue in many domains [**?**, **?**]. By this point of view dynamic clustering, by investigating how clusters form, evolve and die represents a useful tool.

The topic of this thesis is related to the design and application of kernel-based methods to perform dynamic clustering. Kernel methods are a class of machine

learning techniques where two main modelling phases are present. First a mapping of the data into a high dimensional feature space is performed. Then, the design of learning algorithms in that space allows to discover complex and non-linear relations in the original input space [**?**]. A major role in this work is played by least squares support vector machines (LS-SVM) [**?**], which are a kind of support vector machine (SVM) [**?**] based on a constrained optimization framework characterized by the presence of the L_2 loss function in the objective. By modifying the objective and/or the constraints of the core formulation, it is possible to develop models tailored for a given application, with a systematic model selection procedure and high generalization abilities.

1.2 Challenges

{ch:challenges}

The main issues tackled in this thesis can be summarized as follows:

- **Community detection via kernel methods**: A network is a collection of nodes or vertices joined by edges and represents the patterns of connections between the components of complex systems [**?**]. Usually real networks display a high level of order and organization. For example, the distribution of edges is characterized by high concentrations of edges within special groups of vertices, and low concentrations between these groups. The problem of identifying such clusters of nodes is called community detection [**?**]. The main challenges posed by the usage of kernel methods for community detection are related to the choice of the kernel function, the model selection, the out-of-sample extension, the scalability to large data-sets.

- **Analysis of dynamic communities**: Community detection of evolving networks aims to understand how the community structure of a complex network changes over time [**?**, **?**]. A desirable feature of a clustering model which has to capture the evolution of communities is the temporal smoothness between clusters in successive time steps. Providing a consistent clustering at each time results in a smooth view of the changes and a greater robustness against noise [**?**, **?**].

- **Fault detection**: With the development of information and sensor technology many process variables in a power plant like temperature, pressure and many others can be monitored. These measurements give an information on the current status of a machine and can be used to predict the faults and plan an optimal maintenance strategy [**?**, **?**]. A useful model in this case must be able to catch, in an online fashion, the

degradation process affecting the machine, to avoid future failures of the components and unplanned downtimes.

- **Clustering in a non-stationary environment**: In many real-life applications non-stationary data are generated according to some distribution models which change over time. Therefore a proper cluster analysis can be useful to detect important change points and in general to better understand the dynamic of the system under investigation. In this case a clustering algorithm is required to continuously adapt in response to new data and to be computationally efficient for real-time applications [**?**].

1.3 Objectives

In this thesis the following objectives can be delineated:

- **to envisage a whole kernel-based framework for community detection**. In this context many issues arise. First of all it is important to chose a proper kernel function to describe the similarity between the nodes of the network under investigation. Then a key point is represented by the model selection, i.e. finding the natural number of communities which are present in the network and eventually tuning the kernel hyper-parameters. The kernel-based model must be also able to accurately predict the membership of new nodes joining the network, without performing the clustering from scratch. Moreover, since many real-world networks contains millions of nodes and edges, the network data have to be processed in a reasonable time. Finally, the research carried-out for solving the static community detection problem paves the way to the development of models for the analysis of evolving networks.
- **to design a model for community detection in a changing scenario**. An evolving network can be described as a sequence of snapshots, where each snapshot represents the configuration of the network at a particular time instant. When community detection is performed at time t, the clustering should be similar to the clustering at the previous time-step $t-1$, and should accurately incorporate the actual data. In this way, if the data at time t does not deviate from historical expectations, the clustering should be similar to that from time $t-1$, while if the structure of the data changes significantly, the clustering must be modified to account for the new structure. Thus, a good clustering algorithm must trade-off the benefit of maintaining a consistent clustering

over time with the cost of deviating from an accurate representation of the current data.

- **to conceive modelling strategies for clustering stationary and non-stationary data streams**. Data streams are a sequence of data records stamped and ordered by time [**?**]. Clustering data streams in real time is an ambitious problem with ample applications. If the data distribution is stationary, emphasis should be given to the offline construction of the model. Once properly designed, such a model can be used to cluster the data stream in an online fashion by means of the out-of-sample extension. However, if the data distribution is non-stationary, the initial model soon becomes obsolete and must be quickly updated. Therefore the development of a fast, adaptive and accurate model is an important objective.

1.4 Chapter by Chapter Overview

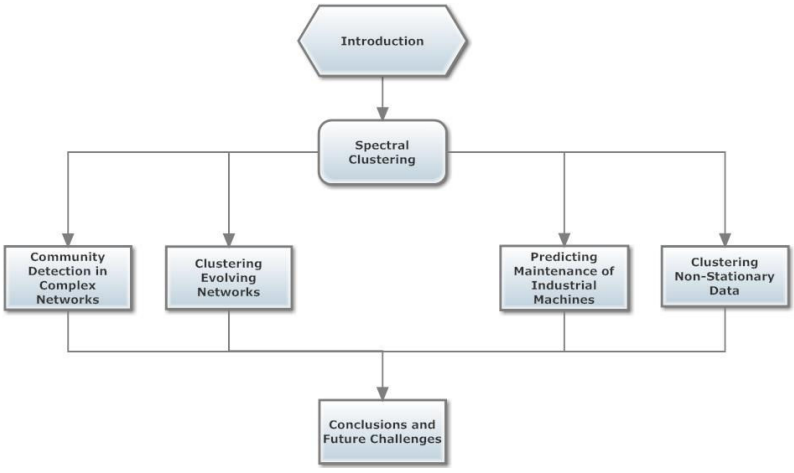
The general structure of this document is sketched in Figure 1.1 and can be described as follows:

- **Chapter** 2 contains four sections. First of all a general introduction to spectral clustering, one of the most successful clustering algorithms, is given. Then kernel spectral clustering (KSC) is reviewed. KSC is a spectral clustering algorithm formulated in the LS-SVM optimization framework, with the possibility to extend the clustering model to out-of-sample data for predictive purposes. Moreover a model selection criterion called the balanced line fit (BLF) is also present. For these reasons it represents the starting point to face the challenges described in section 1.2. In the third section the soft kernel spectral clustering (SKSC) algorithm is introduced. Instead of using the hard assignment method present in KSC, a fuzzy assignment rule based on the cosine distance from the cluster prototypes in the projection space is suggested. We also introduce a related model selection technique, called average membership strength criterion (AMS), which solves the major drawbacks of BLF. Finally we show that SKSC can improve the interpretability of the results and the clustering performance with respect to KSC, mainly in cases of large overlap between the clusters. In the last section we give a summary of some performance measures used to assess the quality of the partitions produced by a clustering algorithm.
- **Chapter** 3 is dedicated to the community detection problem. After reviewing the important literature in the field we illustrate our method-

ology, which is composed by four main cornerstones. First, it is crucial to extract from the given network a small sub-graph representative of its community structure, which is a challenging problem. This sub-graph can then be used to train a KSC model in a computationally efficient way and forms the basis for a good generalization. Second, the correct tuning of the kernel hyper-parameters (if any) and the number of communities is another important issue, which is solved by proposing a new model selection criterion based on Modularity. Third, the kernel functions used to properly describe the similarity between the nodes are presented. Finally, the out-of-sample extension allows not only to accurately predict the community affiliation of new nodes, but also can make the algorithm cluster millions of data in a few time on a desktop computer.

- **Chapter** 4 introduces a novel model called kernel spectral clustering with memory effect (MKSC). This method is designed to cluster evolving networks (described as a sequence of snapshots), aiming to track the long-term drift of the communities by ignoring meanwhile the short-term fluctuations. In this new formulation the desired temporal smoothness is incorporated in the objective function of the primal problem through the maximization of the correlation between the actual and the previous models. Moreover new measures are presented in order to judge the quality of a partition produced at a given time. The new measures are the weighted sum of the snapshot quality and the temporal quality. The former only measures the quality of the current clustering with respect to the current data, while the latter measures the temporal smoothness in terms of the ability of the actual model to cluster the historic data. These new measures are also used to perform the model selection.

- **Chapter** 5 discuss the application of KSC to an industrial use case. Here we assume stationarity, i.e. the regimes experienced by the system under analysis do not change over time. Accelerometer data are collected from a packing machine to monitor its conditions. In order to describe the ongoing degradation process due to the dirt accumulation in the sealing jaws first of all we apply a windowing operation on the data, accounting for historical values of sealing quality. The size of the window, together with the bandwidth of the radial basis function (RBF) kernel and the number of clusters are tuned using the BLF criterion. Then an optimal kernel spectral clustering model is trained offline to identify two main regimes, that we can interpret as normal behaviour and critical conditions (need of maintenance). Thanks to out-of-sample extension property, this model is used online to predict in advance when the machine needs maintenance, which in principle implies the maximization of the production capacity and the minimization of downtimes.



{Fig_chapter_overview}

Figure 1.1: **Thesis overview**.

- **Chapter** 6 deals with clustering non-stationary data. A new adaptive method named incremental kernel spectral clustering (IKSC) is devised. In a first phase a KSC model is constructed, then it is updated online according to the new points belonging to the data stream. The central idea behind the proposed technique concerns expressing the clustering model in terms of prototypes in the eigenspace, which are continuously adapted through the out-of-sample eigenvectors calculation. Moreover the training set is formed only by the cluster centers in the input space, which are also updated in response to new data. This compact representation of the model and the training set in terms of cluster centroids makes the method computationally efficient and allows to properly track the evolution of complex patterns over time.

- **Chapter** 7 concludes the manuscript and proposes future research directions.

In general, in all the experiments discussed in each chapter, the values reported to assess the cluster quality are average measures over 10 runs of the algorithm under investigation. Although a fully statistical significance analysis has not been performed, they give good indications about the performances of different methods.

1.5 Main Contributions

In what follows the main contributions of this thesis are resumed:

- **Community detection via KSC**. We conceived a complete methodology to cluster network data. The whole procedure can be summarized in three main stages: extract from the network a small sub-graph which retains the community structure of the entire graph, train an optimal KSC model using the selected sub-graph as training set, use the out-of-sample extension to assign the memberships to the remaining nodes. The precise tuning of the hyper-parameters and the choice of an appropriate kernel function are also essential. Furthermore, an algorithm to provide moderated outputs named SKSC and a related model selection criterion called AMS are proposed. Finally we show that the technique can be used for large scale applications. The relevant papers are [**?**, **?**, **?**, **?**, **?**, **?**].

- **Kernel spectral clustering with memory effect**. A new model is designed to handle evolving networks. At the level of the primal optimization problem typical of LS-SVM, we introduce a memory effect in the objective function to account for the temporal smoothness of the clustering results over time. Also new model selection criteria specific for the given application are introduced. The related publications are [**?**, **?**].

- **KSC for predictive maintenance**. We have successfully applied KSC to a complex industrial case. We developed a clustering model able to infer the degradation process affecting a packing machine from the vibration signals placed on the sealing jaws. A critical modelling phase concerned the windowing of the data in order to describe the history of the sealing quality. Moreover the model selection stage was also crucial. Finally, to improve the interpretability of the results a probabilistic output has been provided. This contribution is reported in [**?**] and [**?**] .

- **Incremental kernel spectral clustering (IKSC)**. We presented a new algorithm to perform online clustering in a non-stationary environment. IKSC exploits the out-of-sample extension property of KSC to continuously adapt the initial model. In this way it is able to catch the dynamic of the clusters evolving over time. The IKSC method can model merging, splitting, appearance, death, expansion and shrinking of clusters, in a fast and accurate way [**?**].

Chapter 2

Conclusions and Future Challenges

Appendix A

Appendix

{ch:myappendix}

A.1 Cluster Quality Evaluation

In this Appendix we give a description of the main quality measures used in this thesis to evaluate and compare clustering results.

A.1.1 Data Clustering

One of the most important issues in cluster analysis is the evaluation of clustering results to find the partitioning that best captures the underlying structure of the data. Intuitively, any good algorithm should search for clusters whose members are close to each other and well separated. Many validity criteria can be found in the literature [**?**, **?**]. Here we focus on the Silhouette and the Davies-Bouldin Index (DBI).

Silhouette ranges from -1 to 1 and measures how similar data points in the same clusters are to each other compared to data points in other clusters. For each data point the more the Silhouette value is positive the more coherent this data point is to points in the same cluster. Negative values of the Silhouette indicates that this data point should be rather assigned to a different cluster. The definition of the Silhouette for the i-th datum is as follows:

$$\mathrm{Sil}(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average dissimilarity of data point i with all other data within the same cluster and $b(i)$ indicates the lowest average dissimilarity of i with

the data belonging to a different cluster. Finally, the mean Silhouette value (MSV) gives an average measure of the coherence of the groups.

In the case of the DBI a measure of similarity R_{ij} between each pair of clusters A_i and A_j is defined based on the ratio between the dispersion of the clusters s_i and s_j and their dissimilarity d_{ij}: $R_{ij} = \frac{(s_i+s_j)}{d_{ij}}$. Then DBI for k clusters becomes:

$$\text{DBI} = \frac{1}{k}\sum_{i=1}^{k} R_i$$

with $R_i = \max_{i \neq j} R_{ij}, i = 1, \ldots, k$. Roughly speaking DBI quantifies the average similarity between each cluster. Since it is desirable for the clusters to have the minimum possible similarity to each other good partitions correspond to a low value of DBI.

A.1.2 Community Detection

As we mentioned in Chapter **??** clustering network data is mainly referred in the literature as community detection. Given a network, we aim to find a partition such that the nodes within a community are densely connected and the connections are sparse between the communities. Also in this case there has been an extensive work related to assessing the quality of a division of a network into communities [**?**, **?**, **?**]. In this dissertation, we utilized two well known criteria: Modularity and Conductance.

Modularity is the most popular quality function of a graph introduced in [**?**, **?**]. It is very well considered due to the fact that closely agrees with intuition on a wide range of real world networks. It is based on the idea that a random graph is not expected to have a cluster structure, so the possible existence of clusters can be revealed by the comparison between the actual density of inter-community edges and the density one would expect to have in the graph if the vertices were attached randomly, which defines the null model. Positive high values indicate the possible presence of a strong community structure. Moreover it has been shown that finding the Modularity of a network is analogous to finding the ground-state energy of a spin system [**?**]. In formulae the definition for an unweighted network[1] is as follows:

$$\text{Mod} = \frac{1}{2m}\sum_{ij}(S_{ij} - F_{ij})\delta_{ij}$$

with i, j indicating a pair of nodes. The sum runs over all pairs of vertices, S is the similarity matrix, m indicates the total number of edges, and F_{ij}

[1] For weighted graphs an analogue definition has been provided in [**?**].

represents the expected number of edges between vertices i and j in the null model. The δ_{ij} function yields 1 if vertices i and j are in the same community and 0 otherwise. Since the standard null model of Modularity imposes that the expected degree sequence matches the actual degree sequence of the graph, the Modularity can be written as: $\text{Mod} = \frac{1}{2m}\sum_{ij}(S_{ij} - \frac{d_i d_j}{2m})\delta_{ij}$, where we indicate with $d_i = \sum_j S_{ij}$ the degree of the vertex i. Thus, a high value of Mod says that the number of inter-community edges of the graph are few in a statistically significant sense, meaning the presence of a strong community-like structure. Moreover, after some linear algebra calculations [**?**], we can write Modularity as:

$$\text{Mod} = \frac{1}{2m}\text{tr}(X^T M X)$$

where $M = S - \frac{1}{2m}dd^T$ is the Modularity matrix, $d = [d_1, \ldots, d_N]$ indicates the vector of the degrees of each node and X represents the cluster indicator matrix. Finally, as it has been pointed out in [**?**] and [**?**], Modularity suffers from some drawbacks:

- resolution limit: it contains an intrinsic scale that depends on the total number of links in the network. Modules that are smaller than this scale may not be resolved
- exhibits degeneracies: it typically admits an exponential number of distinct high-scoring solutions and often lacks a clear global maximum
- it does not capture overlaps among communities in real networks.

Many of the aforementioned problems of Modularity have been solved by properly modifying its definition, as for example in [**?**] where a multi-resolution Modularity has been introduced. In this case, however, as pointed out in [**?**], multi-resolution Modularity has a double simultaneous bias: it leads to a splitting of large clusters and a merging of small clusters, and both problems cannot be usually handled at the same time for any value of the resolution parameter.

Another meaningful quality function is the Conductance. For every community in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the Conductance is defined as the ratio between the number of edges leaving the cluster and the number of edges inside the cluster [**?**], [**?**]. In particular, Conductance $Cond(S)$ of a set of nodes $\mathcal{A} \subset \mathcal{V}$ is:

$$Cond(S) = c_S / \min(Vol(S), Vol(V \setminus S))$$

where c_S denotes the size of the edge boundary, $c_S = \{(u, v) : u \in S, v \notin S\}$, and $Vol(S) = \sum_{u \in S} d_u$, with d_u representing the degree of node u. In other words the Conductance describes the concept of a good network community as

a set of nodes that has a better internal than external connectivity. Therefore the lower the Conductance the better the community structure.

A.1.3 Comparing Partitions

In order to compare two clustering results two measures of agreement between partitions have been utilized in this thesis: the Adjusted Rand Index (ARI) and the Normalized Mutual Information (NMI).

Given two partitions $\mathcal{U}$ and $\mathcal{V}$, the ARI [**?**] equals 1 when the related cluster memberships agree completely, 0 in case of complete mismatch with equal number of positive and negative mistakes, and -1 for a complete disagreement with all negative mistakes[2]. Its most common use is as measure of agreement between the clustering outcomes of an algorithm and a known grouping which acts like a ground-truth.

The NMI [**?**] measures the statistical information shared between the partitions $\mathcal{U}$ and $\mathcal{V}$ it also takes values in the range $[0, 1]$. It tells us how much knowing one of these clusterings reduces our uncertainty about the other. The higher the NMI, the more useful the information in $\mathcal{V}$ helps us to predict the cluster memberships in $\mathcal{U}$ and vice versa.

[2] A positive mistake means that if a partition groups a pair of points in the same cluster, the other does not. A negative mistake means that a pair of points is grouped in different clusters according to an algorithm while the points are assigned to the same cluster by the other algorithm.

Instructies van de faculteit:

De bibliografie. Departementale richtlijnen terzake te volgen.

Curriculum

Publications by the Author

{ch:publications}

Journal papers

Conference proceedings

Abstracts



FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING (ESAT)
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10, B-3001 Heverlee
B-3001 Heverlee
rocco.langone@esat.kuleuven.be
http://www.esat.kuleuven.be