Advanced Clustering Methods for Mining Chemical Databases in Forensic Science [1]

Frédéric Ratle [a] Christian Gagné [b]
Anne-Laure Terrettaz-Zufferey [c] Mikhail Kanevski [a]
Pierre Esseiva [c] Olivier Ribaux [c]

[a] *Institute of Geomatics and Risk Analysis - Faculty of Earth and Environmental Sciences - University of Lausanne, Amphipôle, CH-1015 - Switzerland*

[b] *Information Systems Institute - HEC - University of Lausanne, Internef, CH-1015 - Switzerland*

[c] *School of Criminal Sciences - Faculty of Law - University of Lausanne, Batochime, CH-1015 - Switzerland*

Abstract

Heroin and cocaine gas chromatography data are analyzed using several clustering techniques. A database with clusters confirmed by police investigation is used to assess the potential of the analysis of the chemical signature of these drugs in the investigation process. Results are compared to standard methods in the field of chemical drug profiling and show that conventional approaches miss the inherent structure in the data, which is highlighted by methods such as spectral clustering and its variants. Also, an approach based on genetic programming is presented in order to tune the affinity matrix of the spectral clustering algorithm. Results indicate that all algorithms show a quite different behavior on the two datasets, but in both cases, the data exhibits a level of clustering, since there is at least one type of clustering algorithm that performs significantly better than chance. This confirms the relevancy of using chemical drugs databases in the process of understanding the illicit drugs market, as information regarding drug trafficking networks can likely be extracted from the chemical composition of drugs.

Key words: forensic science, machine learning, pattern analysis, spectral clustering, kernel methods, gas chromatography

Email address: `frederic.ratle@unil.ch` (Frédéric Ratle).
URL: `www.unil.ch/igar` (Frédéric Ratle).

[1] This work is supported by the Swiss National Science Foundation (grant no.105211-107862). The second author gratefully acknowledges postdoctoral fellow-

1 Introduction

While modern spectroscopy and chromatography provide experimental tools that allow collecting large amounts of data related to forensic science, such as illicit drugs samples composition, machine learning and pattern analysis are now a matter of excitement in the forensic science community, in order for experts to analyze and understand the collected data. Indeed, classical data analysis methods often fail in this context given the high number of variables, the noise (coming from both the phenomenon itself and the experimental analysis) that corrupts the data, and the potentially nonlinear relationships between the different variables.

This work places itself at the border of chemometrics, machine learning and forensic science in order to highlight possibly useful patterns in the chemical composition of illicit drug seizures that may guide the investigation process. Also, since a database with drug samples corresponding to known investigations is available, it is possible to determine if geometrical structures that correspond to *real* production or distribution clusters exist in the space of the input variables (i.e., chemical constituents). Finally, since drug profiling is usually done by using samples intercorrelation measurement, this data will allow us to evaluate this method and to compare it with modern clustering techniques.

Preliminary studies were made by the same authors in [1] and [2], where heroin and cocaine data were studied using conventional machine learning approaches. First, Principal Component Analysis (PCA), k-means clustering and classification algorithms (MLP, PNN, RBF networks and k-nearest neighbors) were applied. Also, cocaine data was studied with nonlinear feature extraction techniques such as kernel PCA [3], isomap [4] and locally linear embedding [5]. Kernel PCA shown to be an efficient and robust method for dimensionality reduction in this context.

A comprehensive review of the field of chemical drug profiling can be found in Guéniat and Esseiva [6]. In this book, authors have tested several statistical methods for heroin and cocaine profiling. Among other methods, they have mainly used similarity measures between samples to determine the main data classes. A methodology based on the cosine function as an intercorrelation measurement is explained in further details in Esseiva *et al.* [7]. Two drug samples are considered as being linked if their correlation is smaller than a given threshold. Also, PCA and Soft Independent Modelling of Class Analogies (SIMCA) have been applied for dimensionality reduction and supervised classification by these authors. A radial basis function neural network has

ships from the ERCIM-SARIT (Europe), the Swiss National Science Foundation, and the FQRNT (Québec).

been trained on the processed data and showed good results. The classes used for classification were based solely on statistical correlations in the chemical composition of the different samples. The profiling methodology was further developed in [8] for heroin and [9] for cocaine.

Madden and Ryder [10] have studied similar data: Raman spectra obtained from solid mixtures containing cocaine. The goal was to predict, based on the Raman spectrum, the cocaine concentration in a solid using k-nearest neighbors (KNN), neural networks and partial least squares. They have also used a genetic algorithm to perform feature selection. However, their study has been constrained by a limited number of experimental samples, even though results were good. Also, the experimental method of sample analysis is fundamentally different from the one used in this study (gas chromatography). Similarly, Raman spectroscopy data was studied in [11] using support vector machines with Gaussian and polynomial kernels, KNN, the C4.5 decision tree and a naive Bayes classifier. The goal of the classification algorithm was to discriminate samples containing acetaminophen (used as a cutting agent) from those that do not. The Gaussian kernel SVM outperformed all the other algorithms on a dataset of 217 samples using 22-fold cross-validation.

2 Method

Spectral clustering and kernel principal component analysis (kernel PCA) are two classes of machine learning algorithms based on the eigenvalue decomposition of a problem-dependent similarity (or dissimilarity) matrix. These methods can both be cast in the general framework of kernel methods. Readers unfamiliar with kernel methods can find an excellent review of this field in [12]. Kernel methods allow to apply mathematically sound linear methods for data analysis to nonlinear datasets, by implicitly projecting the input data in a high-dimensional Hilbert space, called the *feature space*, induced by some distance measure between data points.

2.1 Spectral clustering

Classical clustering algorithms, such as k-means, usually search for ball-shaped clusters by minimizing criteria such as intra-cluster variance. K-means can be summarized as follows, where i is an index over the whole dataset:

```
1: Initialize randomly $K$ cluster centers $m_k$.
2: Compute the cluster assignment vector $C(i) = \text{argmin}_{1 \le k \le K} \|x_i - m_k\|$
```

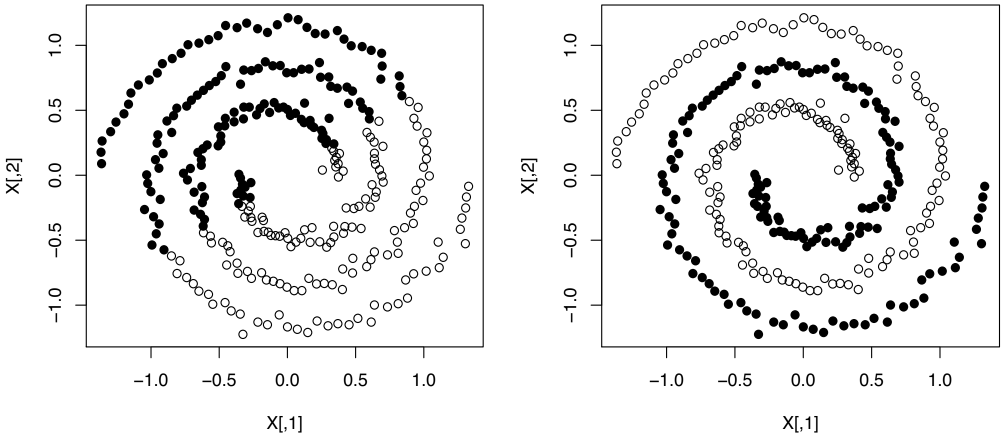

Fig. 1. Clustering the spirals dataset with k-means (left) and spectral clustering (right). The example is taken from [13].

```
3: Compute the new cluster centers $m_k' = \frac{1}{N_k} \sum_{x_i \in c_k} x_i$, where $N_k$ is the number of points included in $c_k$, the $k^{th}$ cluster.
4: Repeat 2 and 3 until the cluster assignment vector does not change.
```

These methods cannot perform well on arbitrary-shaped clusters. Spectral clustering aims at finding clusters that exhibit a specific geometry, albeit not easily found by maximizing cluster compactness. Very often, the mathematical objects formed by the data points mostly lie in a space of inferior dimensionality than that of the input space. For instance, Fig. 1 illustrates clearly the usefulness of such a method. The spirals dataset is two-dimensional, but has an intrinsic dimensionality of one. Rather than working directly with data points, spectral clustering uses an *affinity* matrix, which is a possibly nonlinear measure of similarity between points.

Many formulations of spectral clustering exist. One of the most popular is the algorithm of Ng, Jordan and Weiss [14], which can be stated as:

```
1: Form affinity matrix $A$.
2: Compute $L = D^{-1/2} A D^{-1/2}$, where $D$ is a diagonal matrix whose $(i,i)$-element is the sum of $A$'s $i$-th row.
3: Find the $k$ largest eigenvectors of $L$ and stack them in columns to form $X$.
4: Normalize each row of $X$.
5: Perform ordinary $k$-means on the columns of $X$.
```

Given an appropriate distance measure, it is assumed that we can find an eigenvector basis on which the data can be projected and clustered with a simple algorithm such as k-means. This basis is found by computing L, the Laplacian of the weighted graph induced by the affinity matrix, and by extracting its eigenvectors, on which the input data is projected. Most often, the affinity measured that is used is a Gaussian (or RBF) distance:

$$\mathbf{A}\left(i,j\right) = exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right). \tag{1}$$

As it can be seen from Eq. 1, the parameter σ has to be tuned given a particular dataset.

Apart from this now “classical” method, many improvements have been proposed in order to make spectral clustering more efficient. In [15] and [16], authors propose numerous possible improvements to the standard method:

(1) Introduction of the *conductivity* matrix;
(2) Context-dependent similarity (“asymmetric” spectral clustering);
(3) Clustering of spectral images with k-lines.

First, the conductivity matrix is an affinity measure which, rather than considering only the direct edge (the path on the graph) from one point to the other, takes into account *all* the paths that lead from one point to another. Authors compare figuratively this method to the sum of currents between two nodes in an electrical circuit.

Second, the context-dependent similarity considers the neighborhood of a point in order to estimate σ. Indeed, when dealing with datasets involving various scales in the data, it is unlikely that only one value of σ will suit the whole data. Authors propose here a method to weight the distances by taking into account the density of the neighborhood of each point. There is thus one value of σ per point in the dataset. This is very similar to the approach studied in [17], where authors estimate one value of σ per point by using also the neighborhood density.

Finally, rather than using k-means to cluster the spectral images, k-lines is used. As its name states, k-lines clusters the points around lines rather than points. The k-lines algorithm can be summarized as follows (see [15] for more details):

```
Initialize $K$ lines $\mathbf{m}_k$ randomly or as the first eigenvectors of the spectral data $y$.
for $i = 1$ to $K$ do
    Create the matrix $\mathbf{M}_i = [y_i]_{i \in N_i}$ whose columns are the points $\mathbf{y}_i$ closest to line $\mathbf{m}_i$, forming the neighborhood $N_i$.
    Compute the new line $\mathbf{m'}_i$ as the first eigenvector of $\mathbf{M}_i\mathbf{M}_i^T$.
end for
```

```
6: Repeat 2 until the $\mathbf{m}_i$'s do not change.
```

It can be argued that performing spectral clustering is somehow equivalent to applying k-means on data reduced with kernel PCA, which is described in the next section. In both cases, the problem is to find an appropriate kernel (or *affinity matrix* in the case of spectral clustering). As a kernel represents a dot product (in other words, a distance) in some high-dimensional space, finding a "good" kernel is equivalent to finding a space in which the data can be processed with linear methods. This yields a nonlinear metric in input space to perform clustering.

2.2 Kernel PCA

Classical principal component analysis aims at finding a linear low-dimensional representation of the data using the top eigenvectors of the covariance matrix of the input data. The new data is obtained by projecting the input data on these top eigenvectors. However, very often, the data cannot be linearly reduced, but a nonlinear representation can be found. For example, the application of kernel PCA on the spirals dataset (Fig. 1) would, given an appropriate distance measure, perform implicitly a linear PCA in the space induced by this distance measure, in which the spirals would form two distinct linear structures.

More technically, kernel PCA, introduced in [3], stems from the fact that the centered covariance matrix of the data can be expressed as

$$C = \frac{1}{M}\sum_{j=1}^{M} \mathbf{x}_j \mathbf{x}_j^T, \tag{2}$$

where M is the number of samples. Following this, the eigenvectors $\mathbf{v}$ of the covariance matrix have to be computed:

$$\lambda \mathbf{v} = C\mathbf{v}. \tag{3}$$

The new data is obtained by multiplying the original data by the eigenvectors of the covariance matrix. It is reasonable to assume that if we map the data into a higher-dimensional space, the nonlinear structure can be linearized. We thus replace $\mathbf{x}$ by its projection $\Phi\left(\mathbf{x}\right)$ in a high dimensional space. Eq. (2) can be re-written as:

$$\bar{C} = \frac{1}{M}\sum_{j=1}^{M} \Phi\left(\mathbf{x}_j\right)\Phi\left(\mathbf{x}_j\right)^T. \tag{4}$$

The interesting thing about this formulation is that the projection $\Phi(\mathbf{x})$ need not be computed. Indeed, it can be shown that the product on the right hand side of Eq. 4 corresponds to a dot product in the feature space. All that is needed is a kernel function k which represents this dot product. It is shown in [3] (the demonstration is not shown here for brevity reasons) that solving the modified eigenvalue problem of PCA using the new covariance matrix $\bar{C}$ of eigenvectors $\mathbf{V}$:

$$\lambda \mathbf{V} = \bar{C}\mathbf{V}, \tag{5}$$

is equivalent to solving the following eigenvalue problem

$$M\lambda\alpha = K\alpha, \tag{6}$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\,\Phi(\mathbf{x}_j)$ is called the Gram matrix (of eigenvectors α) corresponding to a kernel function. Very often, K is abbreviately called the kernel matrix.

The projection on the nonlinear manifold of a data point can simply be expressed as

$$\left(\mathbf{V^k} \cdot \Phi(\mathbf{x})\right) = \sum_{i=1}^{M} \alpha_i^k \left(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})\right) = \sum_{i=1}^{M} \alpha_i^k K(\mathbf{x}_i, \mathbf{x}). \tag{7}$$

To sum up, kernel PCA requires about the same operations as performing classical PCA, but instead of extracting the eigenvalues and eigenvectors of the covariance matrix of the input data, these steps are performed on a Gram matrix generated by some user-specified kernel.

There is a formal equivalence between spectral clustering and kernel PCA, which has been shown, for instance, in [18]. It is thus possible to optimize a measure of clustering on data projected on a basis of eigenvectors found with kernel PCA, corresponding to a "good" distance measure. Similarly, a distance measure that performs well for clustering is likely to work well as a kernel for feature extraction.

2.3 Kernel selection for clustering

Kernel selection is a task often neglected by practitioners. Indeed, a Gaussian kernel is usually used, with an isotropic variance σ chosen at best with line search.

When dealing with regression or classification tasks (using support vector machines, for example), kernel selection is a reasonably feasible task. Several methods can be used to asses the kernel "performance":

- The error on a test set (data not used for training);
- The cross-validation error;
- The number of support vectors (a measure of complexity; the fewer the better);
- The kernel alignment with the target, i.e., the classes or response variable (a measure of kernel-target correlation introduced in [19]).

Clustering is a much more ill-defined problem. Indeed, the notion of a "good" clustering, especially for arbitrary-shaped clusters, is problem-dependent. Also, in a clustering context, we do not have access to the true class labels, so these obviously cannot be used in order to optimize any clustering algorithm (even though the true class labels, when available, are usually used to evaluate a clustering method *a posteriori*).

In this paper, we test standard approach to clustering, such as k-means, alongside the methods mentioned earlier to determine the best affinity matrix, namely:

(1) K-lines
(2) Asymmetric spectral clustering
(3) Spectral clustering with conductivity matrix
(4) Spectral clustering with Laplacian matrix (equivalent to Ng *et al.* [14])
(5) K-means with the cosine function as a distance measure

Both k-means and k-lines are used to cluster the spectral images, where applicable.

Last, a method based on Genetic Programming (GP) [20] is assessed as a kernel selection method. GP is a class of evolutionary algorithms (see [21] for an introduction) that aims at learning rules from data. Evolutionary algorithms are a broad class of search and optimization algorithms mimicking an evolutionary process, i.e., potential solutions are randomly initialized, and then mutated and recombined over a number of "generations". This process is illustrated in Fig. 2. Unlike genetic algorithms or evolution strategies, which work with bit strings or real numbers, GP performs a *symbolic* optimization of combinations of mathematical and logical operators.

In GP, every individual, or potential solution (in this case, a kernel), is represented by a tree. Each node of the tree is an operator ($+$, $\times$, $\div$, $exp\,()$, $min\,()$, $max\,()$, etc.). Fig. 3 shows an example of a GP individual.

A population of trees is first initialized, each of them corresponding to a ten-

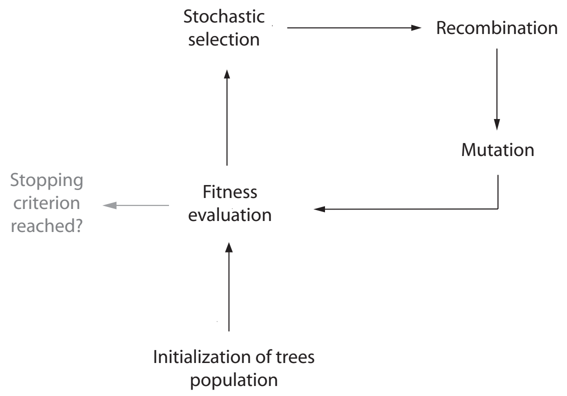
Stochastic selection
Recombination
Mutation
Fitness evaluation
Stopping criterion reached?
Initialization of trees population

Fig. 2. General scheme of an evolutionary algorithm.

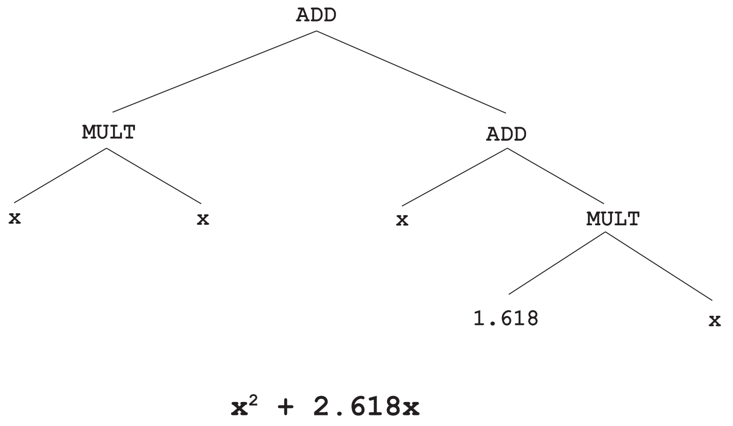

Fig. 3. Example of GP individual.

tative kernel. This population is then evaluated, i.e., each tree is given a performance measure - called the *fitness* in the context of GP - proportional to its ability to render the data suitable for clustering. A recombination step is then performed: pairs of trees are drawn with a probability proportional to their fitness and recombined, i.e., they exchange tree segments, in order to form new trees that hopefully combine good characteristics of their “parents”. Following this, the trees are mutated with small probability (adjunction or removal of random tree segments). The mutation step is carried on in order to prevent the optimization process to fall into a local minimum. The new trees, forming the new generation, are fed into the next loop of the algorithm until a stopping criterion has been reached (maximum number of generations or a certain number of generations without improvement).

The method is similar to that used for classification by Gagné *et al.* [22] and Howley and Madden [23,24]. However, the fitness measure has been customized in order to deal with unsupervised learning problems. The approach we have devised is based on the preservation of local neighborhood after reduction of dimensionality, i.e., after the projection in the space of nonlinear principal components. The method can be summarized as follows:

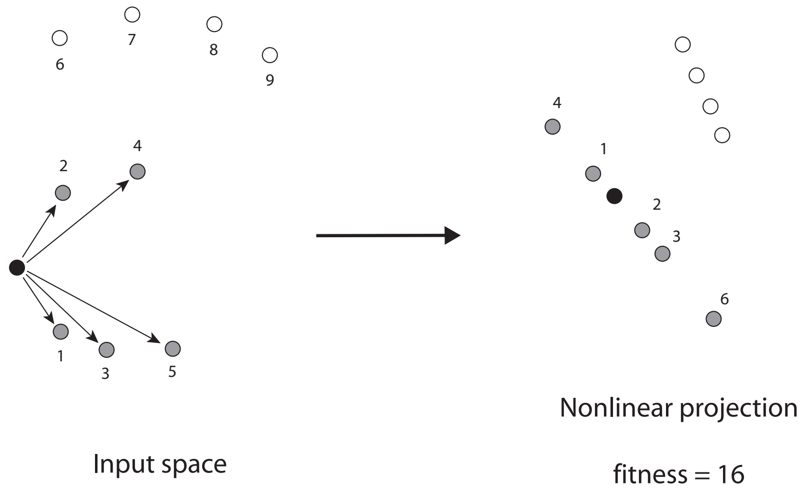

Fig. 4. Fitness measure of the genetic programming algorithm. The aim is to minimize the sum of the ranks of the nearest neighbors of the data points after projection on the nonlinear principal components induced by the kernel to assess. The 5 nearest neighbors in the input space become the 1st, 2nd, 3rd, 4th and 6th nearest neighbors. The fitness value is thus 16.

Given D a dataset and k a kernel function.

```
Compute the $n$-nearest neighbors of each data point.
Compute $K$, the Gram matrix corresponding to $k$.
Perform kernel PCA using $K$ and compute the reduced coordinates of each point in $D$.
sum=0
for $i = 1$ to $n$ do
    Compute the new rank of the $i$-th nearest neighbor from input space, $rank_i$.
    $sum = sum + rank_i$
end for
```

Figure 4 illustrates the performance measure of the method.

3 Experimental

3.1 Data description and statistics

The two studied datasets consist of the major chemical constituents of heroin and cocaine in powdered form, coming from street seizures. These constituents are listed in Table 1 and 2. In these tables, the mean and standard deviation of the raw data are given. The proportions of these variables have been estimated

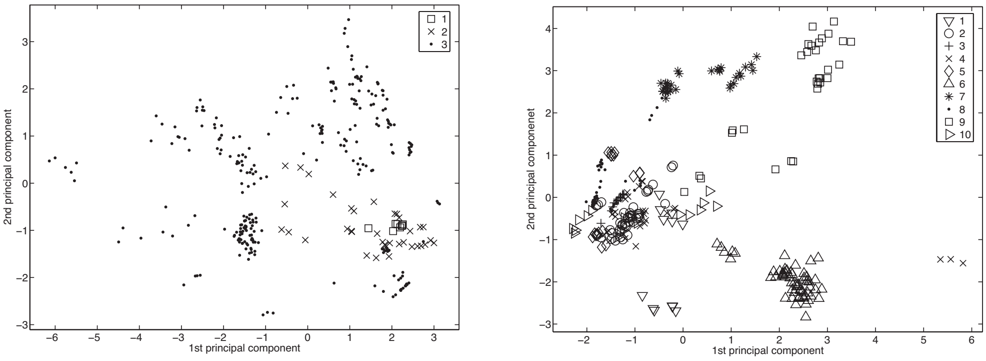

Fig. 5. Projection on the two first principal components of heroin (left) and cocaine (right). The shape of the clusters in the space spanned by the two components is not regular. In the case of heroin, the data exhibit different cluster scales.

by using the area under the peak of the chromatogram corresponding to each of the substances, after removal of the background noise. Note that the large standard deviations for cocaine are due to the fact that for several constituents, the value is zero for a large part of the samples.

A unique characteristic of these datasets is the fact that for many of the samples corresponding to known investigations, the police has confirmed links between them. In other words, two samples are part of the same class within the datasets if the police investigation has linked those two samples to the same case. These "true" class labels being available, it is thus possible to evaluate the performance of various clustering methods. Our aim is to assess whether or not it is possible, to a certain extent, to trace back the links (as confirmed by the police) between seizures on the basis on the samples' chemical composition. Since no links are usually known *a priori*, clustering methods are more useful from a practical point of view than classification methods.

Fig. 5 shows the datasets projected on their two first principal components, in order to give an indication of the type of clusters that might be encountered. These figures show that the clusters vary in shape, which makes the problem very difficult. This could be expected, since the cluster labelling corresponds to networks of people involved in drug trafficking, while the data corresponds to chemical constituents. It is thus of no surprise that the correlation between chemical profiles may not always matches the links found by investigation, since two persons linked within a trafficking network do not necessarily share products that have the same chemical profile. Finding methods that would highlight chemical clusters within those networks is therefore of great interest. Finally, Fig. 6 shows boxplots of the data. Reduced variables (divided by their standard deviation, but not centered) are used for visualization reasons. Indeed, the difference of scales between variable makes raw data hard to visualize altogether.

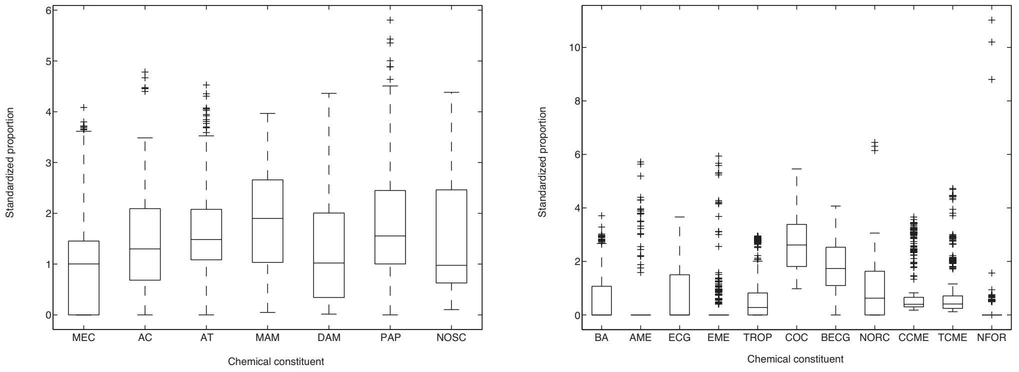

Fig. 6. Boxplots of reduced data for heroin (left) and cocaine (right) samples.

The main purpose of this methodology is to assess and possibly improve the current profiling techniques used in chemical drug profiling. As said previously, profiling is often made using the cosine function, i.e., the correlation between samples, which is formally equivalent to a linear kernel applied on normalized data:

$$cos\left(x_i, x_j\right) = \frac{x_i \cdot x_j}{\|x_i\| \, \|x_j\|} = k\left(x_i, x_j\right). \tag{8}$$

The links confirmed by the police allows to validate if the profiling using a linear kernel between the samples has any meaning at all from a drug intelligence perspective. If not, we will try to pinpoint methods that yield clusters corresponding to actual drug trafficking clusters. This would mean that the chemical signature of the sample *can* reveal something about the network it comes from.

Major constituents of heroin samples	$\mu \times 10^4$	$\sigma \times 10^4$
Meconin	1.6	1.4
Acetylcodein	17	11
Acetylthebaol	4.4	2.7
Monoacetylmorphine	43	24
Diacetylmorphine	155	120
Papaverine	11.3	6.6
Noscapine	40	29

Table 1
The seven major constituents of heroin samples, along with their mean μ and standard deviation σ.

Major constituents of cocaine samples	$\mu \times 10^4$	$\sigma \times 10^4$
Benzoic acid	0.4	0.7
Anhydroecgonine methyl ester	0.1	0.4
Trans-cinnamic acid	0.0	0.0
Anhydroecgonine	0.0	0.0
Ecgonine	0.6	0.9
Ecgonine methyl ester	0.5	1.3
Tropacocaine	2.9	4.3
Cocaine	493	182
Benzoylecgonine	3.2	1.7
Norcocaine	0.9	1.1
Cis-cinnamoylecgonine methyl ester	15	17
Trans-cinnamoylecgonine methyl ester	12	14
N-formylcocaine	0.3	2.0

Table 2
The thirteen major constituents of cocaine samples (11 actually measured in the studied dataset), along with their mean μ and standard deviation σ. The large standard deviations are due to the fact that for several constituents, the value is zero for a large part of the samples. Trans-cinnamic acid and anhydroecgonine being present only in the form of traces, they will not be considered for the remainder of the study.

The data with available true class labels consists of 323 heroin samples (7 variables, 3 classes) and 300 cocaine samples (11 variables, 10 classes). Each of the methods described earlier has been tested, including the GP spectral clustering. In every case, both k-means and k-lines are applied on the standardized coordinates.

3.2 Sample preparation and gas chromatographic analysis

The analyses were performed on a Perkin-Elmer Autosystem gas chromatograph with flame ionisation detection (FID) and equipped with a split/splitless injection system. The procedures for sample preparation and gas chromatographic analysis are described in more details in [7] and [9] for heroin and cocaine, respectively.

The homogeneity was verified by sampling three times each seizure. Reproducibilities of results (expressed as a standard error with a confidence threshold of 99%) were within 1% for all measured compounds (three replicates twice analysed, corresponding to two injections per replicates, i.e., six analyses per sample, with blanks between concentrations to establish that carryover had not occurred). Detection limits did not exceed 0.024 mg (i.e., 0.3% of the sample weight).

3.3 Software

All experiments have been performed in Matlab. The GP code relies on the C++ framework Open BEAGLE [25] and has been customized by the authors to suit the clustering problem. The Matlab code for asymmetric and symmetric spectral clustering has been kindly provided by Prof. Igor Fischer.

3.4 Parameter settings and clustering error

For the methods requiring a scaling parameter σ, an "optimal" value has been assigned using line search. The range of possible values of σ has been sampled at regular intervals and, given that we know the true class labels, the cluster assignment error has been calculated for each σ value and the one yielding the minimum error has been retained. The cluster assignment error E can be expressed as

$$E\left(C, C^{true}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(i\right) \tag{9}$$

where

$$\delta\left(i\right) = \begin{cases} 0 \text{ if } C\left(i\right) = C^{true}\left(i\right) \\ 1 \text{ if } C\left(i\right) \neq C^{true}\left(i\right) \end{cases} \tag{10}$$

C and C^{true} are respectively the obtained and the true assignment vectors and N is the number of data points. The error thus corresponds to the proportion of points that are assigned to the wrong cluster. Of course, as the cluster numbering is arbitrary (each cluster can be renumbered arbitrarily by a particular algorithm from 1 to K, where K is the number of clusters), the error is evaluated for every possible renumbering of the cluster assignment vector, and the minimum error is retained. For example, in a two-class clustering problem, cluster 1 may be renumbered cluster 2 by the method being used. A method achieving a perfect clustering might then lead to an error of 100%, while the error is in fact 0%.

4 Results and discussion

Table 3 summarizes the results obtained for both datasets. Each experiment has been repeated ten times, with the same σ values. The average error is given, along with the corresponding standard deviation.

As it could be expected, the cluster assignment error is far from zero. Indeed, the databases are small and, for the cocaine dataset, the number of clusters is high with respect to this size. Nonetheless, some very interesting conclusions can be drawn from these results. Moreover, it is worth noting that the results for cocaine are more than acceptable, since an error of 26% (the smallest error attained) on a 10 clusters problem is fair for a problem of this difficulty, as the "random guess error rate" grows with the number of clusters. For the heroin dataset, the smallest error obtained is 15%, which is also largely acceptable. These results indicate that it is very likely that the chemical composition of street drug seizures contains information about the drug trafficking networks, since we can find clusters in chemical composition data that are similar to some extent to clusters corresponding to police links.

Method	**Heroin** **3 clusters**	σ_H	**Cocaine** **10 clusters**	σ_C
K-means	0.50 ± 0.03	-	0.28 ± 0.06	-
K-lines	0.62 ± 0.00	-	0.26 ± 0.00	-
Laplac. k-means	0.31 ± 0.14	0.71	0.44 ± 0.06	0.64
Laplac. k-lines	0.16 ± 0.00	0.71	0.37 ± 0.00	0.64
Conduct. k-means	0.16 ± 0.00	0.005	0.48 ± 0.06	0.63
Conduct. k-lines	0.15 ± 0.00	0.005	0.38 ± 0.00	0.63
Asymm. k-means	0.35 ± 0.00	-	0.44 ± 0.05	-
Asymm. k-lines	0.36 ± 0.00	-	0.36 ± 0.00	-
GP k-means	0.49 ± 0.00	-	0.32 ± 0.03	-
GP k-lines	0.41 ± 0.00	-	0.40 ± 0.00	-
Lin. kern. k-means	0.59 ± 0.00	-	0.46 ± 0.01	-
Lin. kern. k-lines	0.51 ± 0.00	-	0.48 ± 0.00	

Table 3
Results for the two datasets. Experiments have been repeated ten times. Simple methods seem efficient on cocaine data, while heroin data requires more sophisticated methods, such as the spectral clustering with conductivity matrix.

From Table 3, we see that using a linear kernel as an affinity measure induces a very large clustering error, especially for heroin, which had very oddly-shaped clusters. For this dataset, the best algorithms were without any doubt the spectral clustering with conductivity matrix (both with k-means and k-lines), and the "Laplacian" spectral clustering with k-lines. Contrarily to what was expected, asymmetric k-means and GP - the two "adaptive" methods - did not perform very well. The worst methods have been k-lines and k-means alone, and the spectral clustering with a linear kernel.

Regarding the cocaine dataset, results are significantly different. The two best methods have been k-lines and k-means alone, with GP with k-means following not far behind. The methods based on the conductivity or Laplacian matrix have not performed well on this dataset. These results suggest that the clusters in cocaine data are much more closer to a simple shape than those of heroin.

Even though further field expertise is necessary to interpret correctly these results, common hypotheses in drug intelligence seem to be confirmed. Indeed, it has been suggested [6] that the heroin market is highly structured, while cocaine comes from numerous independent sources, which makes the market appear less structured. In the latter case we would expect to observe clusters that approach a Gaussian shape, as a consequence of the central limit theorem, and results tend to go in that direction. A highly structured market such as heroin would tend to make the data move away from a Gaussian distribution, and indeed, results show that Gaussian-based methods do not work well for heroin, as opposed to cocaine.

These experiments show that, when using the major chemical constituents as features, spectral methods have a great potential in heroin data, while methods working directly in the input space provide better results in the case of cocaine. These hypothesis will have to be confirmed with larger databases, when those will be available. GP-based clustering showed average results, but may provide more interesting results with larger datasets. Indeed, since it is a highly data-dependent method, if more data is available, its performance can be expected to increase accordingly.

5 Conclusion

In this paper, numerous clustering methods have been compared on labeled heroin and cocaine data. It has been shown that the methods behave very differently on the two datasets. Heroin data has been efficiently clustered using spectral clustering methods based on the Laplacian and conductivity matrix. Spectral clustering with local scaling and GP-based clustering provided intermediate results between the latter and simple methods such as k-means or cosine similarity. Regarding the cocaine dataset, k-means and k-lines outperformed more sophisticated algorithms, with the GP-based algorithm following.

This suggests that even though the data comprises 10 clusters, these are more easily identifiable to compact structures. These facts seem to confirm common hypotheses in the field of drug intelligence. In each case, the results obtained with cosine distance (linear kernel) have been improved and we have shown that information about trafficking networks may be found in the chemical composition of drug seizures.

An interesting extension of this research work is to approach this problem with semi-supervised learning, which uses both labeled and unlabeled data for training classifiers. Indeed, a large database of unlabeled data is available (around 10000 samples). In the context of a classification task, it is possible to use the unlabeled data in order to model the underlying data distribution and improve the classification boundaries provided by the labeled (and usually small) dataset. As for the genetic programming method, other fitness functions could be considered, which might improve the clustering accuracy.

Finally, our current research includes the problem of novelty detection, i.e., detecting inputs that come from a new class rather than an existing one. This would allow determining whether a seizure is likely to be linked to a previously analyzed seizure or not.

6 Acknowledgements

Authors would like to thank Mr. Sylvain Ioset for the preparation of the database.

References

[1] F. Ratle, A.L. Terrettaz-Zufferey, M. Kanevski, P. Esseiva, O. Ribaux, Pattern analysis in illicit heroin seizures: a novel application of machine learning algorithms, *Proc. of the 14th European Symposium on Artificial Neural Networks*, d-side publi., 2006.

[2] F. Ratle, A.L. Terrettaz-Zufferey, M. Kanevski, P. Esseiva, O. Ribaux, Learning manifolds in forensic data, *Proc. of the 16th Int. Conf. on Artificial Neural Networks*, Springer, 2006.

[3] B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* **10**:1299-1319, 1998.

[4] J. Tenenbaum, V. de Silva and J. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, **290**:2319-2323, 2000.

[5] S. Roweis and L. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290**:2323-2326, 2000.

[6] O. Guéniat, P. Esseiva, *Le Profilage de l'Héroïne et de la Cocaïne*, Presses polytechniques et universitaires romandes, Lausanne, 2005.

[7] P. Esseiva, L. Dujourdy, F. Anglada, F. Taroni, P. Margot, A methodology for illicit heroin seizures comparison in a drug intelligence perspective using large databases, *Forensic Science International*, **132**:139-152, 2003.

[8] P. Esseiva, F. Anglada, L. Dujourdy, F. Taroni, P. Margot, E. Du Pasquier, M. Dawson, C. Roux, P. Doble, Chemical profiling and classification of illicit heroin by principal component analysis, calculation of inter sample correlation and artificial neural networks, *Talanta*, **67**:360-367, 2005.

[9] S. Lociciro, P. Hayoz, P. Esseiva, L. Dujourdy, F. Besacier, P. Margot, Cocaine profiling for strategic intelligence purposes, a cross-border project between France and Switzerland: Part I. Optimisation and harmonisation of the profiling method, *Forensic Science International*, **167**:220-228, 2007.

[10] M.G. Madden, A.G. Ryder, Machine Learning Methods for Quantitative Analysis of Raman Spectroscopy Data, In Proceedings of the *International Society for Optical Engineering* (SPIE 2002), **4876**:1130-1139, 2002.

[11] M.L. O'Connell, T. Howley, A.G. Ryder, M.G. Madden, Classification of a target analyte in solid mixtures using principal component analysis, support vector machines, and Raman spectroscopy, In Proceedings of the *International Society for Optical Engineering* (SPIE 2005), **4876**:340-350, 2005.

[12] J. Shawe-Taylor, N. Cristianini. *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[13] A. Karatzoglou, A. Smola, K. Hornik, Kernlab: an S4 package for kernel methods in R, 2006.

[14] A. Ng, M.I. Jordan, Y. Weiss. On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems 14*, 2002.

[15] I. Fischer and J. Poland, New methods for spectral clustering, Technical report no. IDSIA-12-04, IDSIA, 2004.

[16] I. Fischer and J. Poland, Amplifying the block matrix structure for spectral clustering, Technical report no. IDSIA-03-05, IDSIA, 2005.

[17] L. Zelnik-Manor and P. Perona, Self-tuning spectral clustering, *Advances in Neural Information Processing Systems 16*, 2004.

[18] Y. Bengio, O. Delalleau, N. Le Roux, J.F. Paiement, P. Vincent and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA, *Neural Computation* **16**,2004.

[19] N. Cristianini, J. Kandola, A. Elisseeff, J. Shawe-Taylor. On kernel target alignment, *Journal of Machine Learning Research* **1**, 2002.

[20] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

[21] A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.

[22] C. Gagné, M. Schoenauer, M. Sebag M. Tomassini, Genetic programming for kernel-based learning with co-evolving subsets selection, In Proc. of the *Ninth Int. Conference on Parallel Problem Solving from Nature* (PPSN IX), 2006.

[23] T. Howley and M.G. Madden, The genetic kernel support vector machine: description and evaluation, *Artificial Intelligence Review* **24**: 379-395, 2005.

[24] T. Howley and M.G. Madden, An evolutionary approach to automatic kernel construction, *Proc. of the 16th Int. Conf. on Artificial Neural Networks*, Springer, 2006.

[25] C. Gagné and M. Parizeau, Genericity in evolutionary computation software tools: Principles and case-study, *Int. Journal on Artificial Intelligence Tools* **15**:173-194, 2006.