



# Slope Heuristics: Overview and Implementation

Jean-Patrick Baudry, Cathy Maugis, Bertrand Michel

## ► To cite this version:

Jean-Patrick Baudry, Cathy Maugis, Bertrand Michel. Slope Heuristics: Overview and Implementation. 2010. hal-00461639

**HAL Id: hal-00461639**

**<https://hal.science/hal-00461639>**

Submitted on 5 Mar 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Slope Heuristics: Overview and Implementation*

Jean-Patrick Baudry — Cathy Maudis — Bertrand Michel

N° 7223

Mars 2010

Thème COG

A large blue rectangle occupies the lower half of the page. Overlaid on the left side of this rectangle is a large, light grey stylized letter 'R'. To the right of the 'R', the words 'Rapport de recherche' are written in a grey serif font. A horizontal grey brushstroke underline is positioned below the text.

*Rapport  
de recherche*



## Slope Heuristics: Overview and Implementation

Jean-Patrick Baudry <sup>\*</sup>, Cathy Maugis <sup>†</sup>, Bertrand Michel <sup>‡</sup>

Thème COG — Systèmes cognitifs  
Équipe-Projet SELECT

Rapport de recherche n° 7223 — Mars 2010 — 30 pages

**Abstract:** Model selection is a general paradigm which includes many statistical problems. One of the most fruitful and popular approaches to carry it out is the minimization of a penalized criterion. Birgé and Massart (2006) have proposed a promising data-driven method to calibrate such criteria whose penalties are known up to a multiplicative factor: the “slope heuristics”. Theoretical works validate this heuristic method in some situations and several papers report a promising practical behavior in various frameworks. The purpose of this work is twofold. First, an introduction to the slope heuristics and an overview of the theoretical and practical results about it are presented. Second, we focus on the practical difficulties occurring for applying the slope heuristics. A new practical approach is carried out and compared to the standard dimension jump method. All the practical solutions discussed in this paper in different frameworks are implemented and brought together in a Matlab graphical user interface called CAPUSHE.

**Key-words:** Slope Heuristics, Data-driven slope estimation, Dimension jump

<sup>\*</sup> Université Paris-Sud 11, INRIA Projet SELECT, Laboratoire MAP5, Université Paris Descartes and CNRS

<sup>†</sup> Institut de Mathématiques de Toulouse, INSA de Toulouse, Université de Toulouse

<sup>‡</sup> Laboratoire de Statistique Théorique et Appliquée, Université Paris 6

## L'heuristique de pente : vue d'ensemble et mise en pratique

**Résumé :** La sélection de modèles est un paradigme général incluant de nombreux problèmes statistiques. Une des approches les plus connues est fondée sur la minimisation d'un critère pénalisé. Birgé et Massart (2006) ont proposé une méthode pour calibrer de tels critères pour des pénalités connues à une constante multiplicative près : l'heuristique de pente. Cette heuristique est validée par des travaux théoriques dans quelques situations et son bon comportement en pratique est validé dans une plus large variété de contextes. L'objectif de cet article est double. Nous proposons tout d'abord une synthèse des travaux théoriques et pratiques concernant cette méthode. Puis, nous nous intéressons aux difficultés rencontrées lors de la mise en pratique de cette heuristique. Nous proposons une nouvelle approche qui sera comparée à la méthode standard du "saut de dimension". Toutes les solutions pratiques discutées dans cet article dans différents contextes sont disponibles dans CAPUSHE, une interface graphique développée sous le logiciel Matlab.

**Mots-clés :** Heuristique de pente, estimation de pente, saut de dimension

## 1 Introduction

The model selection paradigm relies on the choice of a risk function, depending on the aim, to be minimized. Since its expected value is typically unknown (as the sample distribution), it is estimated through its empirical version. This is a generalization of popular estimation approaches such as maximum likelihood and least squares. Now, if a model collection is considered, the model minimizing the risk can be chosen based on the data. This model selection should not be based only on the empirical risk minimum value in each model since an optimistic bias for the evaluation of the expected risk would be introduced. Thus a feature of the models, called *statistical complexity*, has to be considered in order to enable the control of this bias. Typically the complexity corresponds to the model dimension when the model is a linear vector space but in some situations, the complexity choice is neither intuitive nor easy. The more complex the model, the larger the bias for the estimation of the expected risk: a popular solution is then to select the model minimizing the empirical risk minimum value penalized with an increasing function of the complexity. Thus the aim is to provide a sensible penalty function.

In this paper, we focus on the so-called *slope heuristics* method proposed by Birgé and Massart (2001, 2006). This procedure allows to calibrate penalty functions known up to a multiplicative constant. The purpose of this paper is twofold: providing an overview on the slope heuristics theoretical and practical available works and providing practical solutions for its implementation.

From a theoretical point of view, the principle of the slope heuristics is introduced and proved for the first time by Birgé and Massart (2006) in the context of Gaussian homoscedastic least squares regression with fixed design. They show that there exists a *minimal* penalty, namely such that the complexity and the risk of models selected with lighter penalties explode. Moreover, they prove that *considering a penalty equal to twice this minimal penalty allows to select a model close to the oracle model in term of risk* (see Section 2 for a reminder about the definition of the oracle). This rule of thumb is the main statement of the slope heuristics. Birgé and Massart (2006) then propose to estimate the minimal penalty in a data-driven manner, which enables us to overcome the difficulty that some constants needed to design the penalty are unknown and to deduce an optimal penalty from this estimate. In the framework they consider, the penalty shape they derive is proportional to the model dimension when the model family is not too large and involves an additional logarithmic term when the model family is huge. Arlot and Massart (2009) extend these results to the heteroscedastic regression with random design framework without Gaussian assumption. They have to restrict the considered models to histograms but conjecture that this is only due to technical reasons and that the heuristics remains valid in other least squares regression frameworks. They consider the case of reasonably rich model families (namely the number of models grows as a power of  $n$ ) and derive penalties depending of the dimension. In a density estimation framework, Lerasle (2009c) validates the slope heuristics and proves oracle inequalities for both independent (Lerasle, 2009b) and mixing data (Lerasle, 2009a). Some theoretical results by Verzelen (2009) partially validate the slope heuristics in a Gaussian Markov random field framework.

Moreover the conjecture that the slope heuristics may be valid in a wider range of model selection frameworks is supported by the results of several encouraging applications: estimation of oil reserves (Lepez, 2002); change-point detection in a Gaussian least squares framework (Lebarbier, 2005); selection of the number of

non-zero mean components in a Gaussian framework with application to genomics (Villers, 2007); simultaneous variable selection and clustering in a Gaussian mixture models setting with applications to the study of oil production through curve clustering and to genomics (Maugis and Michel, 2010); selection of the suitable neighborhood in a Gaussian Markov random field framework (Verzelen, 2009); estimation of the number of interior knots in a B-spline regression model (Denis and Molinari, 2009); choice of a simplicial complex in the computational geometry field (Caillerie and Michel, 2009) and simulations in both the frameworks of Gaussian mixture models likelihood and model-based clustering (Baudry, 2009). This enumeration illustrates that the slope heuristics brings solution to real needs and the good results reported in those simulated and real data experiments contribute to confirm its usefulness. This is enthusiastic evidence on how fruitful are these efforts to fill the gap between the theory of non asymptotic model selection and the practical applications.

In practice, the penalty is known up to a multiplicative constant (e.g. proportional to the noise level) and the main issue is to estimate the minimal penalty. The most studied and applied approach is the so-called *dimension jump*. This method is based on a complexity jump and, in the first studied frameworks, the complexity was actually the model dimension. It consists of considering the complexity of the selected model as a function of the multiplicative constant in the penalty. Then increasing the constant value from 0, a nonincreasing and piecewise function is obtained. The minimal penalty is calibrated with the constant corresponding to the greatest jump of complexity or to the first jump greater than a chosen threshold. The choice of the threshold (or of the most complex models involved) is delicate and may be decisive for the final model selection. Another approach is based on the expectation that a linear relation exists between the penalty shape and the contrast value for the most complex models. This method called *data-driven slope estimation* in this paper consists of estimating the slope of this linear part for calibrating the minimal penalty. A new strategy based on graphical methods to apply this second approach is proposed in order to answer to practical difficulties. It notably has the advantage to validate that the slope heuristics can be applied. The dimension jump and the data-driven slope estimation approaches, presented in this paper, are implemented in a Matlab graphical user interface called CAPUSHE (CALibrated Penalty Using Slope HEuristics). Hopefully it will contribute to a better understanding and a wider use of the slope heuristics.

In Section 2, principles for the contrast minimization and model selection paradigm are reminded, and the theoretical basis of the slope heuristics are presented. The dimension jump approach is presented in Section 3. Section 4 is devoted to the data-driven slope estimation approach and our strategy. Finally, Section 5 illustrates the results obtained by those approaches through the package CAPUSHE for various problems.

## 2 Contrast minimization and the slope heuristics

Before discussing the calibration issue of model selection via penalization, the estimation method by contrast minimization is briefly recalled.

Let  $\mathbf{X} = (X_1, \dots, X_n)$ ,  $X_i \in \mathbb{R}^d$ , be an i.i.d sample from an unknown probability distribution. The quantity of interest, denoted as  $s$ , is related to the unknown sample distribution and belongs to a set  $\mathcal{S}$ . The method is based on the existence

of a *contrast* function  $\gamma : \mathcal{S} \times \mathbb{R}^d \rightarrow \mathbb{R}$  fulfilling the fundamental property that

$$s = \operatorname{argmin}_{t \in \mathcal{S}} \mathbb{E}_X [\gamma(t, X)],$$

where the expectation is taken with respect to  $X$  distributed as the sample (the minimum is expected to be uniquely reached). The associated *loss function*, which enables us to evaluate each element of  $\mathcal{S}$ , is defined by:

$$\forall t \in \mathcal{S}, l(s, t) = \mathbb{E}_X [\gamma(t, X)] - \mathbb{E}_X [\gamma(s, X)].$$

The target  $s$  being a minimizer of the contrast over  $\mathcal{S}$ , it is also a minimizer over  $\mathcal{S}$  of the expectation of the *empirical contrast* defined by

$$\forall t \in \mathcal{S}, \gamma_n(t) = \frac{1}{n} \sum_{i=1}^n \gamma(t, X_i).$$

A minimizer of the empirical contrast over a model  $S$ , a subspace of  $\mathcal{S}$ , is then considered and denoted as  $\hat{s}$ . Substituting the empirical contrast  $\gamma_n$  to its expectation and minimizing  $\gamma_n$  over  $S$ , it is expected that  $\hat{s}$  is a sensible estimator of  $s$ . The quality of such an estimator can be measured by its *risk*

$$\mathcal{R}(\hat{s}) = \mathbb{E}_X [l(s, \hat{s})].$$

For instance, in the density estimation framework, the popular maximum likelihood and least squares estimators are both minimum contrast estimators. Suppose that the sample has a density  $s$  with respect to a measure  $\mu$ . Then the contrast

$$\gamma(t, x) = -\ln[t(x)]$$

where  $t$  denotes another density is the maximum likelihood contrast. The corresponding loss function is the Kullback-Leibler divergence defined by

$$\text{KL}(s, t) = \int s \ln\left(\frac{s}{t}\right) d\mu.$$

If  $s$  is supposed to be in  $\mathbb{L}_2(\mu)$  then the contrast

$$\gamma(t, x) = \|t\|^2 - 2t(x)$$

where  $\|\cdot\|$  denotes the norm in  $\mathbb{L}_2(\mu)$  is the least squares contrast. The corresponding loss function is then given by

$$l(s, t) = \|s - t\|^2.$$

Other examples of contrasts for regression, classification and Gaussian white noise can be found in the book of Massart (2007).

## 2.1 Model selection via penalization

A countable collection of models  $(S_m)_{m \in \mathcal{M}}$  with the corresponding estimator collection  $(\hat{s}_m)_{m \in \mathcal{M}}$  is now considered. An important question is how to choose the “best” estimator among this collection? Let  $\hat{m}$  be the model selected by a given model selection procedure. The selected estimator is then  $\hat{s}_{\hat{m}}$ , where both  $\hat{s}_m$



(for any  $m$ ) and  $\hat{m}$  are built from the same sample  $\mathbf{X}$ . Such a procedure may be evaluated from either an asymptotic or a non asymptotic point of view.

The ideal model  $S_{m^*}$  for a given  $n$  and a given dataset is such that

$$m^* \in \operatorname{argmin}_{m \in \mathcal{M}} l(s, \hat{s}_m). \quad (1)$$

However the corresponding estimator  $\hat{s}_{m^*}$ , called the *oracle*, depends on the unknown sample distribution. Nevertheless, this oracle is a benchmark while building a model selection procedure.

From a non asymptotic point of view, the model collection  $\mathcal{M}$  may depend on  $n$ , and the aim is to build a model selection procedure such that the selected model  $\hat{m}$  is *optimal*. More precisely, it fulfills an oracle inequality:

$$l(s, \hat{s}_{\hat{m}}) \leq C_n l(s, \hat{s}_{m^*}) + \eta_n$$

with  $C_n$  as close to 1 as possible and  $\eta_n$  a remainder term negligible with respect to  $l(s, \hat{s}_m)$ . This inequality is expected to hold either with high probability or in expected value, or even, when such results are too difficult to be achieved, under a weaker form:

$$\mathbb{E}_{\mathbf{X}} [l(s, \hat{s}_{\hat{m}})] \leq C_n \inf_{m \in \mathcal{M}} \mathbb{E}_{\mathbf{X}} [l(s, \hat{s}_m)] + \eta_n.$$

Let us stress that even if there exists  $m_0$  such that  $s \in S_{m_0}$ , there is no reason that  $m^* = m_0$ , since  $m^*$  has to take the model complexity into account. The loss can be decomposed into an approximation and an estimation parts

$$l(s, \hat{s}_m) = l(s, s_m) + \mathbb{E}_X [\gamma(s_m, X) - \gamma(\hat{s}_m, X)],$$

where  $s_m$ , a minimizer of  $\mathbb{E}_X [\gamma(t, X)]$  over  $S_m$ , is one of the best approximation of  $s$  in  $S_m$ . This illustrates that a bias/variance trade-off has to be reached.

The main approaches to design such model selection procedures are hold-out and cross-validation procedures (see Arlot and Celisse, 2009), or penalized criteria. Nevertheless, cross-validation procedures are time consuming and thus penalization is preferable in many cases. Penalization consists of defining a proper penalty function  $\text{pen} : \mathcal{M} \rightarrow \mathbb{R}^+$  and of selecting  $\hat{m}$  minimizing the associated penalized criterion

$$\forall m \in \mathcal{M}, \text{crit}(m) = \gamma_n(\hat{s}_m) + \text{pen}(m). \quad (2)$$

Choosing the penalty is tricky but obviously crucial. Some well-known penalized criteria with fixed penalties such as AIC (Akaike, 1973) or BIC (Schwarz, 1978) have been widely studied (Burnham and Anderson, 2002). The use of these penalties is mainly motivated by asymptotic arguments that may be wrong in a non asymptotic context. In the regression framework, other famous penalized criteria are Mallows's  $C_p$  (Mallows, 1973) and GCV (Craven and Wahba, 1978). Nevertheless, Mallows's  $C_p$  depends on the noise level  $\sigma^2$  of the true regression model which is unknown (if it does exist) and  $\sigma^2$  is thus difficult to estimate. Similarly, GCV depends on a tuning parameter whose best value is actually  $\sigma^2$ . The solution proposed by the GCV method is to choose this tuning parameter from the data via cross-validation, and once again an unknown parameter has to be estimated.

More recent works based on concentration inequalities have led to optimal penalties which are known up to a multiplicative constant  $\kappa$ . In this framework,

the penalty shape is then denoted as  $\text{pen}_{\text{shape}}(\cdot)$  and an unknown constant  $\kappa_{\text{opt}}$  exists such that

$$\text{pen}_{\text{opt}} : m \in \mathcal{M} \mapsto \kappa_{\text{opt}} \text{pen}_{\text{shape}}(m) \quad (3)$$

is an optimal penalty. Two different kinds of results usually lead to such a penalty shape:

- **Deterministic penalty shapes.** Specific deterministic functions  $m \mapsto \text{pen}_{\text{shape}}(m)$  can be used to define an optimal penalty (see Massart, 2007, for some examples of such penalties). For instance, in a general maximum likelihood framework, Theorem 7.11 in Massart (2007) provides a solution to choose a penalty shape and insures the existence of a constant  $\kappa_{\text{opt}}$  such that  $\text{pen}_{\text{opt}}(\cdot) = \kappa_{\text{opt}} \text{pen}_{\text{shape}}(\cdot)$  follows an oracle inequality. The value of  $\kappa_{\text{opt}}$  which can be derived from the theory is much too pessimistic and a reasonable value has to be guessed from the data.
- **Resampling penalty shapes.** In a regression framework, Arlot (2009) uses resampling to design the penalty corresponding to each model and derives non asymptotic results for the corresponding procedures. These penalties actually have to be calibrated by a multiplicative constant. Lerasle (2009b) provides analogous results in a density estimation framework.

*Remark 1.* Note that such a situation where an optimal penalty is known up to a multiplicative constant also arises with usual asymptotic criteria. For example, Mallows'  $C_p$ , known to be asymptotically optimal in a fixed design and homoscedastic regression framework, relies on the penalty  $\frac{2\sigma^2 D_m}{n}$ , where  $D_m$  is the dimension of the model  $S_m$ . The variance being typically unknown, a value estimated from the data can be plugged in the penalty. Another possibility consists of considering  $\frac{2D_m}{n}$  as a penalty shape and of guessing a good multiplicative constant from the data.

## 2.2 Slope Heuristics

Recently, some efforts have been paid to overcome the difficulty of penalty calibration. Birgé and Massart (2006) propose a practical method based on theoretical and heuristic ideas for defining efficient penalty functions from the data. This so-called slope heuristics is validated in the framework of Gaussian regression with a homoscedastic fixed design (Birgé and Massart, 2006) and generalized in the heteroscedastic random-design case (Arlot and Massart, 2009). It has also been validated for least squares density estimation (Lerasle, 2009b) and has been partially validated for the selection of a suitable neighborhood in a Gaussian Markov random field framework (Verzelen, 2009). Furthermore, its practical validity has been illustrated in many other different frameworks as cited in the introduction.

If the penalty is defined by the function  $m \in \mathcal{M} \mapsto l(s, \hat{s}_m) - \gamma_n(\hat{s}_m)$ , then according to (1) and (2), the oracle model is always selected. Such a penalty can be decomposed into

$$\begin{aligned} l(s, \hat{s}_m) - \gamma_n(\hat{s}_m) &= \mathbb{E}_X [\gamma(\hat{s}_m, X)] - \mathbb{E}_X [\gamma(s_m, X)] \\ &\quad + \mathbb{E}_X [\gamma(s_m, X)] - \mathbb{E}_X [\gamma(s, X)] \\ &\quad + \gamma_n(s_m) - \gamma_n(\hat{s}_m) \\ &\quad + \gamma_n(s) - \gamma_n(s_m) \\ &\quad - \gamma_n(s). \end{aligned} \quad (4)$$

Since  $-\gamma_n(s)$  does not depend on  $m$ , the *ideal penalty* can be defined as

$$\text{pen}^*(m) = v_m + \hat{v}_m + \Delta_n(s_m)$$

where  $v_m = \mathbb{E}_X [\gamma(\hat{s}_m, X) - \gamma(s_m, X)]$  is a “variance” term,  $\hat{v}_m = \gamma_n(s_m) - \gamma_n(\hat{s}_m)$  is an empirical “variance” term and  $\Delta_n(s_m) = \{\mathbb{E}_X [\gamma(s_m, X)] - \mathbb{E}_X [\gamma(s, X)]\} + \{\gamma_n(s) - \gamma_n(s_m)\}$  is the difference between a “bias” term and the associated empirical “bias” term. Now the main idea is to estimate this ideal penalty from the data so as to build an optimal penalty function. For all  $m \in \mathcal{M}$ ,

$$l(s, \hat{s}_m) + \gamma_n(s) = \gamma_n(\hat{s}_m) + \text{pen}^*(m)$$

according to the expression of the ideal penalty (4). Moreover for any penalty function  $\text{pen}(\cdot)$ ,

$$\forall m \in \mathcal{M}, \gamma_n(\hat{s}_{\hat{m}}) + \text{pen}(\hat{m}) \leq \gamma_n(\hat{s}_m) + \text{pen}(m)$$

according to the definition of  $\hat{m}$  and (2). This leads to

$$l(s, \hat{s}_{\hat{m}}) + [\text{pen}(\hat{m}) - \text{pen}^*(\hat{m})] \leq \inf_{m \in \mathcal{M}} \{l(s, \hat{s}_m) + [\text{pen}(m) - \text{pen}^*(m)]\}. \quad (5)$$

Thus it is relevant to look for a penalty close to the ideal penalty for any  $m$  in order to derive an oracle inequality. To this aim, the slope heuristics relies on the two following points [SH1] and [SH2].

**[SH1] Minimal penalty.** If the chosen penalty function is  $\text{pen}(m) = \hat{v}_m$ , the penalized criterion is

$$\begin{aligned} \text{crit}(m) &= \gamma_n(\hat{s}_m) + \hat{v}_m \\ &= \gamma_n(s_m), \end{aligned}$$

which concentrates around its expectation  $\mathbb{E}_X [\gamma(s_m, X)]$  for large  $n$ . Hence, this procedure selects a model minimizing the bias. The variance is not taken into account: such a criterion has high probability of selecting a too complex model. If the chosen penalty is  $\text{pen}(m) = \kappa \hat{v}_m$ , the criterion can be written as

$$\text{crit}(m) = (1 - \kappa)\gamma_n(\hat{s}_m) + \kappa\gamma_n(s_m).$$

Therefore two main cases occur:

- if  $\kappa < 1$  then the criterion decreases as the complexity increases (the two terms being decreasing): the selected model is for sure one of the most complex ones,
- if  $\kappa > 1$ , for the most complex models, the criterion increases with the complexity since these models have almost the same bias, and thus they are ruled out.

This suggests that  $\text{pen}_{\min}(m) = \hat{v}_m$  is a *minimal* penalty, namely that lighter penalties give rise to a selection of the most complex models, whereas higher penalties should select models with “reasonable” complexity. This phenomenon corresponds to the first point of the slope heuristics.

**[SH2] Ideal penalty: twice minimal penalty.** The first point of the slope heuristics is to assume that  $v_m \approx \hat{v}_m$ . One reason to believe in such an assumption is that  $\hat{v}_m$  is the empirical counterpart of  $v_m$ . Then, since it is expected that the fluctuations of  $\Delta_n(s_m)$  around its zero expectation can be controlled through concentration results, the ideal penalty may be approximated as follows:

$$\begin{aligned} \text{pen}^*(m) &\approx v_m + \hat{v}_m \\ &\approx 2\hat{v}_m. \end{aligned}$$

Hence the ideal penalty is about twice the minimal penalty, which is the second point of the slope heuristics.

In practice, this heuristics is useful when an optimal penalty  $\text{pen}_{\text{opt}}(\cdot) = \kappa_{\text{opt}} \text{pen}_{\text{shape}}(\cdot)$  is known up to a multiplicative factor. Note that the slope heuristics is derived by considering the ideal penalty, whereas it is applied to a particular penalty shape chosen by the user. Thus, it is not necessarily guaranteed that the ideal penalty itself is of the shape  $\kappa^* \text{pen}_{\text{shape}}(\cdot)$ . This is a further assumption that a given optimal penalty fulfills the same properties as the ideal penalty, namely that half this optimal penalty is a minimal penalty. This relies on the assumption that the chosen penalty shape is fine enough so that the derived optimal penalty is close to the ideal penalty. Thus the keystone of the slope heuristics is that  $\frac{\kappa_{\text{opt}}}{2} \text{pen}_{\text{shape}}(m)$  is a good estimate of  $\hat{v}_m$  and provides a minimal penalty.

For the two application methods of the slope heuristics presented in Sections 3 and 4, it is assumed that a complexity measure  $C_m$  of the models is given. This complexity measure, depending on the framework, is typically the model dimension or the number of free parameters in parametric frameworks. Mostly the penalty shape can be written as a function of  $C_m$ . When its definition is no obvious a priori, the complexity measure can be chosen as the penalty shape itself (as in Caillerie and Michel, 2009). The penalty shape can also be guessed itself from the data, for example with resampling penalties. Table 1 gives complexities  $C_m$  and  $\text{pen}_{\text{shape}}$  for a large list of model selection works.

For the two application methods of the slope heuristics presented in Sections 3 and 4, it is assumed that:

- (C1) The empirical contrast  $\gamma_n(\hat{s}_m)$  decreases with the complexity  $C_m$ .
- (C2) The penalty shape  $\text{pen}_{\text{shape}}(\cdot)$  increases with the complexity  $C_m$ .

The two methods differ by the way the minimal penalty involved in point [SH1] is estimated. The first one is the so-called *dimension jump* method introduced in Birgé and Massart (2006). The second one consists of directly estimating the “slope”  $\kappa_{\text{opt}}$  in a data-driven fashion.

*Remark 2.* Besides numerical issues while computing  $\hat{s}_m$ , condition (C1) is satisfied for instance with nested models along which the complexity increases.

## 3 Dimension jump

### 3.1 Principle

The so-called *dimension jump* is a method for penalty calibration which takes advantage of [SH1] and [SH2] to efficiently determine the unknown penalty constant  $\kappa_{\text{opt}}$  in (3). Let  $m(\kappa)$  be the model selected by the penalized criterion

Framework	References	Model complexity: $C_m =$	$\text{pen}_{\text{shape}}(\mathbf{m}) =$
Gaussian linear models: <ul style="list-style-type: none"> <li>• few models</li> <li>• complete variable selection</li> <li>• change-point detection</li> </ul>	<ul style="list-style-type: none"> <li>• Birgé and Massart (2001)</li> <li>• Birgé and Massart (2001)</li> <li>• Lebarbier (2005)</li> </ul>	<ul style="list-style-type: none"> <li>• <math>D_m</math> : model dimension</li> <li>• <math>D_m</math> : number of variables</li> <li>• <math>D_m</math> : partition size</li> </ul>	<ul style="list-style-type: none"> <li>• <math>D_m</math></li> <li>• <math>D_m \left\{ 1 + \sqrt{2[1 + \ln(2N/D_m)]} \right\}^2</math> where <math>N</math> is the total number of variables</li> <li>• <math>D_m \{1 + c \ln(n/D_m)\}</math></li> </ul>
Histogram density estimation: <ul style="list-style-type: none"> <li>• regular histograms</li> <li>• complete family of irregular histograms</li> </ul>	Castellan (2000) or Massart (2007, chap 7.3)	$D_m$ : number of bins minus one	<ul style="list-style-type: none"> <li>• <math>D_m</math></li> <li>• <math>D_m [1 + c \ln(N/D_m)]</math> where <math>N</math> is the maximum number of bins</li> </ul>
Gaussian mixture models	Maugis and Michel (2009), Baudry (2009)	$D_m$ : number of parameters of the mixture	$D_m$
Regressogram selection	Arlot (2009)	$D_m$ or $\text{pen}_{\text{shape}}(m)$	by resampling
Least squares density estimation	Lerasle (2009b)	$\text{pen}_{\text{shape}}(m)$	by resampling
Neighborhood selection of Gaussian field	Verzelen (2009)	$d_m$ : size of the support neighborhood	$d_m/(p_1 p_2)$ where $p_1 \times p_2$ is the size of the rectangular lattice.
Spline regression	Denis and Molinari (2009)	$D_m = k + d + 1$ where $k$ is the number of knots and $d$ is the spline degree	$D_m$
Graph selection	Caillerie and Michel (2009)	$\text{pen}_{\text{shape}}(m)$	logarithm of the graph length

Table 1: Model complexities  $C_m$  and penalty shape functions  $\text{pen}_{\text{shape}}(\cdot)$  for a non exhaustive list of model selection works.

$m \mapsto \gamma_n(\hat{s}_m) + \kappa \text{pen}_{\text{shape}}(m)$ . Under (C1) and (C2),  $\kappa \mapsto C_{m(\kappa)}$  is a nonincreasing and piecewise constant function. According to the minimal penalty definition, it is expected that the selected model  $m(\kappa)$  has a large complexity when  $\kappa \text{pen}_{\text{shape}}(\cdot) < \text{pen}_{\min}(\cdot)$  and a reasonably large complexity if  $\kappa \text{pen}_{\text{shape}}(\cdot) > \text{pen}_{\min}(\cdot)$ . Thus,  $\kappa \mapsto C_{m(\kappa)}$  should present an abrupt jump around a value  $\hat{\kappa}$  (see Figure 1). The penalty  $\hat{\kappa} \text{pen}_{\text{shape}}(\cdot)$  is then expected to be close to the minimal penalty and according to [SH2], the penalty  $2\hat{\kappa} \text{pen}_{\text{shape}}(\cdot)$  is expected to be an optimal penalty ( $\kappa_{\text{opt}} \approx 2\hat{\kappa}$ ).

As a matter of course, the choice of complexity measure is crucial for this method (see Section 5.3). If several complexity measures seem relevant for the user, they can all be tested to find the one that shows the clearest jump.

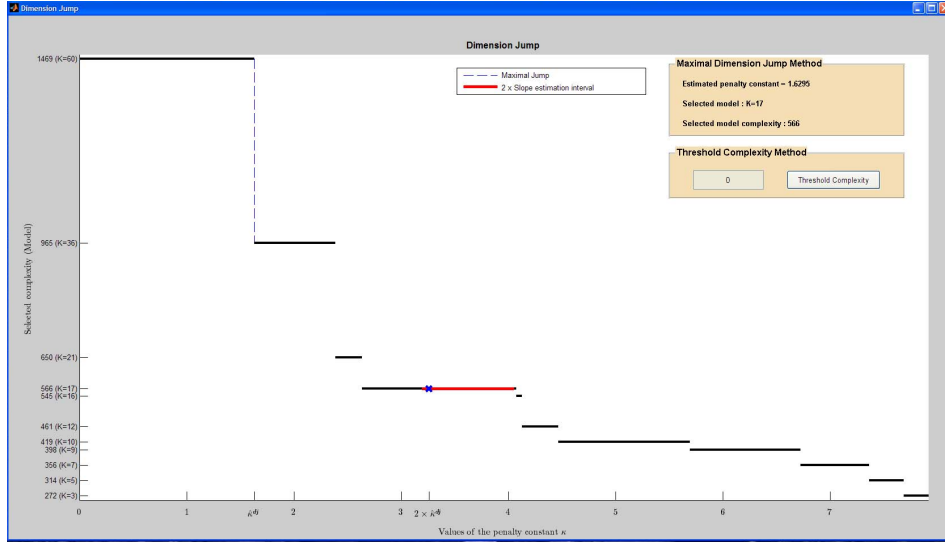


Figure 1: Representation of the nonincreasing and piecewise constant function  $\kappa \mapsto C_{m(\kappa)}$ .

### 3.2 The dimension jump method in practice

In order to apply the dimension jump method, the following steps have to be proceeded:

1. Compute

$$\forall \kappa > 0, m(\kappa) \in \operatorname{argmin}_{m \in \mathcal{M}} \{ \gamma_n(\hat{s}_m) + \kappa \text{pen}_{\text{shape}}(m) \};$$

2. Find  $\hat{\kappa}$  such that  $C_{m(\kappa)}$  is large if  $\kappa < \hat{\kappa}$  and has a “reasonable” order otherwise.
3. Select  $\hat{m} = m(2\hat{\kappa})$ .

For the first step, the algorithm proposed by Arlot and Massart (2009) is implemented in our graphical interface CAPUSHE. This algorithm makes this first step computationally tractable since it only requires at most  $\text{card}(\mathcal{M}) - 1$  steps, and actually probably much less. This provides the location of jumps, namely an increasing sequence  $(\kappa_i)_{0 \leq i \leq i_{\max}}$  with  $\kappa_0 = 0$ ,  $\kappa_{i_{\max}} = +\infty$ , the number of jumps  $i_{\max} \in \{1, \dots, \text{card}(\mathcal{M}) - 1\}$ , and the associated selected model sequence  $(m_i)_{0 \leq i \leq i_{\max}}$  where  $m_i = m(\kappa_i)$  for all  $\kappa$  in  $[\kappa_i, \kappa_{i+1})$  and for all  $i < i_{\max}$ .

For the second step, two different strategies are available in CAPUSHE:

- **Maximal dimension jump**

This first method is the most popular. It consists of choosing the constant, denoted as  $\hat{\kappa}^{\text{dj}}$ , corresponding to the greatest jump of complexity:

$$\hat{\kappa}^{\text{dj}} = \kappa_{i_{\text{dj}}}$$

where  $i_{\text{dj}} \in \underset{0 \leq i \leq i_{\max}-1}{\operatorname{argmax}} \{C_{m_{i+1}} - C_{m_i}\}$ . If several values of  $\kappa$  reach the maximum value, Lebarbier (2005) suggests to choose the largest  $\kappa$  in order to select the less complex model.

- **Threshold complexity**

The second method, proposed by Arlot and Massart (2009), consists of choosing a threshold complexity  $C_{\text{thresh}}$  such that complexities smaller than  $C_{\text{thresh}}$  are reasonable but larger ones are not. Then the chosen constant denoted as  $\hat{\kappa}^{\text{thresh}}$  is the smallest value of  $\kappa$  for which the corresponding penalty selects a complexity smaller than  $C_{\text{thresh}}$ :

$$\hat{\kappa}^{\text{thresh}} = \inf\{\kappa > 0 : C_{m(\kappa)} \leq C_{\text{thresh}}\}.$$

In the regression framework, these authors suggest to choose  $C_{\text{thresh}}$  of order  $\frac{n}{\log n}$  or  $\frac{n}{(\log n)^2}$ .

Those alternative methods are not equivalent. Arlot and Massart (2009) expect that they should yield the same selection as the dimension jump is clear or as there are several dimension jumps close to each other, but might not otherwise. They report simulations according to which it could happen quite seldom. When the selected models differ, they recommend that the user looks at the graphic himself.

## 4 Data-driven slope estimation method

### 4.1 Principle

This alternative method consists of directly estimating the constant  $\kappa_{\text{opt}}$  by the “slope” of the expected linear relation of  $-\gamma_n(\hat{s}_m)$  with respect to the penalty shape values  $\text{pen}_{\text{shape}}(m)$ . Currently, this second method is less employed than the dimension jump procedure. This might be due to some difficulties related to its implementation: Lebarbier (2005) partly presents this method and discusses it, but chooses the dimension jump approach notably because of the lack of stability she encountered when trying to estimate the slope. It is also presented and studied in Baudry et al. (2008) and Maugis and Michel (2010). In this section, we propose some solutions so as to make possible and reliable the application of the slope heuristics thanks to a stability study of the selected model.

We recall that the optimal penalty  $\text{pen}_{\text{opt}}(m) = \kappa_{\text{opt}} \text{pen}_{\text{shape}}(m)$  is expected to be close to

$$\begin{aligned} 2\hat{v}_m &= 2[\gamma_n(s_m) - \gamma_n(\hat{s}_m)] \\ &= 2[\gamma_n(s_m) - \gamma_n(s)] + 2[\gamma_n(s) - \gamma_n(\hat{s}_m)]. \end{aligned}$$

The empirical bias term  $\gamma_n(s_m) - \gamma_n(s)$  gets stable for the most complex models for which the approximation of the target  $s$  cannot be appreciably improved. Hence the behavior of  $\kappa_{\text{opt}} \text{pen}_{\text{shape}}(m)$  is known through  $-2\gamma_n(\hat{s}_m)$  for models of large complexities, and thus of large penalty shape values according to (C2). Thus  $-\gamma_n(\hat{s}_m)$  is expected to behave linearly with respect to  $\text{pen}_{\text{shape}}(m)$  with a slope around  $\frac{\kappa_{\text{opt}}}{2}$ , as shown in the left graph of Figure 2. Finally, if  $\hat{\kappa}$  denotes an estimation of the slope of the linear regression of  $-\gamma_n(\hat{s}_m)$  on  $\text{pen}_{\text{shape}}(m)$ , the optimal penalty is estimated by  $2\hat{\kappa} \text{pen}_{\text{shape}}(\cdot)$ .

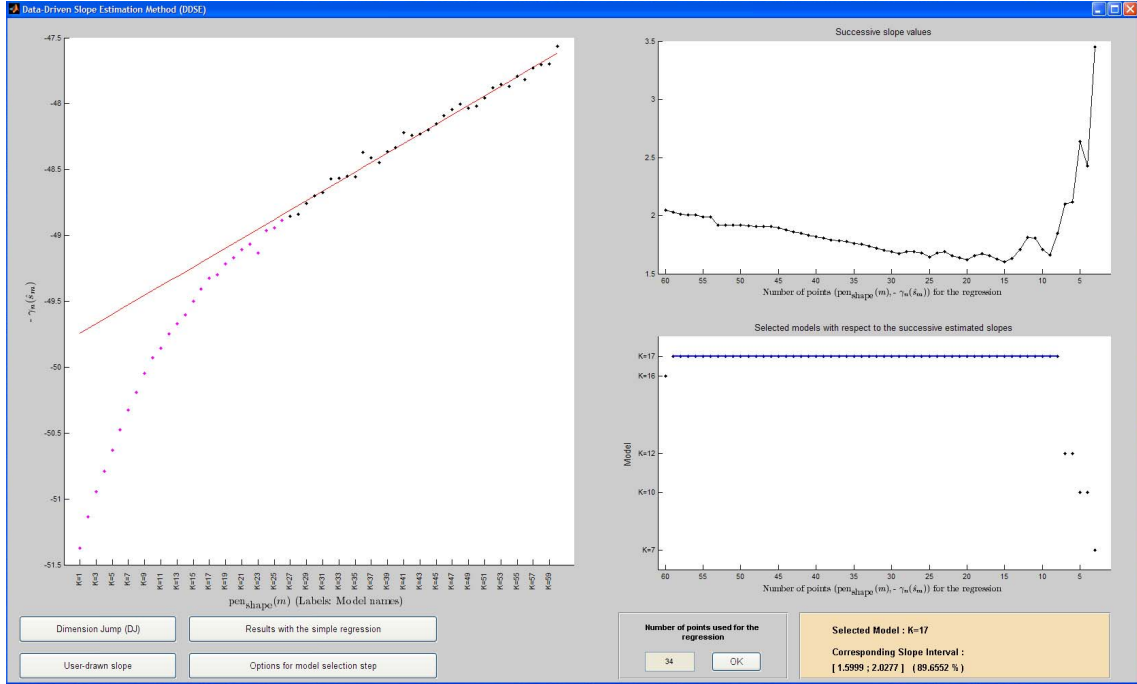


Figure 2: An example of the results given by CAPUSHE. The left graph represents  $-\gamma_n(\hat{s}_m)$  with respect to  $\text{pen}_{\text{shape}}(m)$  allowing to check the linear behavior assumption. The top-right graph gives the successive estimated slope as a function of the number of couples  $(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m))$  used for the linear regression coefficient estimation. The bottom-right graph represents the selected models as a function of the number of points used for the linear regression coefficient estimation. The last plateau for which the length  $N_i$  is greater than  $\text{pct} \sum_{l=1}^I N_l$  is then detected and the corresponding model  $\hat{m}(p_i)$  is selected. The “Corresponding slope interval” given on bottom right is the interval  $[p_i, p_{i+1})$  leading to select this model.



## 4.2 Practice of the data-driven slope estimation method

The main issue about this method is how to choose a subset of points  $(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m))$  corresponding to large values of  $\text{pen}_{\text{shape}}(m)$  where the slope can be estimated. In practice, it is usually chosen at sight. The method proposed in this paper to answer this problem is based on the model selection stabilization. More precisely, the slope is sequentially estimated from the couples  $(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m))$  where the couple with the smallest penalty shape value is removed at each step. An area where the slope estimation is stable has to be observed according to Section 4.1. The slope estimation in this area corresponds to an estimation of  $\kappa_{\text{opt}}/2$  and thus the same model is selected. Denoting  $\mathcal{P} = \{\text{pen}_{\text{shape}}(m), m \in \mathcal{M}\}$ , the corresponding algorithm is decomposed as follows:

**Step 1** If several models in the collection have the same penalty shape value, only the model having the smallest contrast value  $\gamma_n(\hat{s}_m)$  is kept according to (2). To make easier the reading of this algorithm, the model indexation is not modified.

**Step 2** For  $p \in \mathcal{P}$ , the slope  $\hat{\kappa}(p)$  of the linear regression on the couples of points  $\{(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m)); \text{pen}_{\text{shape}}(m) \geq p\}$  is computed using a robust regression method.

**Step 3** For  $p \in \mathcal{P}$ , the model fulfilling the following condition is selected:

$$\hat{m}(p) = \underset{m \in \mathcal{M}}{\text{argmin}} \{ \gamma_n(\hat{s}_m) + 2\hat{\kappa}(p) \text{pen}_{\text{shape}}(m) \}.$$

We obtain an increasing sequence of change-points  $(p_i)_{1 \leq i \leq I+1}$  such that

- $\forall 1 \leq i \leq I-1, \hat{m}(p_i) \neq \hat{m}(p_{i+1})$  ;
- $\forall 1 \leq i \leq I, \forall p \in [p_i, p_{i+1}), \hat{m}(p) = \hat{m}(p_i)$  ;
- $\hat{m}(p_{I+1}) = \hat{m}(p_I)$ .

We observe a “plateau” sequence and compute the plateau sizes  $(N_i)_{1 \leq i \leq I}$  defined by

- $\forall 1 \leq i \leq I-1, N_i = \text{card}\{[p_i, p_{i+1}) \cap \mathcal{P}\}$  ;
- $N_I = \text{card}\{[p_I, p_{I+1}] \cap \mathcal{P}\}$ .

**Step 4** For a given value of  $pct$  (see hereafter) the model  $\hat{m}(p_i)$  such that

$$\hat{i} = \max \left\{ i \in \{1, \dots, I\}; N_i \geq pct \sum_{l=1}^I N_l \right\}$$

is selected. We also return the interval of slope values  $[p_i, p_{i+1})$  and the proportion  $N_i / \sum_{l=1}^I N_l$ . Graphically, this corresponds to selecting the “most to the right” plateau whose length is greater than the threshold (see the bottom-right graph in Figure 2).

This algorithm requires a tuning parameter  $pct$  in Step 4 in order to determine which plateau corresponds to a stabilization of the model selection. By default, this threshold  $pct$  is fixed to 15% in CAPUSHE. This is rather an arbitrary choice, which should be reconsidered with respect to the application at hand. Remark that whatever the choice at this step, the reported actual proportion  $N_i / \sum_{l=1}^{I-1} N_l$  measures the stability of the method.

*Remark 3.* For the successive slope estimations in Step 2, a robust regression (Huber, 1981) is advised in order to attenuate the influence of possible estimation errors of the sequel  $(\hat{s}_m)_{m \in \mathcal{M}}$ . As shown on Figure 3, with the robust regression, the successive estimation of the slopes are more stable and the length of the selected plateau is larger than with classical regression.

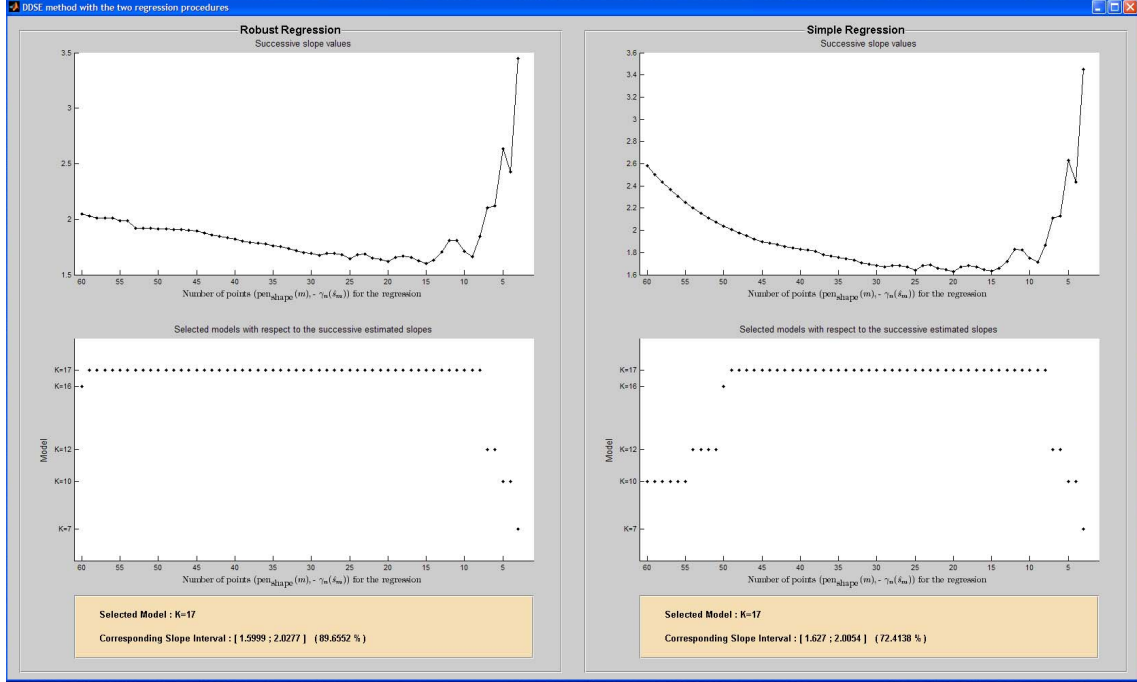


Figure 3: Comparison of the model selection method using the robust regression and the classical linear regression with CAPUSHE. See the description of the right graphs of Figure 2 for more details.

This method is based on a linear relation between  $-\gamma_n(\hat{s}_m)$  and  $\text{pen}_{\text{shape}}(m)$  for the largest values of the penalty shape. Non evidence of such linear relation should warn the user that the slope heuristics should probably not be applied. It should then be verified that enough complex models have been involved in the study and the penalty shape should be questioned. To help the user to validate the linear behavior assumption, some graphical tools are proposed in CAPUSHE. In particular, the use of the “Validation Step” option is illustrated in an example in Section 5.1.2.

## 5 Applications

The aim of this section is to illustrate how the slope heuristics can be proceeded using the Matlab interface CAPUSHE. The practical difficulties encountered and the differences between the dimension jump and our data-driven slope estimation method are highlighted on simulated and real datasets.

## 5.1 Number of Gaussian mixture components

In the model-based clustering framework, assessing the number of components of Gaussian mixtures is a crucial question. In this framework,  $S_m$  is the set of Gaussian mixtures with  $m$  components:

$$S_m = \left\{ \sum_{k=1}^m p_k \Phi(\cdot | \mu_k, \Sigma_k); \begin{array}{l} (p_1, \dots, p_m) \in [0, 1]^m, \sum_{k=1}^m p_k = 1 \\ \mu_k \in \mathbb{R}^d, \Sigma_k \in \mathcal{D}^+ \end{array} \right\}$$

where  $\Phi(\cdot | \mu, \Sigma)$  corresponds to the density of the  $d$ -dimensional Gaussian distribution with mean vector  $\mu \in \mathbb{R}^d$  and covariance matrix  $\Sigma$  belonging to a subset  $\mathcal{D}^+$  of  $d \times d$  positive definite matrices. The maximum likelihood estimators  $\hat{s}_m$  are computed with the EM algorithm using MIXMOD software (Biernacki et al., 2006) or MCLUST (Fraley and Raftery, 2003) for instance. Following Maugis and Michel (2009), we consider a penalized loglikelihood criterion with a penalty proportional to  $\text{pen}_{\text{shape}}(m) = D_m$ , the number of free parameters for a mixture with  $m$  components. Note that this last quantity  $D_m$  is a natural complexity measure of  $S_m$ . In practice, the maximum number of components  $M_{\max}$  of the mixture models has to be chosen first. The model collection is then restricted to  $(S_m)_{1 \leq m \leq M_{\max}}$ .

### 5.1.1 Bubbles experiment

This simulated dataset, plotted in Figure 4, is composed of  $n = 1000$  observations in  $\mathbb{R}^3$ . It consists of an equiprobable mixture of three large “bubble” groups centered at  $\nu_1 = (0, 0, 0)$ ,  $\nu_2 = (6, 0, 0)$  and  $\nu_3 = (0, 6, 0)$  respectively. Each bubble group  $j$  is simulated from a mixture of seven components according to the following density distribution:

$$x \in \mathbb{R}^3 \mapsto 0.4\Phi(x | \mu_1 + \nu_j, I_3) + \sum_{k=2}^7 0.1\Phi(x | \mu_k + \nu_j, 0.1I_3)$$

with  $\mu_1 = (0, 0, 0)$ ,  $\mu_2 = (0, 0, 1.5)$ ,  $\mu_3 = (0, 1.5, 0)$ ,  $\mu_4 = (1.5, 0, 0)$ ,  $\mu_5 = (0, 0, -1.5)$ ,  $\mu_6 = (0, -1.5, 0)$  and  $\mu_7 = (-1.5, 0, 0)$ . Thus the distribution of this dataset is actually a 21-component Gaussian mixture. The reader is referred to Baudry (2009, Chapter 5) for more details. A model collection  $(S_m)_{1 \leq m \leq M_{\max}}$  of spherical Gaussian mixtures is considered with covariance matrices  $\Sigma_k = \lambda_k I_3$  with  $\lambda_k \in \mathbb{R}_+^*$ .

The outputs of CAPUSHE are explained with this simulated example. Figure 5 gives the graphical outputs obtained by the data-driven slope estimation method for the model collection with  $M_{\max} = 50$ . In this example, the linear behavior for the most complex models is clearly observed. Using the robust regression, the true Gaussian mixture distribution with 21 components is selected in 89.6% of the successive slope estimations. The choice of a multiplicative constant  $2\kappa$  in the penalized criterion with  $\kappa \in [1.1783 \cdot 10^{-3}; 2.611 \cdot 10^{-3}]$  leads to select  $\hat{m} = 21$ . In the same way, there is no ambiguity for the result with the dimension jump since the maximal complexity jump is really clear (see Figure 6).

In order to compare the two slope heuristics methods with the classical criteria BIC and AIC, an experiment is conducted with 100 simulations of the bubbles dataset. Model collections with  $M_{\max} = 40$  and  $M_{\max} = 50$  are successively considered. For the data-driven slope estimation method, the default parameter value

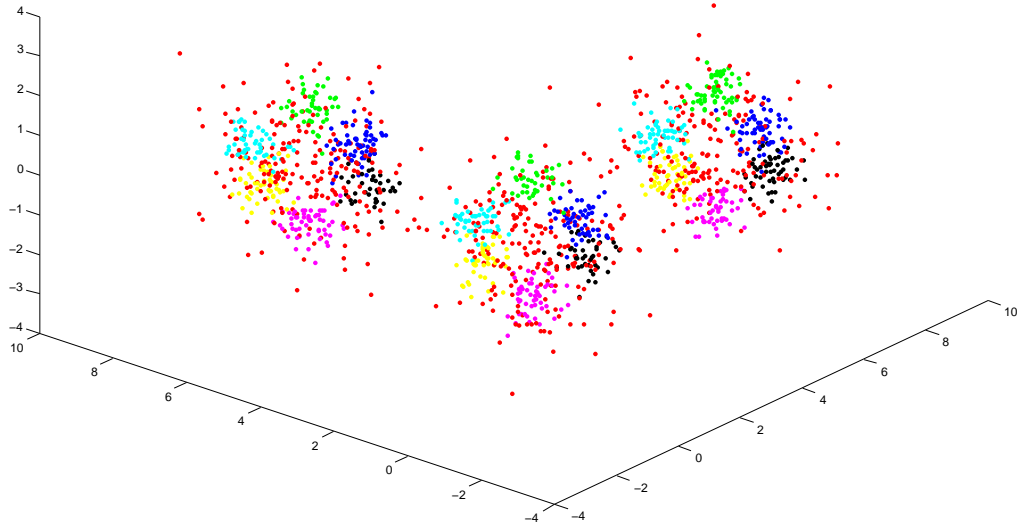


Figure 4: Representation of a bubbles dataset composed of 1000 observations.

Selected number of components $\hat{m}$	$M_{\max}$	3	4	15–18	19	20	21	22	23	24	25	$\geq 35$	Risk ratio
Oracle	40, 50					1	<b>76</b>	15	3	3	2		1
AIC	40, 50											<b>100</b>	2.59
BIC	40, 50			3	6	23	<b>57</b>	9	1	1			1.17
DDSE	50				3	7	<b>59</b>	20	6	3	2		1.06
DDSE	40			1	3	7	<b>61</b>	18	4	4	2		1.09
DJ	50	4	2		3	7	<b>59</b>	18	2	3	2		1.49
DJ	40	<b>28</b>	2		2	4	<b>51</b>	10	2	1			3.27

Table 2: Number of times a model  $m$  is selected among the 100 simulations by AIC, BIC, the data-driven slope estimation method (DDSE) and the dimension jump method (DJ). The last column is the ratio between the risk of the selected estimator by each method and the oracle risk.

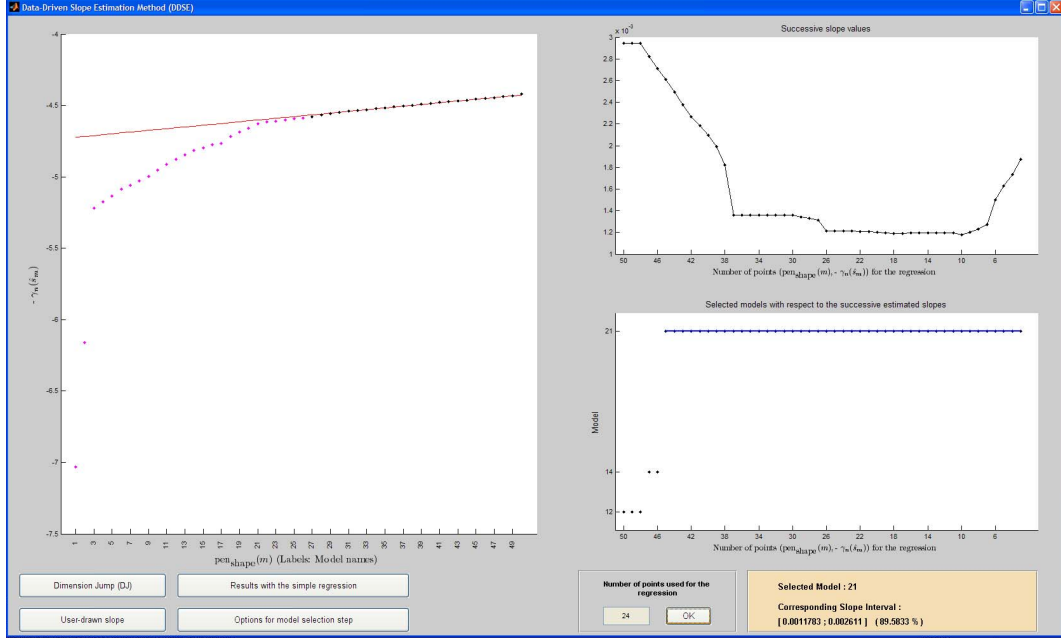


Figure 5: Graphical output obtained by the data-driven slope estimation method for the bubbles experiment with  $M_{\max} = 50$ .

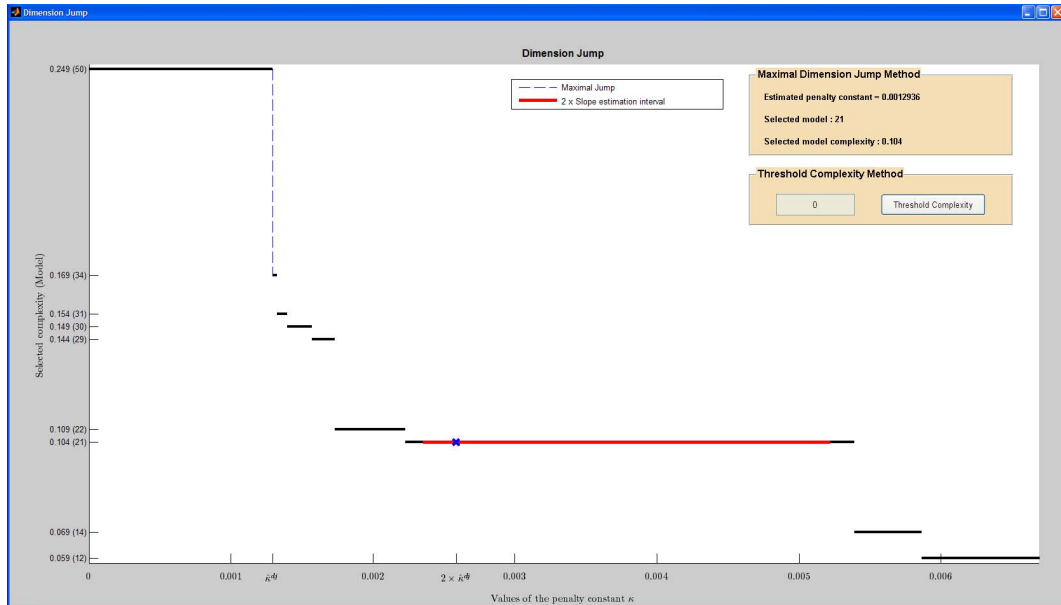


Figure 6: Graphical output obtained by the dimension jump method for the bubbles experiment with  $M_{\max} = 50$ .

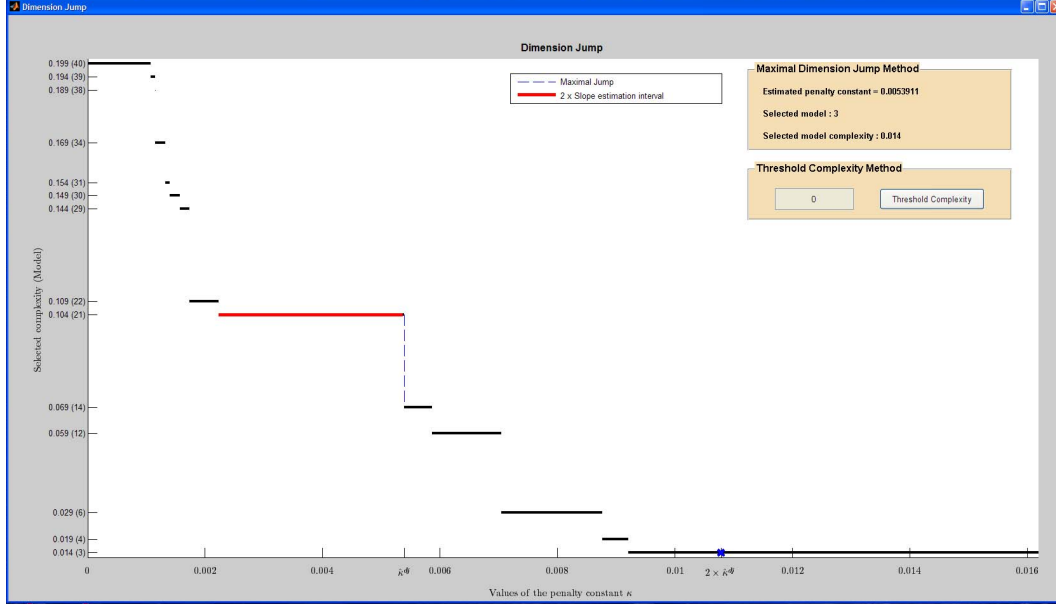


Figure 7: Graphical output obtained by the dimension jump method for the bubbles experiment with  $M_{\max} = 40$ .

$pct = 15\%$  is used. Table 2 gives the number of times a model is selected by each criterion over the 100 simulations. It also provides the ratio between the risk of the selected estimator by each method and the oracle risk. For each simulation, the oracle model is defined as the model which estimator minimizes the Kullback-Leibler divergence to the true density distribution.

Mostly, the oracle is close to the true distribution. As usual in a mixture framework, AIC obviously underpenalizes the model complexity. BIC mostly recovers the true number of components which is not surprising according to Keribin (2000): in this experiment the true distribution belongs to the model collection and  $n$  is quite large. For the model collection with  $M_{\max} = 50$ , the dimension jump approach yields the same selection as the data-driven slope estimation approach, but in 6% of the datasets. As compared to the oracle risk, the slope heuristics applied with the data-driven slope estimation approach gives the best risk results (ratio close to 1), closely followed by BIC. The dimension jump method has a larger risk because it selects sometimes small models.

When  $M_{\max} = 40$ , the results illustrate a difficulty which can be encountered while applying the dimension jump. This approach selects 28 times the model  $\hat{m} = 3$  in the simulation study, which is a poor result. The reason of this difficulty is illustrated in Figure 7: there seemingly occurs a dimension jump for the most complex models, but it occurs in several steps. Therefore the largest of those “sub-jumps” is still smaller than the jump leading to select  $\hat{m} = 3$ , which is quite large because of the data structure. This shows the sensibility of the dimension jump approach to the choice of the most complex models involved in the study. The data-driven slope estimation results are only worsened a little if  $M_{\max} = 40$  instead of  $M_{\max} = 50$ .

### 5.1.2 Transcriptome dataset

The following transcriptome dataset was studied by Maugis et al. (2009). It consists of 1020 genes of *Arabidopsis thaliana* described in 20 experiments. We consider a collection of Gaussian mixtures where the covariance matrices are assumed to be equal:  $\forall k \in \{1, \dots, m\}, \Sigma_k = \Sigma$ . In practice, the choice of  $M_{\max}$  is crucial since the linear behavior of the contrast can be observed for the most complex models. If we consider  $M_{\max} = 20$  in this example, the linear part is not observed and the selection with the data-driven slope estimation method is not satisfying because there is no long and clear plateau (see the top of Figure 8). In order to find a trade-off between the global estimation time and the observation of the linear area, the option “Validation Step” is proposed. This option allows us to graphically test whereas the considered model collection is large enough for applying the data-driven slope estimation method. The slope, estimated on the subset of couples  $\{(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m)); m \in \{1, \dots, M_{\max}\}\}$ , is plotted and the user can graphically test whereas other such couples for larger complex models are in this linear regression line or not. For our transcriptome data example, the points corresponding to mixtures with 40, 50 and 60 components are tested in the bottom of Figure 8. This three points are below and away from the estimated linear line, showing that the choice  $M_{\max} = 20$  is not large enough. For  $M_{\max} = 60$ , the linear area is then clearly observed and the data-driven slope estimation method can be correctly applied according to the graphical outputs given in Figure 2.

## 5.2 Change-point detection

Change-point detection is studied in Lebarbier (2005) with a model selection point of view. This section gets back on a simulation given in Lebarbier (2005) to illustrate the slope heuristics with CAPUSHE in this context.

Let us consider the fixed design regression model

$$y_i = s(u_i) + \varepsilon_i \quad (6)$$

where the  $y_i$ 's are observed at regular points  $u_i = \frac{i}{n}$ ,  $i = 1, \dots, n$ . The errors  $\varepsilon_i$  are assumed to be i.i.d. centered random variables with variance  $\sigma^2$ . Let  $\mathcal{M}$  be the set of all the partitions of the grid  $\{u_1, \dots, u_n\}$ . The model  $S_m$  corresponding to the partition  $m = \{I_k\}_{1 \leq k \leq |m|}$  is defined by

$$S_m = \left\{ \sum_{k=1}^{|m|} \beta_k \mathbb{1}_{I_k}; (\beta_k)_{1 \leq k \leq |m|} \in \mathbb{R}^{|m|} \right\}.$$

A natural measure of the model complexity is the dimension  $D_m = |m|$ , namely the partition size. For each model, a least squares estimator of  $s$  can be defined by minimizing the contrast:  $\gamma(t, (y, u)) = (y - t(u))^2$  over  $S_m$ . The aim is to determine the best estimator for the  $\ell^2$  risk. Lebarbier (2005) shows that a convenient penalty shape for this problem is

$$\text{pen}_{\text{shape}}(m) = \frac{D_m}{n} \left( 2.5 + \ln \frac{n}{D_m} \right).$$

Case (b) of the simulations proposed in Lebarbier (2005) is considered (see Section 4.1.2 in Lebarbier, 2005, for more details). A sample of 300 observations is

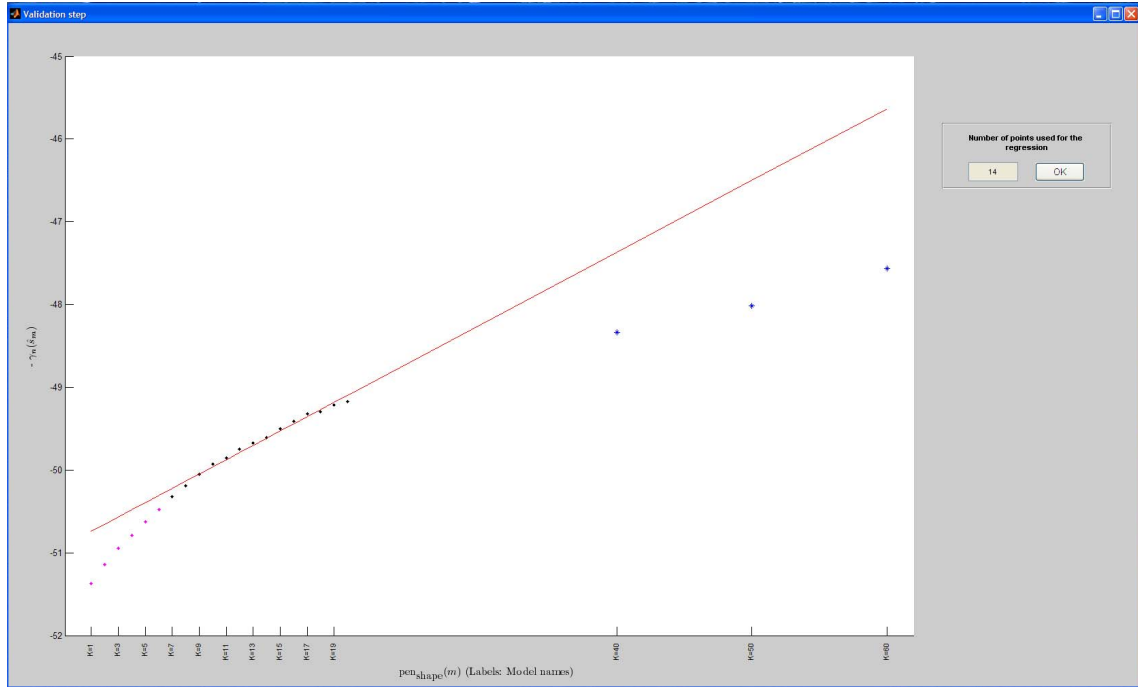
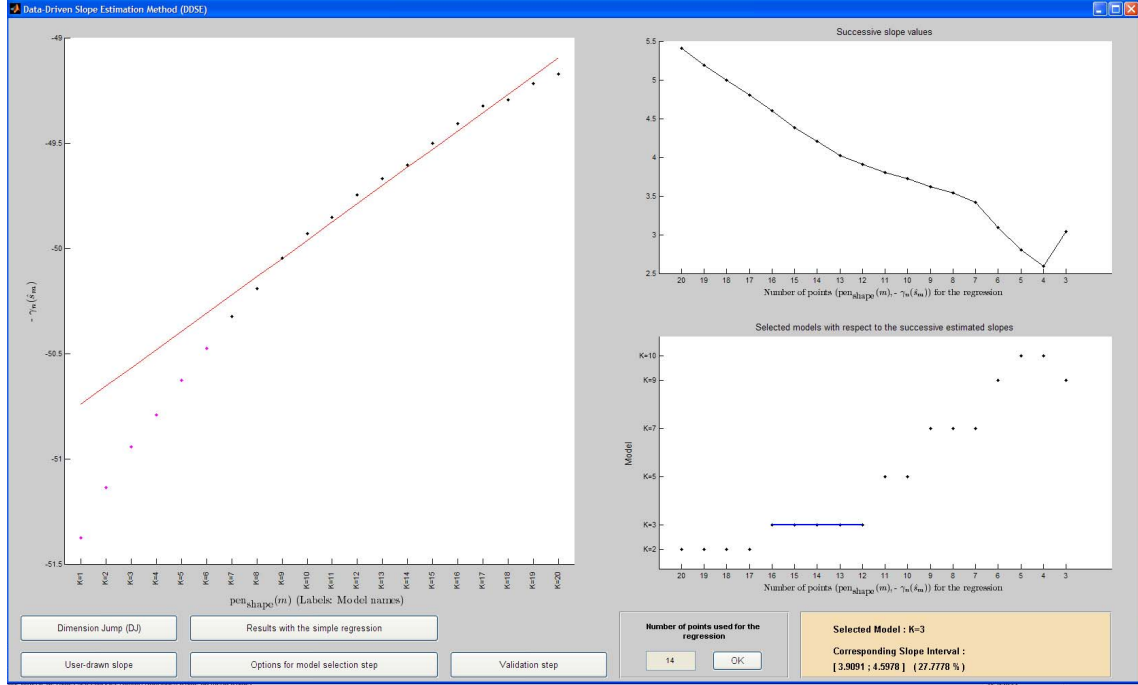


Figure 8: Selection with the data-driven slope estimation method when  $M_{\max} = 20$  for the transcriptome dataset study (top). On the bottom plot, the Validation option is used with three points corresponding to  $m \in \{40, 50, 60\}$  in order to graphically test whether the linear area is reached or not.



simulated according to (6),  $s$  being a piecewise constant function with six pieces and  $\sigma = 1$ . Figure 9 shows the results for the two methods on this sample. A long plateau corresponding to  $|m| = 6$  can be observed on the top graph. The greatest jump also leads to the selection of this model on the bottom graph. Note that on this example, one could think that for the dimension jump method, the greatest jump is actually between  $|m| = 12$  and  $|m| = 22$ . By aggregating the sequence of small and close jumps in this interval, this would yield a different model selection. As for the Bubbles experiment, the data-driven slope estimation method gives a clearer answer to the model selection problem.

### 5.3 Graph selection

In practice, choosing a convenient penalty can be a real issue if no theoretical results are available for the considered problem. In particular, the user may need to choose between several possible penalties. This section illustrates on a geometrical example how bad penalties can be detected thanks to CAPUSHE.

In Caillerie and Michel (2009), a penalized least squares criterion is proposed for fitting simplicial complexes on a cloud of points. A simplicial complex is a collection of simplices such that any two simplices of the collection intersect along a common face if at all, and all the faces of a simplex of the collection belongs to the complex too. Simplicial complexes allow dimension estimation procedures and topological inference methods, see the cited paper for references. For one dimensional complexes, namely for graphs, it is shown in Caillerie and Michel (2009) that the penalty has to be chosen proportional to the logarithm of the graph length. Without knowing these theoretical results, it is not obvious which penalty can be chosen to select a convenient graph in a given collection.

We take up the example of the Lissajous curve simulation proposed in Section 5.1 of Caillerie and Michel (2009). Let  $X_1, \dots, X_n$  be some points in  $\mathbb{R}^2$  sampled in the neighborhood of a Lissajous curve (Figure 10). A nested collection of graphs  $(S_m)_{m \in \mathcal{M}}$  is defined on a set of fixed landmark points belonging to the Lissajous curve. For each  $m$ , the following Gaussian model is considered

$$\forall i = 1, \dots, n \quad X_i = \bar{x}_i + \varepsilon_i$$

where the unknown  $\bar{x}_i$ 's belong to  $S_m$  and the  $\varepsilon_i$ 's are i.i.d centered Gaussian variables with level noise  $\sigma$ . The example consists of  $n = 5000$  observed points and  $\sigma = 0.005$ . For each  $i$  and each  $m$ , the least squares estimator  $\hat{x}_{i,m}$  of  $\bar{x}_i$  is the closest point to  $X_i$  belonging to the graph  $S_m$ . This leads to consider the following penalized criterion to be minimized

$$\text{crit}(m) = SS(m) + \text{pen}(m)$$

where  $SS(m) := \sum_{i=1}^n \|\hat{x}_{i,m} - X_i\|^2$ . Let  $l(m)$  be the length of  $S_m$ . Two different kinds of penalties are considered:  $\text{pen}_{\text{shape}}^{(1)}(m) = l(m)$  and  $\text{pen}_{\text{shape}}^{(2)}(m) = \ln l(m)$  (as in Caillerie and Michel, 2009).

In Figure 11, a large jump can be observed for  $\text{pen}_{\text{shape}}^{(2)}(\cdot)$  whereas it is not the case for  $\text{pen}_{\text{shape}}^{(1)}(\cdot)$ . As shown in Figure 12, the second penalty leads to a better linear behavior of  $SS(\cdot)$  than with the first one and there is no clear stability of the selected model with  $\text{pen}_{\text{shape}}^{(1)}(\cdot)$ . Using the Validation Step option with the last eight points, we can also check that only the regression with  $\text{pen}_{\text{shape}}^{(2)}(\cdot)$  gives

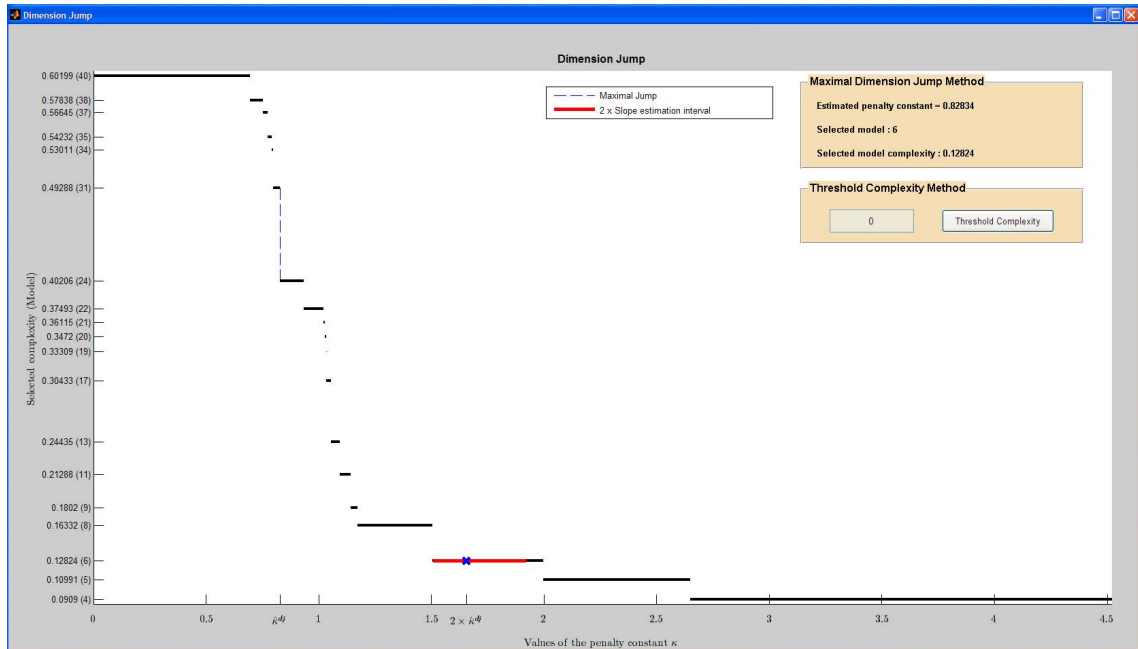
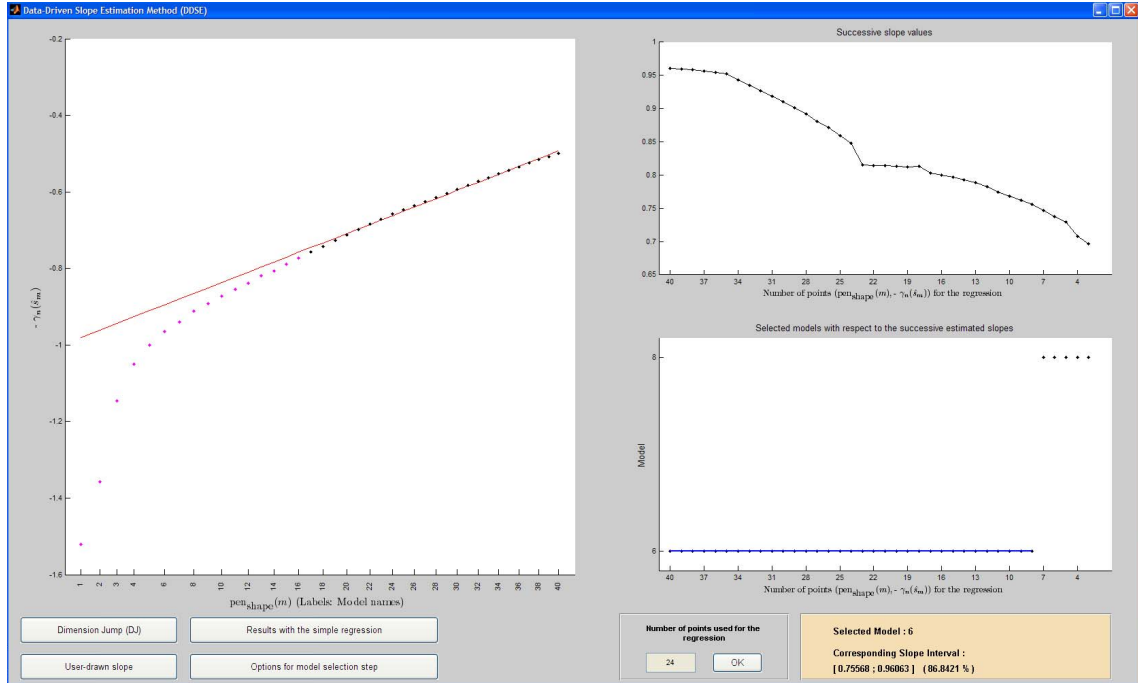


Figure 9: Data-driven slope estimation method (top) and dimension jump (bottom) for the change-point detection problem. The two methods select the same model  $|\hat{m}| = 6$ .

satisfying results (see Figure 13). These last remarks show that in this context the second penalty should be preferred to select a convenient graph.

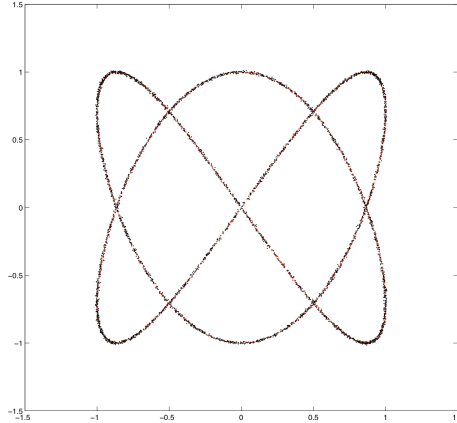


Figure 10: Lissajous curve simulation. Red points are landmarks and black one are observed points.

## 6 Discussion

The slope heuristics is a promising approach for calibrating penalized criteria in model selection contexts. The available theoretical and practical justifications for its use in various frameworks increasing, this paper aims at providing an overview of those theoretical and experimental results.

Although efforts have been paid to fill the gap between the theoretical results on the slope heuristics and its application, the dimension jump method and the data-driven slope estimation method are not totally justified. Regarding the dimension jump, the theory does not say if the dimension jump occurs in one single jump or several successive jumps. Thus, aggregating successive jumps could be an option for future works. Concerning the data-driven slope estimation, note that the linear relation between  $-\gamma_n(\hat{s}_m)$  and  $\text{pen}_{\text{shape}}(m)$  for the largest values of the penalty shape is actually only valid in expectation.

The encountered practical difficulties for applying this heuristics are highlighted and different solutions are compared. We also propose a new method based on data-driven slope estimation which is implemented in the Matlab graphical interface CAPUSHE. Thanks to this graphical tool, it is possible to check that the slope heuristics is valid for a given penalty shape. Moreover it allows the user to compare this method with the more popular dimension jump method. The data-driven slope estimation is easier to calibrate: if both methods involve tuning parameters (choice of the method and parameter to define a “plateau” for the data-driven slope estimation; choice of the most complex involved model or of the complexity threshold for the dimension jump), the choice can be made on a more universal ground in the case of the data-driven slope estimation (for example as a percentage of the total number of involved models). The “Bubbles” experiment moreover illustrates that the data-driven slope estimation may behave better than the dimension jump, notably as the estimation in complex models is expensive.

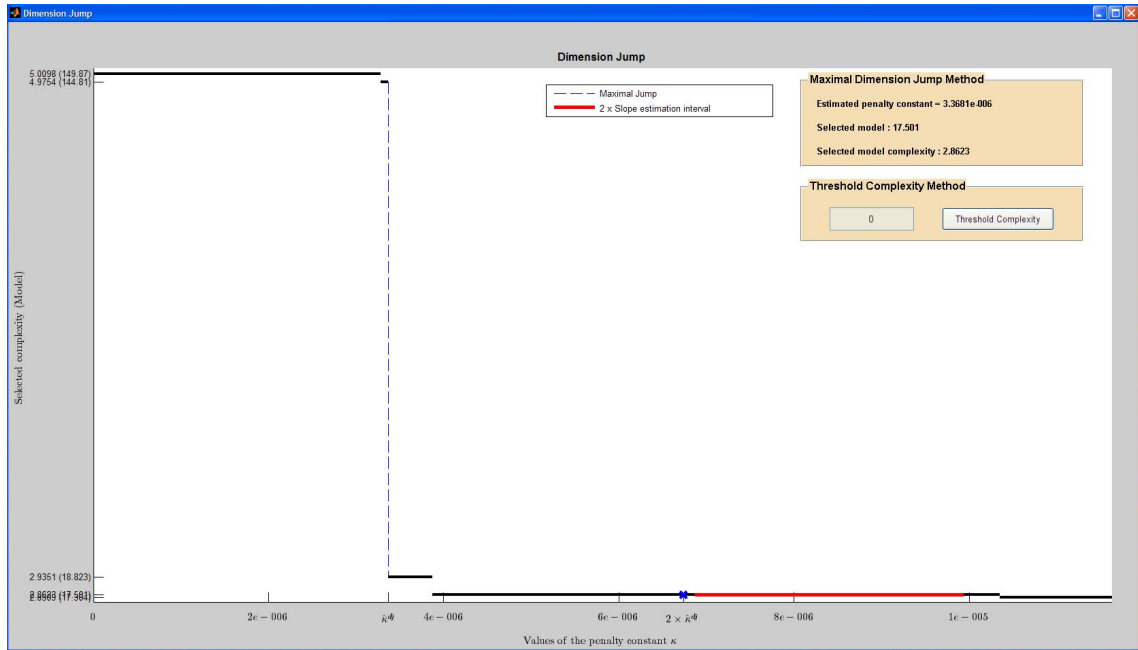
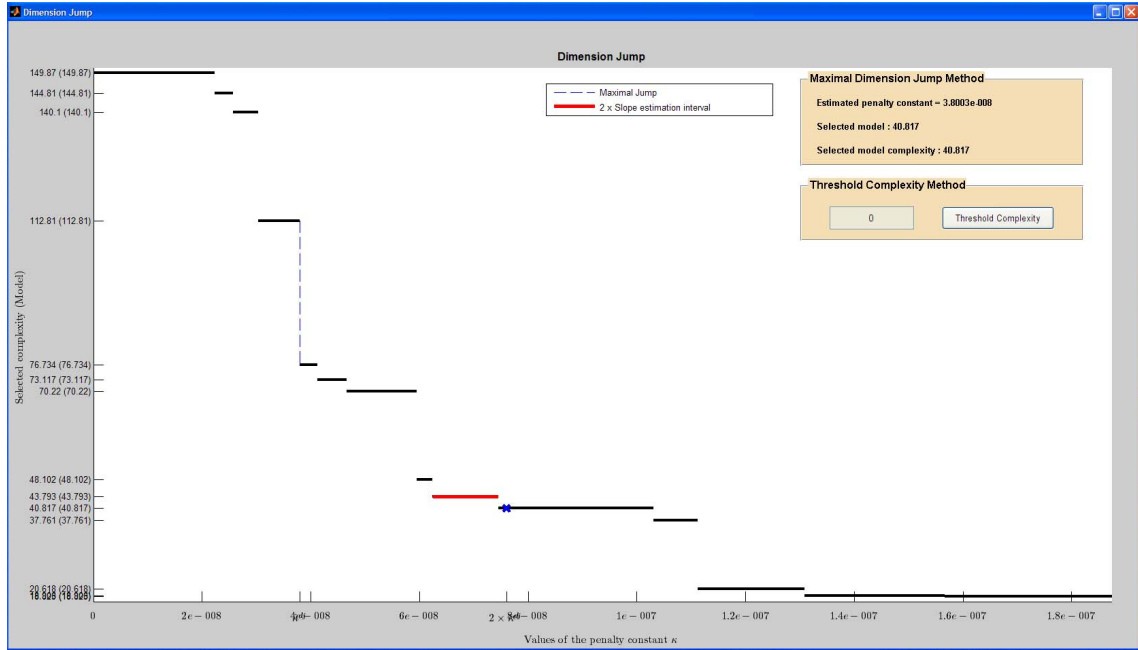


Figure 11: Dimension jump method for  $\text{pen}_{\text{shape}}^{(1)}(\cdot)$  (top) and for  $\text{pen}_{\text{shape}}^{(2)}(\cdot)$  (bottom). Many jumps can be observed with the first penalty shape whereas there is only one large single jump for the second penalty shape.

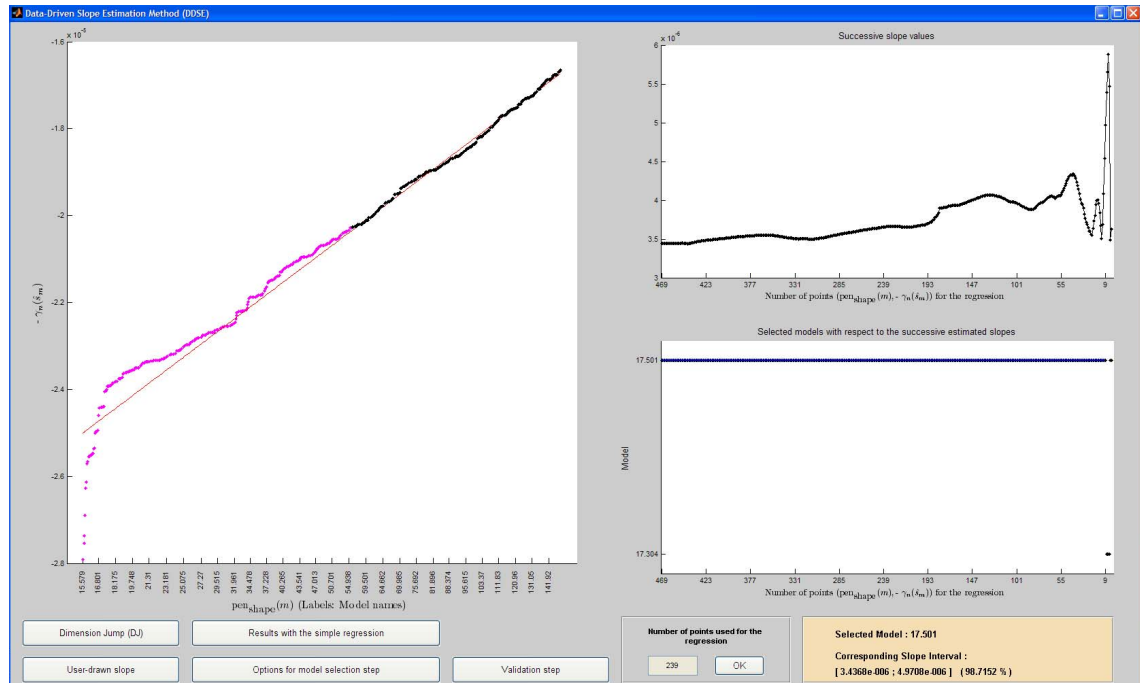
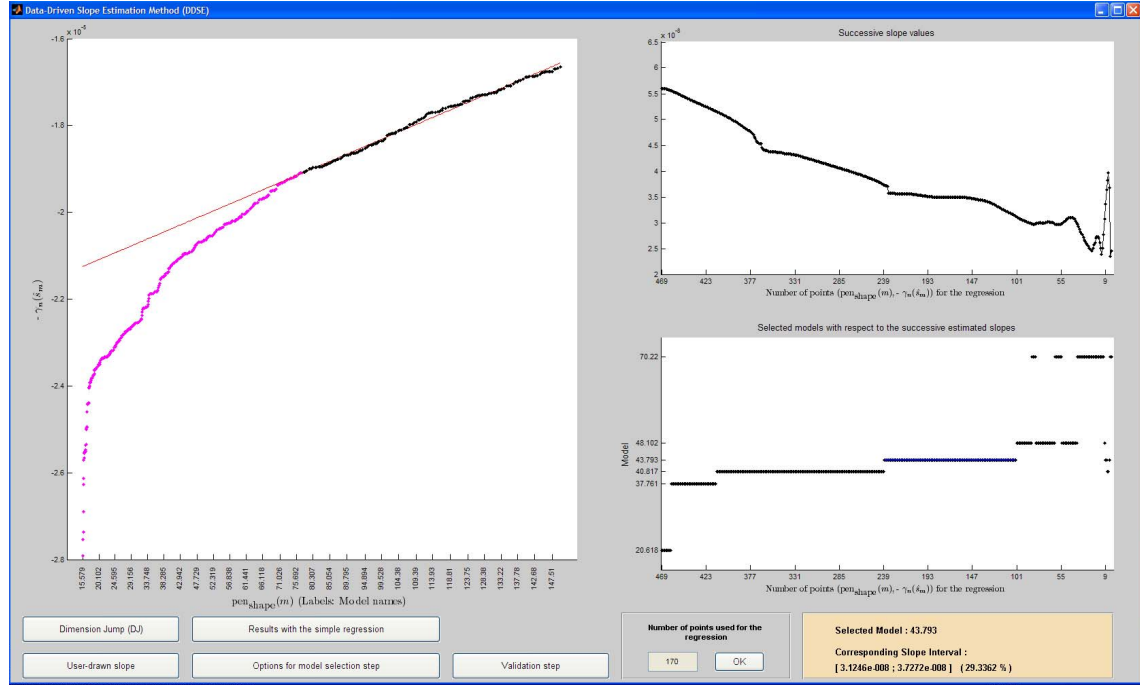


Figure 12: Data-driven slope estimation method for  $\text{pen}_{\text{shape}}^{(1)}$  (top) and for  $\text{pen}_{\text{shape}}^{(2)}$  (bottom). The second penalty shape leads to a better linear behavior than with the first one and there is no clear stability of the selected model with  $\text{pen}_{\text{shape}}^{(1)}$ .

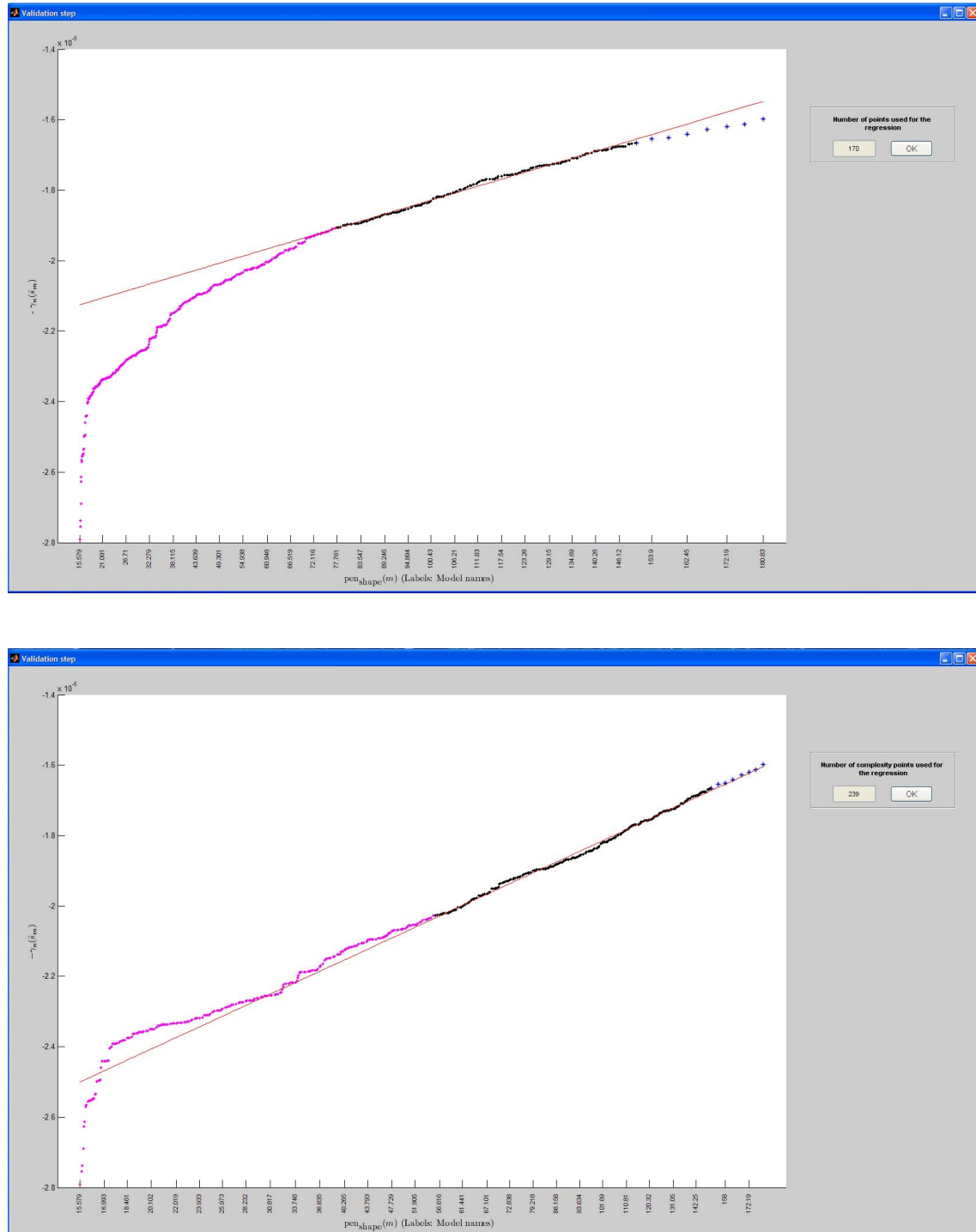


Figure 13: Validation for  $\text{pen}_{\text{shape}}^{(1)}(\cdot)$  (top) and for  $\text{pen}_{\text{shape}}^{(2)}(\cdot)$  (bottom): only the second penalty shape gives satisfying results.

But this comparison study has to be continued and deepened, both theoretically and practically.

The CAPUSHE interface makes the slope heuristics easy to apply for any statistician who would like to try it without having to care much about the practical difficulties it involves. Hopefully it shall contribute to a more widespread use of the slope heuristics. As it shall be more used, there will be an increasing quantity of available material to pursue the study and understanding of this approach. Moreover the package is a convenient tool to directly study some questions raised by the slope heuristics study. It may be useful for example in order to compare the two available strategies for the application of the dimension jump: the maximal dimension jump versus the threshold complexity.

The slope heuristics are being studied in new situations, which may uncover new difficulties and solutions. For example Arlot and Bach (2009) propose an oracle procedure to select among linear estimators, where the minimal penalty shape is different from the optimal penalty shape. By the way, this is an instance of a situation where the minimal penalty shape is not proportional to the model dimension. Selecting the estimator based on twice the minimal penalty leads to overpenalizing in this case. This suggests that future versions of the package may have to involve new functionalities: the current one does not enable to handle such a situation.

**Acknowledgements:** The authors thank S. Arlot (CNRS/ENS/INRIA UMR 8548) and G. Celeux (Inria Saclay - Île-de-France) for helpful discussions and valuable comments. The data used in Section 5.2 were obtained from E. Lebarbier (AgroParisTech). This work was supported by the SELECT Project (INRIA Saclay - Île-de-France).

**Availability:** The Matlab graphical interface CAPUSHE is available at <http://www.math.univ-toulouse.fr/~maugis/CAPUSHE>.

## References

- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proceedings, 2nd Internat. Symp. on Information Theory*, pages 267–281.
- Arlot, C. and Celisse, A. (2009). A survey of cross-validation procedures for model selection. Preprint, arXiv:0907.4728v1.
- Arlot, S. (2009). Model selection by resampling penalization. *Electronic Journal of Statistics*, 3:557–624 (electronic).
- Arlot, S. and Bach, F. (2009). Data-driven calibration of linear estimators with minimal penalties. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 46–54.
- Arlot, S. and Massart, P. (2009). Data-driven calibration of penalties for least-squares regression. *Journal of Machine Learning Research*, 10:245–279 (electronic).

- Baudry, J.-P. (2009). *Sélection de modèle pour la classification non supervisée. Choix du nombre de classes*. PhD thesis, University Paris XI. [http://www.math-info.univ-paris5.fr/~baudryjp/Docs/Baudry\\_Manuscrit\\_100126.pdf](http://www.math-info.univ-paris5.fr/~baudryjp/Docs/Baudry_Manuscrit_100126.pdf).
- Baudry, J.-P., Celeux, G., and Marin, J.-M. (2008). Selecting models focussing on the modeller's purpose. In *COMPSTAT 2008: Proceedings in Computational Statistics*, pages 337–348, Heidelberg. Physica-Verlag.
- Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006). Model-based cluster and discriminant analysis with the MIXMOD software. *Computational Statistics and Data Analysis*, 51(2):587–600.
- Birgé, L. and Massart, P. (2001). Gaussian model selection. *Journal of the European Mathematical Society*, 3(3):203–268.
- Birgé, L. and Massart, P. (2006). Minimal penalties for gaussian model selection. *Probability Theory and Related Fields*, 138(1-2):33–73.
- Burnham, K. and Anderson, D. (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, New York, 2nd edition.
- Caillerie, C. and Michel, B. (2009). Model selection for simplicial approximation. RR 6981, INRIA.
- Castellan, G. (2000). Sélection d'histogrammes à l'aide d'un critère de type Akaike. *C. R. Acad. Sci. Paris Sér. I Math.*, 330(8):729–732.
- Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, 31(4):377–403.
- Denis, M. and Molinari, N. (2009). Choix du nombre de noeuds en régression spline par l'heuristique des pentes. In *41èmes Journées de Statistique, SFdS, Bordeaux*, Bordeaux, France.
- Fraley, C. and Raftery, A. E. (2003). Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST. *Journal of Classification*, 20(2):263–286.
- Huber, P. J. (1981). *Robust Statistics*. Wiley.
- Keribin (2000). Consistent estimation of the order of mixture models. *Sankhya Series A*, 62(1):49–66.
- Lebarbier, E. (2005). Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal Processing*, 85(4):717–736.
- Lepez, V. (2002). *Some estimation problems related to oil reserves*. PhD thesis, University Paris XI.
- Lerasle, M. (2009a). Adaptive density estimation of stationary  $\beta$ -mixing and  $\tau$ -mixing processes. *Math. Methods Statist.*, 18(1):59–83.



- Lerasle, M. (2009b). Optimal model selection in density estimation. Preprint, arXiv : 0910.1654.
- Lerasle, M. (2009c). *Rééchantillonnage et sélection de modèles optimale pour l'estimation de la densité*. PhD thesis, University of Toulouse. <http://www-gmm.insa-toulouse.fr/~mlerasle/index2.html>.
- Mallows, C. L. (1973). Some comments on cp. *Technometrics*, 15(4):661–675.
- Massart, P. (2007). *Concentration Inequalities and Model Selection. École d'été de Probabilités de Saint-Flour 2003. Lecture Notes in Mathematics*. Springer.
- Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009). Variable selection for Clustering with Gaussian Mixture Models. *Biometrics*, 65:701–709.
- Maugis, C. and Michel, B. (2009). A non asymptotic penalized criterion for Gaussian mixture model selection. *ESAIM: P & S*. To appear.
- Maugis, C. and Michel, B. (2010). Data-driven penalty calibration: a case study for Gaussian mixture model selection. *ESAIM: P & S*. To appear.
- Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6:461–464.
- Verzelen, N. (2009). Data-driven neighborhood selection of a Gaussian field. RR 6798, INRIA.
- Villers, F. (2007). *Tests et sélection de modèles pour l'analyse de données protéomiques et transcriptomiques*. PhD thesis, University Paris XI. <http://www.proba.jussieu.fr/~villers/manuscript.pdf>.



---

Centre de recherche INRIA Saclay – Île-de-France  
Parc Orsay Université - ZAC des Vignes  
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex  
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier  
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq  
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex  
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex  
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex  
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399