Stat Comput (2012) 22:455–470
DOI 10.1007/s11222-011-9236-1

Slope heuristics: overview and implementation

Jean-Patrick Baudry · Cathy Maugis · Bertrand Michel

Received: 30 March 2010 / Accepted: 4 February 2011 / Published online: 20 April 2011

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 (Probab. Theory Relat. Fields 138:33–73, 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. Supplemental Materials containing further information and an additional application, the CAPUSHE package and the datasets presented in this paper, are available on the journal Web site.

Keywords Data-driven slope estimation · Dimension jump · Model selection · Penalization · Slope heuristics

1 Introduction

In this paper, we focus on the so-called *slope heuristics* proposed by Birgé and Massart (2001, 2006) for model selection via penalization. The purpose of this paper is twofold. First an overview of the theoretical and practical works on the slope heuristics is given. Second we propose a new practical solution for its implementation and compare it with the other available solutions.

To introduce the problematic of the slope heuristics, let us consider a familiar example in statistics: the Gaussian homoscedastic least squares regression with fixed design. Suppose that one observes $X_1, \dots, X_n$ with

$$\forall i \in \{1, \dots, n\}, \quad X_i = s(u_i) + \varepsilon_i,$$

where $s \in \mathbb{L}_2([0,1])$, $0 \le u_1 \le \cdots \le u_n \le 1$ is a fixed design and the regression errors ε_i are i.i.d. with centered Gaussian distribution of variance σ^2. The regression function s may be estimated by considering piecewise constant functions on $[0,1]$. Each partition of $[0,1]$ provides a set of such histogram functions. In particular, let S_m be the

This work was supported by the SELECT Project (INRIA Saclay—Île-de-France).

Electronic supplementary material The online version of this article (doi:10.1007/s11222-011-9236-1) contains supplementary material, which is available to authorized users.

J.-P. Baudry (✉)
Laboratoire de Mathématiques d'Orsay, Université Paris-Sud 11 and INRIA SELECT, Bâtiment 425, 91405 Orsay Cedex, France
e-mail: Jean-Patrick.Baudry@upmc.fr

J.-P. Baudry
Laboratoire MAP5, Université Paris Descartes and CNRS, Paris, France

C. Maugis
INSA de Toulouse, Département de Génie Mathématique, Institut de Mathématiques de Toulouse, 135, avenue de Rangueil, 31077 Toulouse Cedex 4, France

B. Michel
Laboratoire de Statistique Théorique et Appliquée, Université Pierre et Marie Curie, place Jussieu, 75005 Paris, France

set of histograms defined on the regular partition of $[0,1]$ into m intervals. Any such function subspace of $\mathbb{L}_2([0,1])$ is called a model. A natural estimator $\hat{s}_m$ of s is obtained by minimizing the least squares empirical contrast $\gamma_n : t \in \mathbb{L}_2([0,1]) \mapsto \frac{1}{n}\sum_{i=1}^{n}(X_i - t(u_i))^2$ over S_m. Let $\mathcal{M}$ be a finite set of integers m. The aim is to minimize the risk over the collection $(\hat{s}_m)_{m\in\mathcal{M}}$ based on the data, where the risk of $\hat{s}_m$ is $\mathcal{R}(\hat{s}_m) = \mathbb{E}[\|s - \hat{s}_m\|^2]$. Due to overfitting this choice cannot be done by only minimizing $\gamma_n(\hat{s}_m)$ since an optimistic bias for the evaluation of the risk would be introduced. Thus a model is selected by minimizing a penalized criterion $m \in \mathcal{M} \mapsto \gamma_n(\hat{s}_m) + \mathrm{pen}(m)$. The more the model involves parameters, the larger the bias for the risk estimation. A popular solution is then to define the penalty function proportional to the model dimension: $\mathrm{pen}(m) = \kappa m$. Moreover it is natural to allow the model collection to increase with n: this is a typical non asymptotic situation (see the introduction of Massart 2007). In this context the multiplicative constant κ is a priori unknown and has to be calibrated.

This problem actually occurs in many model selection frameworks: since many non asymptotic theoretical results provide penalties known up to a multiplicative constant. To answer this question, Birgé and Massart (2001, 2006) have proposed the slope heuristics method.[1] From a theoretical point of view, the principle of the slope heuristics is introduced and proved for the first time in 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 dimension and the risk of models selected with lighter penalties become very large. Moreover, they prove that *considering a penalty equal to twice this minimal penalty allows to select a model close to the oracle model in terms of risk* (see Sect. 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 and to deduce an optimal penalty from this estimate. This enables to overcome the difficulty that some constants needed to design the penalty are unknown. 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 on 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 main issue is to determine the minimal penalty. To this aim, it is assumed in this paper that a complexity measure 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. For instance in the regression example detailed before, the complexity is exactly the number of intervals in the partition. The most studied and applied approach to determine the minimal penalty is the so-called *dimension jump*.[2] It consists of considering the complexity of the selected model as a function of the multiplicative constant κ in the penalty. Then increasing the constant value from 0, a non increasing 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 after which the selected complexity is smaller than a chosen threshold. The choice of the threshold (or of the

[1] The slope heuristics takes its name from a slope estimation. In our introduction regression example, the method requires the estimation of the slope of $m \mapsto \gamma_n(\hat{s}_m)$ (see below).

[2] This method is based on a "complexity jump" but in the first studied frameworks, the complexity was actually the model dimension.

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. The 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 practical difficulties. It notably has the advantage to validate whether 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 (CAlibrating Penalty Using Slope HEuristics). Hopefully it will contribute to a better understanding and a wider use of the slope heuristics.

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

2 Contrast minimization and 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 $\gamma : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$ 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:

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

Let us define the empirical contrast:

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

Let S be a model, namely a subset of $\mathcal{S}$. A minimizer of the empirical contrast over the model S is then considered and denoted as $\hat{s}$. Indeed it is expected that $\hat{s}$ is a sensible estimator of s since, under reasonable conditions, $\gamma_n(t)$ converges to $\mathbb{E}[\gamma(t, X)]$. The quality of such an estimator can be measured by its *risk* $\mathcal{R}(\hat{s}) = \mathbb{E}_{\mathbf{X}}[l(s, \hat{s})]$.

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 μ. Let t denote another density with respect to the same measure. Then the contrast $\gamma(t, x) = -\ln[t(x)]$ is the maximum likelihood contrast. The corresponding loss function is the Kullback-Leibler divergence defined by $\mathrm{KL}(s,t) = \int s \ln(\frac{s}{t}) d\mu$. 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 estimators 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 $S_{\hat{m}}$ 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

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

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 $S_{\hat{m}}$ is *optimal*. More precisely, it fulfills an oracle inequality:

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

with A_n as close to 1 as possible and η_n a remainder term small 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:

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

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 approximations 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 pen : $\mathcal{M} \longrightarrow \mathbb{R}^+$ and of selecting $\hat{m}$ minimizing the associated penalized criterion

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

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 Mallow's C_p (Mallows 1973) and GCV (Craven and Wahba 1978). Nevertheless, Mallow's C_p depends on the noise level σ^2 of the true regression model which is unknown (if it does exist) and σ^2 is thus difficult to estimate. Similarly, GCV depends on a tuning parameter which best value is actually σ^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 κ. In this framework, the penalty shape is then denoted as $\mathrm{pen}_{\mathrm{shape}}(\cdot)$ and an unknown constant κ_{opt} exists such that

$$\mathrm{pen}_{\mathrm{opt}} : m \in \mathcal{M} \mapsto \kappa_{\mathrm{opt}}\, \mathrm{pen}_{\mathrm{shape}}(m) \tag{3}$$

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

Deterministic penalty shapes Specific deterministic functions $m \mapsto \mathrm{pen}_{\mathrm{shape}}(m)$ 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 κ_{opt} such that $\mathrm{pen}_{\mathrm{opt}}(\cdot) = \kappa_{\mathrm{opt}}\, \mathrm{pen}_{\mathrm{shape}}(\cdot)$ follows an oracle inequality. The value of κ_{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.

According to (1) and (2), to select the oracle model, the penalty must be chosen as $m \in \mathcal{M} \mapsto l(s, \hat{s}_m) - \gamma_n(\hat{s}_m)$. Of course, it is not possible in practice since s is unknown but it may help for defining a penalty which fulfills an oracle inequality. Such a penalty can be decomposed into

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

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

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

where $v_m = \mathbb{E}_X[\gamma(\hat{s}_m, X) - \gamma(s_m, X)]$ is an "estimation error" term, $\hat{v}_m = \gamma_n(s_m) - \gamma_n(\hat{s}_m)$ is an empirical "estimation error" term and

$$\Delta_n(s_m) = \left\{\mathbb{E}_X\left[\gamma(s_m, X)\right] - \mathbb{E}_X\left[\gamma(s, X)\right]\right\} \\ - \left\{\gamma_n(s_m) - \gamma_n(s)\right\}$$

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 $m \in \mathcal{M}$,

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

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

$$\forall m \in \mathcal{M}, \quad \gamma_n(\hat{s}_{\hat{m}}) + \mathrm{pen}(\hat{m}) \le \gamma_n(\hat{s}_m) + \mathrm{pen}(m)$$

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

$$l(s, \hat{s}_{\hat{m}}) + \left[\mathrm{pen}(\hat{m}) - \mathrm{pen}^*(\hat{m})\right] \\ \le \inf_{m \in \mathcal{M}} \left\{l(s, \hat{s}_m) + \left[\mathrm{pen}(m) - \mathrm{pen}^*(m)\right]\right\}.$$

Thus it is relevant to look for a penalty close to the ideal penalty for any m 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 $\mathrm{pen}(m) = \hat{v}_m$, the penalized criterion is $\mathrm{crit}(m) = \gamma_n(\hat{s}_m) + \hat{v}_m = \gamma_n(s_m)$, 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 $\mathrm{pen}(m) = \kappa \hat{v}_m$, the criterion can be written as $\mathrm{crit}(m) = (1 - \kappa)\gamma_n(\hat{s}_m) + \kappa\gamma_n(s_m)$. Therefore two 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 almost have the same bias, and thus they are ruled out.

This suggests that $\mathrm{pen}_{\min}(m) = \hat{v}_m$ 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. 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:

$$\mathrm{pen}^*(m) \approx v_m + \hat{v}_m \approx 2\hat{v}_m.$$

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

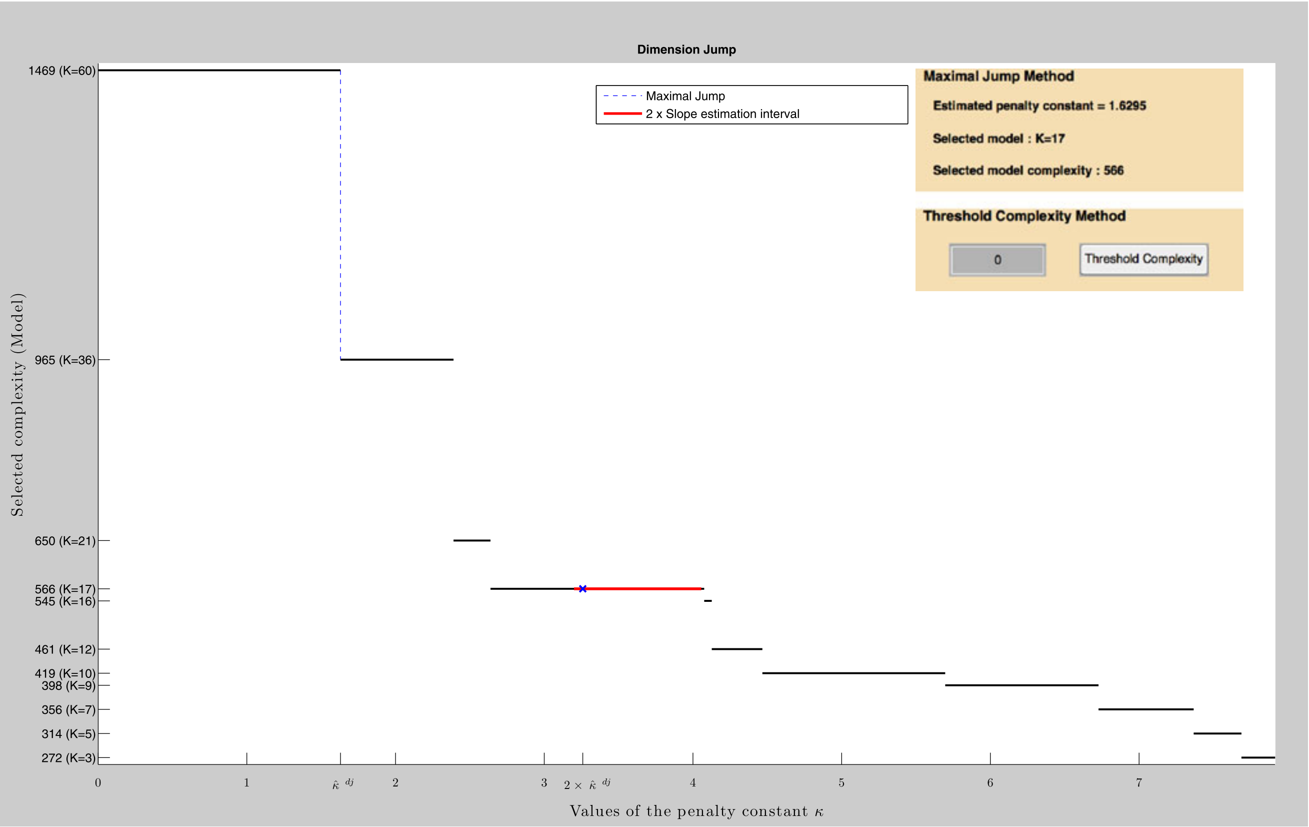
In practice, this heuristics is useful when an optimal penalty $\mathrm{pen}_{\mathrm{opt}}(\cdot) = \kappa_{\mathrm{opt}}\,\mathrm{pen}_{\mathrm{shape}}(\cdot)$ is known up to a multiplicative factor. Note that the slope heuristics is derived by considering the ideal penalty, whereas it is applied to a particular penalty shape chosen by the user. Thus, it is not necessarily guaranteed that the ideal penalty itself is of the shape $\kappa^*\,\mathrm{pen}_{\mathrm{shape}}(\cdot)$. 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 $\frac{\kappa_{\mathrm{opt}}}{2}\,\mathrm{pen}_{\mathrm{shape}}(m)$ is a good estimate of $\hat{v}_m$ and provides a minimal penalty.

For the two application methods of the slope heuristics presented in Sects. 3 and 4, it is assumed that a complexity measure C_m of the models is given. As it has already been explained before, this complexity measure is typically the model dimension or the number of free parameters in parametric frameworks, as in the regression example in the Introduction Section. Generally speaking, the penalty shape can be written as a function of C_m. When its definition is not 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 in the Supplemental Materials gives the expression of the complexities C_m and the penalty shapes $\mathrm{pen}_{\mathrm{shape}}$ for a large list of model selection works.

For the two methods to apply the slope heuristics presented in Sects. 3 and 4, it is required that:

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



Maximal Jump Method

Estimated penalty constant = 1.6295

Selected model : K=17

Selected model complexity : 566

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

sart (2006). The second one consists of directly estimating the "slope" κ_{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 κ_{opt} in (3). Let $m(\kappa)$ be the model selected by the penalized criterion $m \mapsto \gamma_n(\hat{s}_m) + \kappa\,\text{pen}_{\text{shape}}(m)$. 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 Fig. 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 fact, the choice of complexity measure is crucial for this method (see Sect. 3 in the Supplemental Materials for an application coming from Caillerie and Michel (2009) which illustrates this). If several complexity measures seem relevant for the user, they can all be tested to find the one that shows the clearest jump.

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, for all $\kappa > 0$,
$$m(\kappa) \in \operatorname*{argmin}_{m \in \mathcal{M}} \left\{ \gamma_n(\hat{s}_m) + \kappa\ \text{pen}_{\text{shape}}(m) \right\};$$
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 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 $(i_{\max} - 1) \in \{0, \dots, \text{card}(\mathcal{M}) - 1\}$, and the associated selected model sequence $(m_i)_{0\leq i<i_{\max}}$ where $m_i = m(\kappa)$ for all κ in $[\kappa_i, \kappa_{i+1})$ and for all $i < i_{\max}$.

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

Maximal jump
This first method is the most popular. It consists of choosing the constant $\hat{\kappa}^{\text{dj}}$ corresponding to the greatest jump of complexity: $\hat{\kappa}^{\text{dj}} = \kappa_{i_{\text{dj}}}$, with $i_{\text{dj}} \in \text{argmax}_{0\leq i<i_{\max}-1}\{C_{m_i} - C_{m_{i+1}}\}$. If several values of κ reach the maximum value, Lebarbier (2005) suggests to choose the largest κ 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_{thresh} such that complexities smaller than C_{thresh} are reasonable but larger ones are not. Then the chosen constant $\hat{\kappa}^{\text{thresh}}$ is the smallest value of κ for which the corresponding penalty selects a complexity smaller than C_{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_{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 κ_{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 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 while estimating the slope. It is also presented and studied in Baudry et al. (2008) and Maugis and Michel (2010). In this section, we propose 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

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

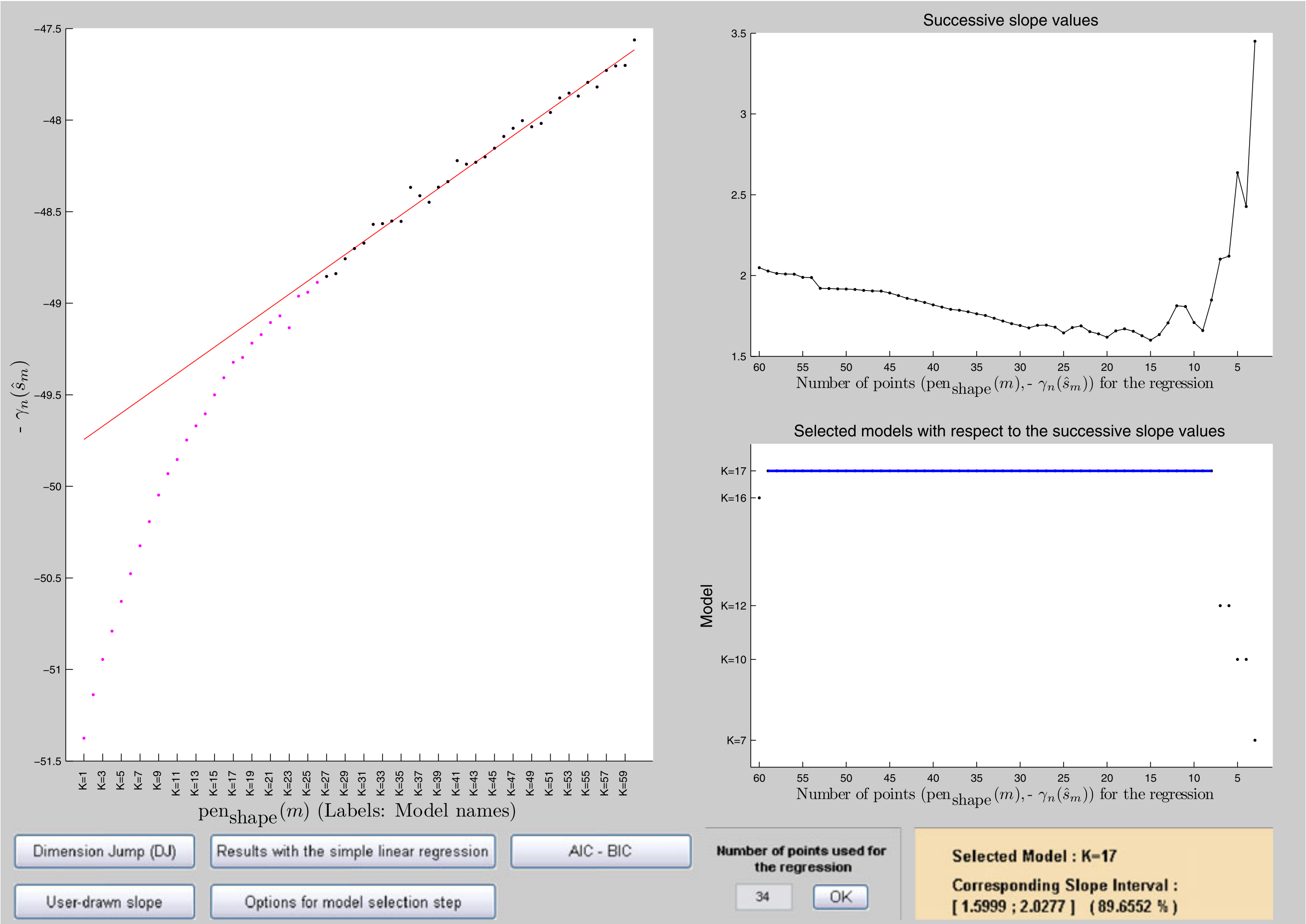
The empirical bias term $\gamma_n(s_m) - \gamma_n(s)$ gets stable for the most complex models for which the approximation of the target s cannot be appreciably improved. Hence the behavior of $\kappa_{\text{opt}}\,\text{pen}_{\text{shape}}(m)$ is known through $-2\gamma_n(\hat{s}_m)$ for models of large 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 Fig. 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 $2\hat{\kappa}\,\text{pen}_{\text{shape}}(\cdot)$.

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 Sect. 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:

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 any $p \in \mathcal{P}$, 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.


-47.5
-48
-48.5
-49
-49.5
-50
-50.5
-51
-51.5
$-\gamma_n(\hat{s}_m)$
K=1
K=3
K=5
K=7
K=9
K=11
K=13
K=15
K=17
K=19
K=21
K=23
K=25
K=27
K=29
K=31
K=33
K=35
K=37
K=39
K=41
K=43
K=45
K=47
K=49
K=51
K=53
K=55
K=57
K=59
$\text{pen}_{\text{shape}}(m)$ (Labels: Model names)
Successive slope values
3.5
3
2.5
2
1.5
60
55
50
45
40
35
30
25
20
15
10
5
Number of points $(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m))$ for the regression
Selected models with respect to the successive slope values
Model
K=17
K=16
K=12
K=10
K=7
60
55
50
45
40
35
30
25
20
15
10
5
Number of points $(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m))$ for the regression
Dimension Jump (DJ)
Results with the simple linear regression
AIC - BIC
User-drawn slope
Options for model selection step
Number of points used for the regression
34
OK

Selected Model : K=17

Corresponding Slope Interval :
[1.5999 ; 2.0277] (89.6552 %)

Fig. 2 An example of the CAPUSHE results. *The left graph* represents $-\gamma_n(\hat{s}_m)$ with respect to $\text{pen}_{\text{shape}}(m)$ to check the linear behavior assumption. *The top-right* (resp. *bottom-right*) *graph* gives the estimated slope (resp. the selected model) as a function of the number of couples $(\text{pen}_{\text{shape}}(m), -\gamma_n(\hat{s}_m))$ used for the linear regression. The last plateau for which the length $N_{\hat{\imath}}$ is greater than $pct \sum_{l=1}^{I} N_l$ is detected and the corresponding model $\hat{m}(p_{\hat{\imath}})$ is selected. The "Corresponding slope interval" given *bottom right* is the interval $[p_{\hat{\imath}}, p_{\hat{\imath}+1})$ leading to select $\hat{m}(p_{\hat{\imath}})$

Step 3 For any $p \in \mathcal{P}$, 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 $(p_i)_{1 \le i \le I+1}$ such that

$$\forall 1 \le i \le I-1,\ \forall p \in \mathcal{P},$$
$$\begin{cases} \hat{m}(p) = \hat{m}(p_i) & \iff \quad p \in [p_i, p_{i+1}), \\ \hat{m}(p) = \hat{m}(p_I) & \iff \quad p \in [p_I, p_{I+1}]. \end{cases}$$

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

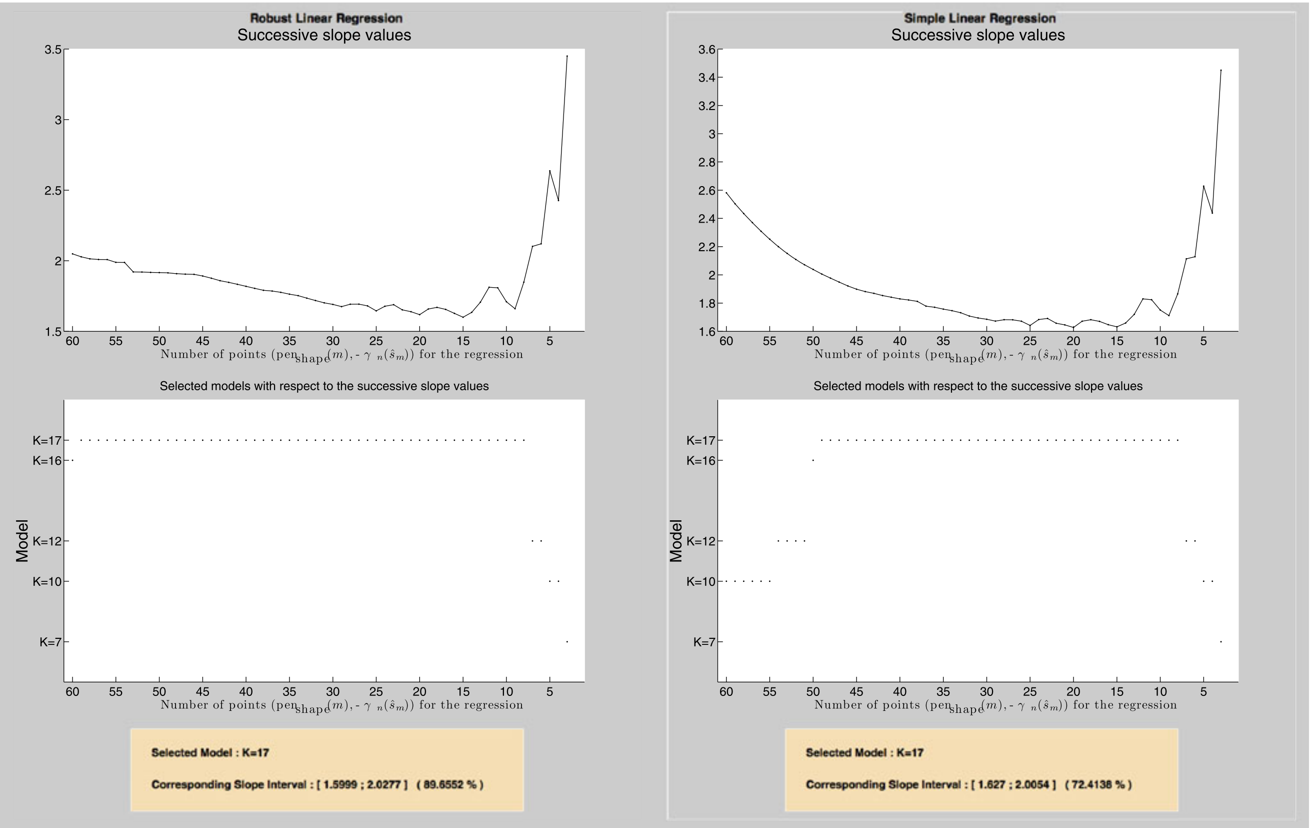
$$\forall 1 \le i \le I-1,$$
$$N_i = \text{card}\{[p_i, p_{i+1}) \cap \mathcal{P}\} \quad \text{and}$$
$$N_I = \text{card}\{[p_I, p_{I+1}] \cap \mathcal{P}\}.$$

Step 4 The model $\hat{m}(p_{\hat{\imath}})$ such that

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

is selected (see hereafter for the choice of the *pct* value). We also return the interval of slope values $[p_{\hat{\imath}}, p_{\hat{\imath}+1}[$ and the proportion $N_{\hat{\imath}} / \sum_{l=1}^{I} N_l$. Graphically, this corresponds to selecting the "most to the right" plateau whose length is greater than the threshold (see the bottom-right graph in Fig. 2).

This algorithm requires to tune the parameter *pct* at Step 4 in order to determine which plateau corresponds to


Robust Linear Regression
Successive slope values
3.5
3
2.5
2
1.5
60 55 50 45 40 35 30 25 20 15 10 5
Number of points (pen_shape(m), - γ_n(ŝ_m)) for the regression
Selected models with respect to the successive slope values
Model
K=17
K=16
K=12
K=10
K=7
60 55 50 45 40 35 30 25 20 15 10 5
Number of points (pen_shape(m), - γ_n(ŝ_m)) for the regression
Simple Linear Regression
Successive slope values
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
60 55 50 45 40 35 30 25 20 15 10 5
Number of points (pen_shape(m), - γ_n(ŝ_m)) for the regression
Selected models with respect to the successive slope values
Model
K=17
K=16
K=12
K=10
K=7
60 55 50 45 40 35 30 25 20 15 10 5
Number of points (pen_shape(m), - γ_n(ŝ_m)) for the regression

Selected Model : K=17

Corresponding Slope Interval : [1.5999 ; 2.0277] (89.6552 %)

Selected Model : K=17

Corresponding Slope Interval : [1.627 ; 2.0054] (72.4138 %)

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

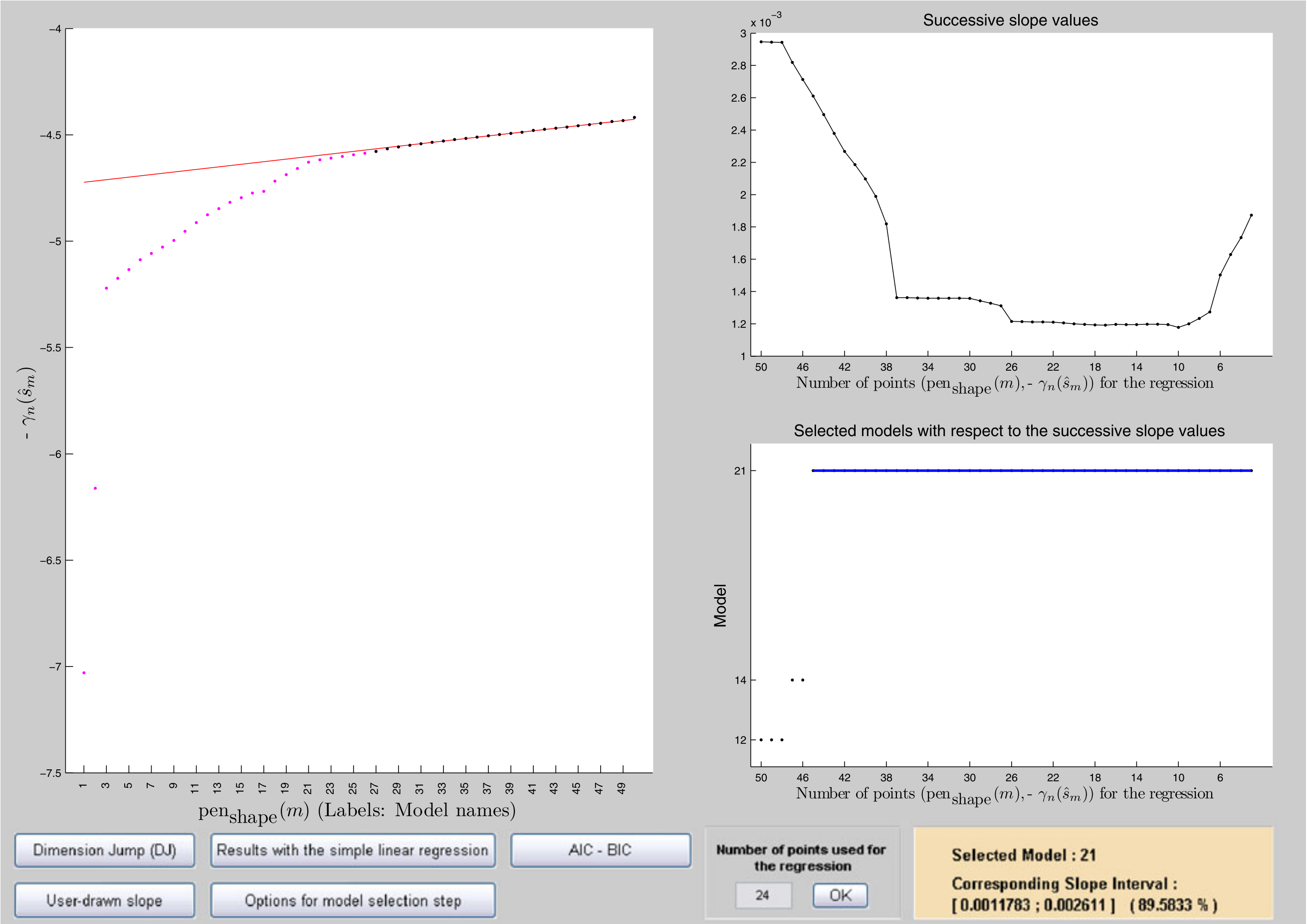
a stabilization of the model selection. By default, *pct* is set to 15% in CAPUSHE. This choice may be reconsidered according to the application at hand, and particularly to the size of $\mathcal{M}$ and to whether it is expected that many too complex models have been involved in the study. However, the *pct* deeply impacts the model selection only in situations for which no linear behavior can be observed (see the Transcriptome dataset example in Sect. 5.1.2). On the contrary, it is expected that the method is not much sensitive to *pct* in favorable situations (see the "Bubbles" experiment in Sect. 5.1.1). Remark that whatever the choice at this step, the reported actual proportion $N_{\hat{\imath}}/\sum_{l=1}^{I-1} N_l$ measures the stability of the method: the higher this value the more confidently the method can be applied. Moreover, the graph of $p \in \mathcal{P} \mapsto \hat{m}(p)$ (bottom-right in Fig. 2) enables to evaluate whether the choice of the good plateau is obvious or not, namely there is a plateau clearly larger than the others to the right of this graph.

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

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 complex enough 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 Sect. 5.1.2.

5 Applications

This section illustrates 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.


-4
-4.5
-5
-5.5
-6
-6.5
-7
-7.5
$-\gamma_n(\hat{s}_m)$
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
$\mathrm{pen}_{\mathrm{shape}}(m)$ (Labels: Model names)
Successive slope values
x 10^-3
3 2.8 2.6 2.4 2.2 2 1.8 1.6 1.4 1.2 1
50 46 42 38 34 30 26 22 18 14 10 6
Number of points ($\mathrm{pen}_{\mathrm{shape}}(m)$,- $\gamma_n(\hat{s}_m)$) for the regression
Selected models with respect to the successive slope values
Model
21
14
12
50 46 42 38 34 30 26 22 18 14 10 6
Number of points ($\mathrm{pen}_{\mathrm{shape}}(m)$,- $\gamma_n(\hat{s}_m)$) for the regression
Dimension Jump (DJ)
Results with the simple linear regression
AIC - BIC
User-drawn slope
Options for model selection step
Number of points used for the regression
24
OK

Selected Model : 21

Corresponding Slope Interval :
[0.0011783 ; 0.002611] (89.5833 %)

Fig. 4 Graphical output obtained by the data-driven slope estimation method for the Bubbles experiment with $M_{\max} = 50$

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:

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

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 Σ 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 $\mathrm{pen}_{\mathrm{shape}}(m) = D_m$, 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 $(S_m)_{1 \le m \le M_{\max}}$.

5.1.1 Bubbles experiment

This simulated dataset (plotted in Fig. 1 of the Online Resource file *Supp.pdf*) 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:

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

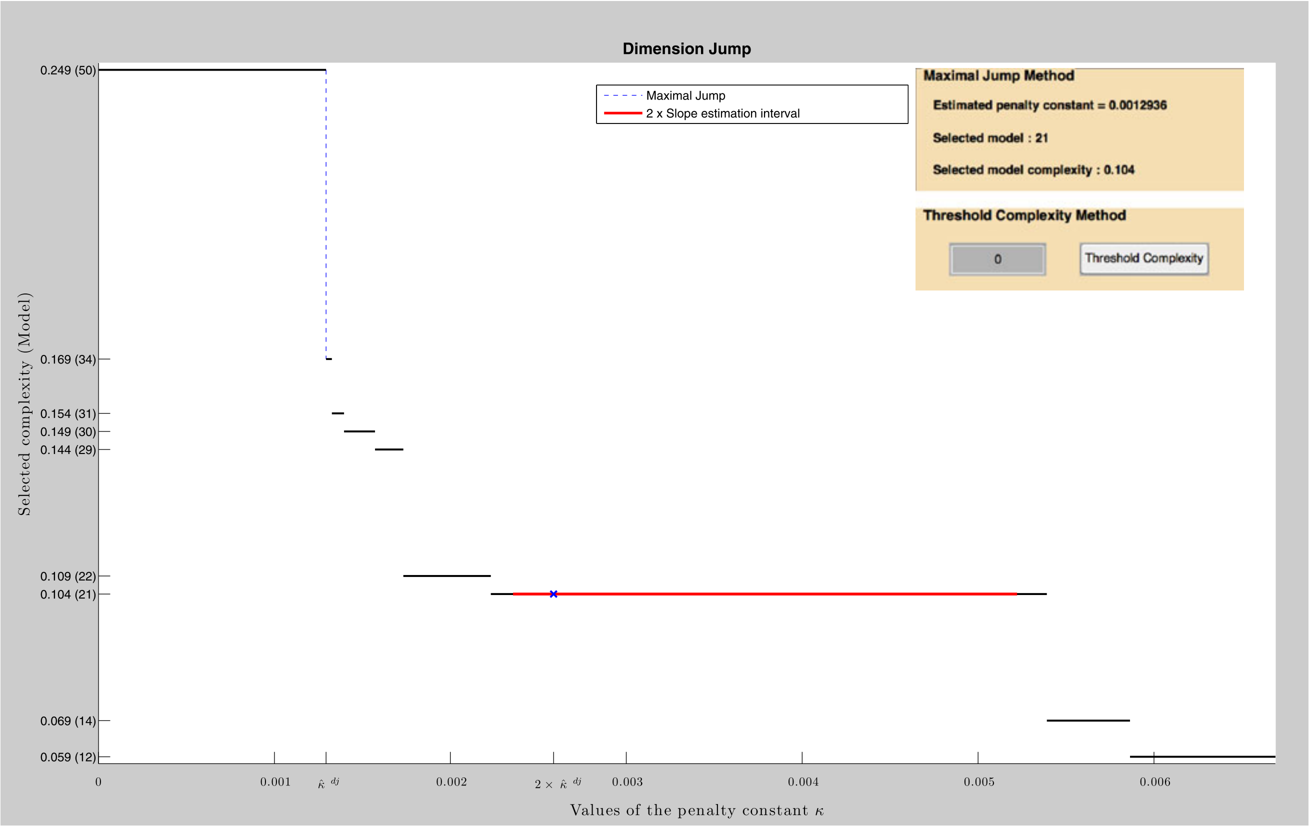


Fig. 5 Graphical output obtained by the dimension jump method for the Bubbles experiment with $M_{\max} = 50$

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 , Chap. 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 $\lambda_k \in \mathbb{R}_+^\star$.

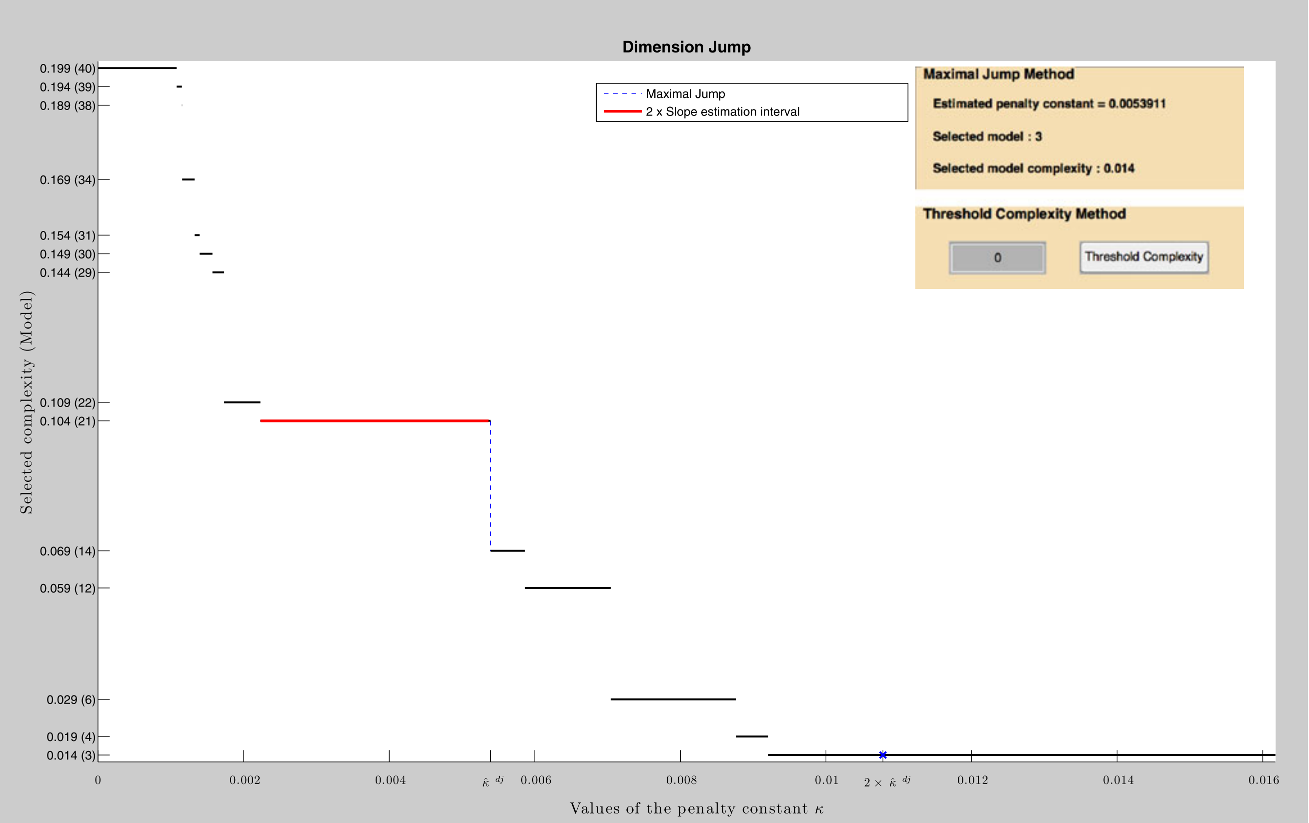
The outputs of CAPUSHE are explained with this simulated example. Figure 4 gives the graphical outputs obtained by the data-driven slope estimation method for the model collection with $M_{\max} = 50$. In this example, the linear behavior for the most complex models is clearly observed. Using the robust regression, the true Gaussian mixture with 21 components is selected in 89.6% of the successive slope estimations. The choice of a multiplicative constant 2κ in the penalized criterion with $\kappa \in [1.1783 \times 10^{-3}; 2.611 \times 10^{-3}]$ leads to select $\hat{m} = 21$. There is no ambiguity for the result with the dimension jump since the maximal complexity jump is really clear (see Fig. 5).

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, different values of *pct* have been tried: 5%, 10%, 15% and 20%. As expected in this example where the slope heuristics can obviously be applied, all values of *pct* yield similar results (with corresponding risks ratios ranging from 1.02 to 1.06). The results for the default value 15% are reported. Table 1 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 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 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 10% 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

Table 1 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

Selected number of components $\hat{m}$	$M_{\max}$	3	4	15–18	19	20	21	22	23	24	25	26	≥35	Risk ratio
Oracle	40, 50					1	**76**	15	3	3	2			1
AIC	40, 50												**100**	2.59
BIC	40, 50			3	6	23	**57**	9	1	1				1.17
DDSE	50				3	7	**60**	18	7	3	2			1.06
DDSE	40			1	3	7	**61**	18	4	4	2			1.08
DJ	50	10	2		3	6	**54**	19	1	3	2			1.94
DJ	40	40	2		2	3	**43**	7	2	1				4.15



Maximal Jump Method

Estimated penalty constant = 0.0053911

Selected model : 3

Selected model complexity : 0.014

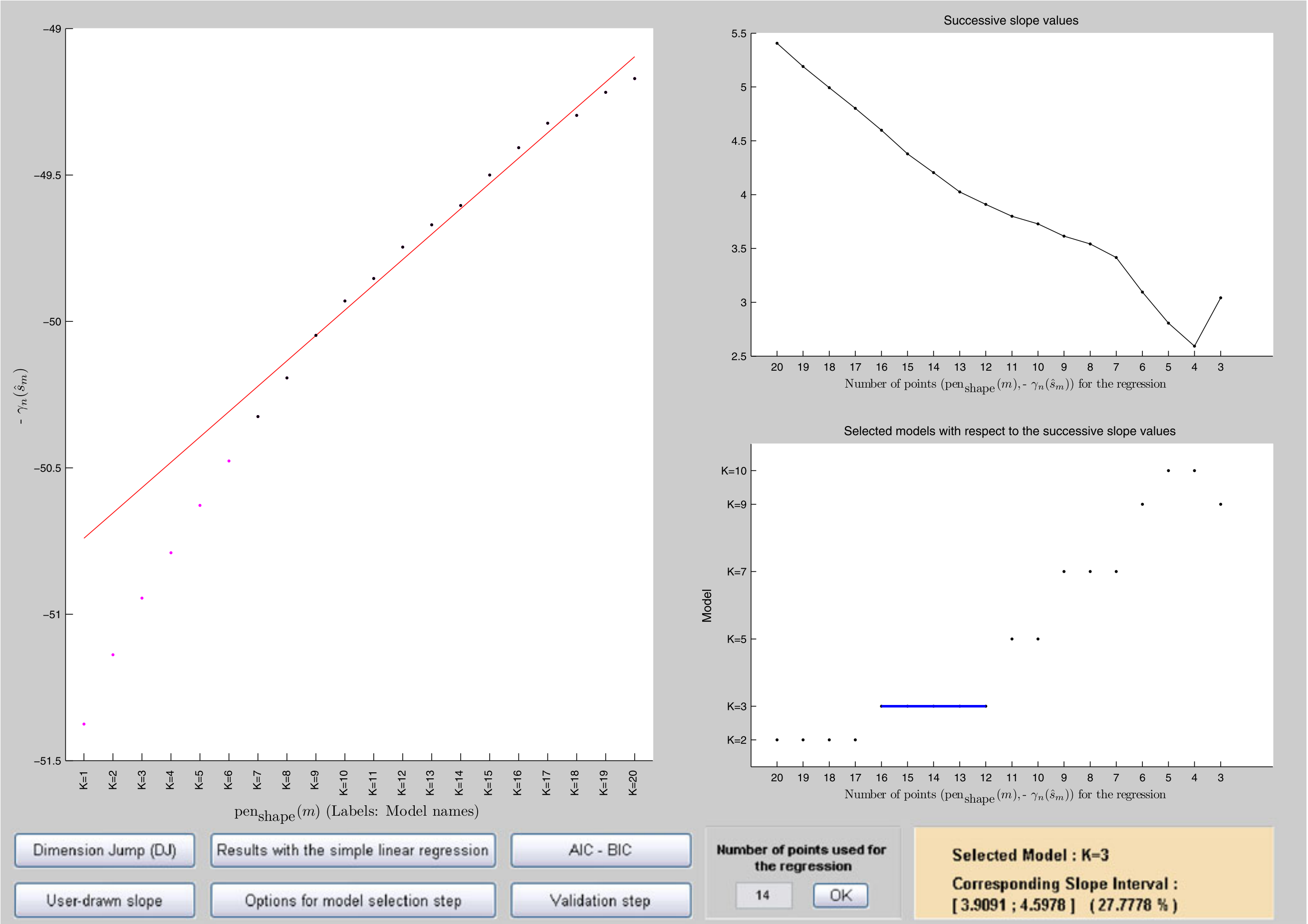
Fig. 6 Graphical output obtained by the dimension jump method for the Bubbles experiment with $M_{\max} = 40$

jump method has a larger risk because it sometimes selects small models.

When $M_{\max} = 40$, the results illustrate a difficulty which can be encountered while applying the dimension jump. This approach selects 40 times the model $\hat{m} = 3$ in the simulation study, which is a poor result. The reason of this difficulty is illustrated in Fig. 6: 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*


-49
-49.5
-50
-50.5
-51
-51.5
$-\gamma_n(\hat{s}_m)$
K=1 K=2 K=3 K=4 K=5 K=6 K=7 K=8 K=9 K=10 K=11 K=12 K=13 K=14 K=15 K=16 K=17 K=18 K=19 K=20
$\mathrm{pen}_{\mathrm{shape}}(m)$ (Labels: Model names)
Successive slope values
5.5
5
4.5
4
3.5
3
2.5
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3
Number of points $(\mathrm{pen}_{\mathrm{shape}}(m), -\gamma_n(\hat{s}_m))$ for the regression
Selected models with respect to the successive slope values
Model
K=10
K=9
K=7
K=5
K=3
K=2
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3
Number of points $(\mathrm{pen}_{\mathrm{shape}}(m), -\gamma_n(\hat{s}_m))$ for the regression
Dimension Jump (DJ)
Results with the simple linear regression
AIC - BIC
User-drawn slope
Options for model selection step
Validation step
OK

Number of points used for the regression
14

Selected Model : K=3
Corresponding Slope Interval :
[3.9091 ; 4.5978]   (27.7778 %)

Fig. 7 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

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 Fig. 7). 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 whether the considered model collection is large enough for applying the data-driven slope estimation method. The slope, estimated on the subset of couples $\{(\mathrm{pen}_{\mathrm{shape}}(m), -\gamma_n(\hat{s}_m));\ m \in \{1, \dots, M_{\max}\}\}$, is plotted and the user can graphically test whether other such couples for more 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 Fig. 7. Those 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 Fig. 2.

Remark that with $M_{\max} = 20$ the model selected through the data-driven slope estimation varies considerably with the choice of *pct*. This is apparent from the top of Fig. 7 (see bottom-right graph). It is obvious from Fig. 2 that the results are much more stable and reliable with $M_{\max} = 60$: a large spectrum of *pct* leads to selecting the same model, at least for values of *pct* larger than $2/60$.

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

$$X_i = s(u_i) + \varepsilon_i \tag{5}$$

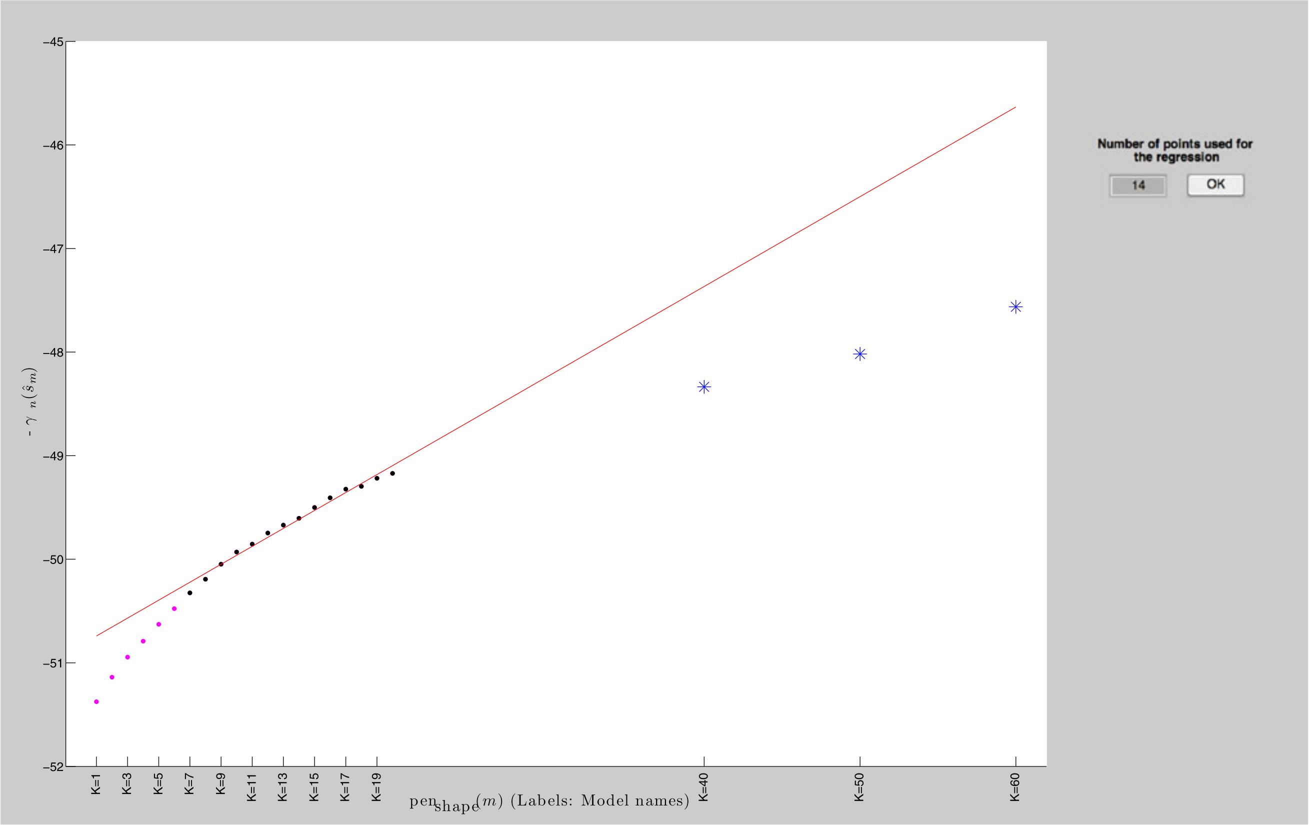


Fig. 8 Data-driven slope estimation method (*top*) and dimension jump (*bottom*) for the change-point detection problem. The two methods select the same model $|\hat{m}| = 6$

where the X_i's are observed at regular points $u_i = \frac{i}{n}$, $i = 1, \ldots, n$. The errors ε_i are assumed to be i.i.d. centered random variables with variance σ^2. Let $\mathcal{M}$ be the set of all the partitions of the grid $\{u_1, \ldots, u_n\}$. 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: $\gamma(t, (x, u)) = (x - t(u))^2$ over S_m. The aim is to determine the best estimator for the ℓ^2 risk. Lebarbier (2005) shows that a convenient penalty shape for this problem is

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

Case (b) of the simulations proposed in Lebarbier (2005) is considered (see Sect. 4.1.2 in Lebarbier 2005, for more details). A sample of 300 observations is simulated according to (5), s being a piecewise constant function with six pieces and $\sigma = 1$. Figure 8 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.

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 $\mathrm{pen}_{\mathrm{shape}}(m)$ 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: 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), however 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. But this comparison study has to be continued and deepened, both theoretically and practically.

CAPUSHE 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 cope with questions raised by the slope heuristics study. It may be useful for example to compare the two available strategies for the application of the dimension jump: the maximal dimension jump versus the threshold complexity.

The slope heuristics is 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 Sect. 5.2 were obtained from E. Lebarbier (AgroParisTech). The paper has been substantially improved thanks to the reviewers suggestions.

References

Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Proceedings, 2nd Internat. Symp. on Information Theory, pp. 267–281 (1973)

Arlot, S.: Model selection by resampling penalization. Electron. J. Stat. **3**, 557–624 (2009) (electronic)

Arlot, S., Bach, F.: Data-driven calibration of linear estimators with minimal penalties. In: Advances in Neural Information Processing Systems (NIPS), vol. 22, pp. 46–54 (2009)

Arlot, C., Celisse, A.: A survey of cross-validation procedures for model selection. Preprint arXiv:0907.4728v1 (2009)

Arlot, S., Massart, P.: Data-driven calibration of penalties for least-squares regression. J. Mach. Learn. Res. **10**, 245–279 (2009) (electronic). http://www.jmlr.org/papers/volume10/arlot09a/arlot09a.pdf

Baudry, J.P.: Sélection de modèle pour la classification non supervisée. Choix du nombre de classes. PhD thesis, University Paris XI. http://tel.archives-ouvertes.fr/tel-00461550/fr/ (2009)

Baudry, J.P., Celeux, G., Marin, J.M.: Selecting models focussing on the modeller's purpose. In: COMPSTAT 2008: Proceedings in Computational Statistics, pp. 337–348. Physica, Heidelberg (2008)

Biernacki, C., Celeux, G., Govaert, G., Langrognet, F.: Model-based cluster and discriminant analysis with the MIXMOD software. Comput. Stat. Data Anal. **51**, 587–600 (2006)

Birgé, L., Massart, P.: Gaussian model selection. J. Eur. Math. Soc. **3**, 203–268 (2001)

Birgé, L., Massart, P.: Minimal penalties for Gaussian model selection. Probab. Theory Relat. Fields **138**, 33–73 (2006)

Burnham, K., Anderson, D.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd edn. Springer, New York (2002)

Caillerie, C., Michel, B.: Model selection for simplicial approximation. RR 6981, INRIA. http://hal.inria,.fr/inria-00402091/en/ (2009)

Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. Numer. Math. **31**, 377–403 (1978)

Denis, M., Molinari, N.: Choix du nombre de noeuds en régression spline par l'heuristique des pentes. In: 41èmes Journées de Statistique, SFdS, Bordeaux, Bordeaux, France (2009). http://hal.inria.fr/inria-00386618/en/

Fraley, C., Raftery, A.E.: Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST. J. Classif. **20**, 263–286 (2003)

Huber, P.J.: Robust Statistics. Wiley, New York (1981)

Keribin: Consistent estimation of the order of mixture models. Sankhya, Ser. A **62**, 49–66 (2000)

Lebarbier, E.: Detecting multiple change-points in the mean of Gaussian process by model selection. Signal Process. **85**, 717–736 (2005)

Lepez, V.: Some estimation problems related to oil reserves. PhD thesis, University Paris XI. http://tel.archives-ouvertes.fr/tel-00460802/fr/ (2002)

Lerasle, M.: Adaptive density estimation of stationary β-mixing and τ-mixing processes. Math. Methods Stat. **18**, 59–83 (2009a)

Lerasle, M.: Optimal model selection in density estimation. Preprint. arXiv:0910.1654 (2009b)

Lerasle, M.: 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 (2009c)

Mallows, C.L.: Some comments on CP. Technometrics **15**, 661–675 (1973). http://www.jstor.org/stable/1267380

Massart, P.: Concentration inequalities and model selection. In: École d'été de Probabilités de Saint-Flour 2003. Lecture Notes in Mathematics. Springer, Berlin (2007)

Maugis, C., Michel, B.: A non asymptotic penalized criterion for Gaussian mixture model selection. ESAIM Probab. Stat. (2009). doi:10.1051/ps/2009004. http://www.esaim-ps.org/index.php?option=article&access=standard&Itemid=129&url=/articles/ps/pdf/forth/ps0842.pdf

Maugis, C., Michel, B.: Data-driven penalty calibration: a case study for Gaussian mixture model selection. ESAIM Probab. Stat. (2010). doi:10.1051/ps/2010002. http://www.esaim-ps.org/index.php?option=article&access=standard&Itemid=129&url=/articles/ps/pdf/forth/ps0843.pdf

Maugis, C., Celeux, G., Martin-Magniette, M.L.: Variable selection for clustering with Gaussian mixture models. Biometrics **65**, 701–709 (2009)

Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)

Verzelen, N.: Data-driven neighborhood selection of a Gaussian field. RR 6798, INRIA (2009)

Villers, F.: 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 (2007)