Model Selection with the Loss Rank Principle

Marcus Hutter

RSISE @ ANU and SML @ NICTA, Canberra, ACT, 0200, Australia
marcus@hutter1.net www.hutter1.net

Minh-Ngoc Tran

Department of Statistics and Applied Probability,
National University of Singapore, ngoctm@nus.edu.sg

2 March 2010

Abstract

A key issue in statistics and machine learning is to automatically select the "right" model complexity, e.g., the number of neighbors to be averaged over in k nearest neighbor (kNN) regression or the polynomial degree in regression with polynomials. We suggest a novel principle - the Loss Rank Principle (LoRP) - for model selection in regression and classification. It is based on the loss rank, which counts how many other (fictitious) data would be fitted better. LoRP selects the model that has minimal loss rank. Unlike most penalized maximum likelihood variants (AIC, BIC, MDL), LoRP depends only on the regression functions and the loss function. It works without a stochastic noise model, and is directly applicable to any non-parametric regressor, like kNN.

Contents

Keywords

Model selection, loss rank principle, non-parametric regression, classification, general loss function, k nearest neighbors.

1 Introduction

Regression. Consider a regression or classification problem in which we want to determine the functional relationship $y_i \approx f_{\text{true}}(x_i)$ from data $D=\{(x_1,y_1),...,(x_n,y_n)\}$, i.e., we seek a function $r(.|D) \equiv r(D)(.)$ such that $r(x|D) \equiv r(D)(x)$ is close to the unknown $f_{\text{true}}(x)$ for all x. One may define $r(.|D)$ directly, e.g., "average the y values of the k nearest neighbors (kNN) of x in D", or select $r(.|D)$ from a class of functions $\mathcal{F}$ that has smallest (training) error on D. If the class $\mathcal{F}$ is not too large, e.g., the polynomials of fixed reasonable degree d, this often works well.

Model selection. What remains is to select the right model complexity c, like k or d. This selection cannot be based on the training error, since the more complex the model (large d, small k) the better the fit on D (perfect for $d=n$ and $k=1$). This problem is called overfitting, for which various remedies have been suggested.

The most popular ones in practice are based on a test set D' used for selecting the c for which the function $r_c(.|D)$ has smallest (test) error on D', or improved versions like cross-validation [All74]. Typically D' is cut from D, thus reducing the sample size available for regression. Test set methods often work well in practice, but the reduced sample decreases accuracy, which can be a serious problem if n is small. We will not discuss empirical test set methods any further. See [Mac92] for a comparison of cross-validation with Bayesian model selection.

There are also various model selection methods that allow to use all data D for regression. The most popular ones can be regarded as penalized versions of Maximum Likelihood (ML). In addition to the function class $\mathcal{F}_c$ (subscript c belonging to some set indexing the complexity), one has to specify a sampling model $\text{P}(D|f)$, e.g., that the y_i have independent Gaussian distribution with mean $f(x_i)$. ML chooses $r_c(D) = \text{argmax}_{f\in\mathcal{F}_c}\text{P}(D|f)$, Penalized ML (PML) then chooses $\hat{c} = \text{argmin}_c\{-\text{logP}(D|r_c(D))+\text{Penalty}(c)\}$, where the penalty depends on the used approach (MDL [Ris78], BIC [Sch78], AIC [Aka73]). All PML variants rely on a proper sampling model (which may be difficult to establish), ignore (or at least do not tell how to incorporate) a potentially given loss function (see [Yam99, Grü04] for exceptions), are based on distribution-independent penalties (which may result in bad performance for specific distributions), and are typically limited to (semi)parametric models.

Main idea. The main goal of the paper is to establish a criterion for selecting the "best" model complexity c based on regressors r_c given as a black box without insight into the origin or inner structure of r_c, that does not depend on things often not given (like a stochastic noise model), and that exploits what is/should be given (like the loss function, note that the criterion can also be used for loss-function selection, see Section 8). The key observation we exploit is that large classes $\mathcal{F}_c$ or more flexible regressors r_c can fit more data well than more rigid ones. We define the *loss rank* of r_c as the number of other (fictitious) data D' that are fitted better by $r_c(D')$ than D is fitted by $r_c(D)$, as measured by some loss function. The loss rank is large for regressors fitting D not well *and* for too flexible regressors (in both

cases the regressor fits many other D' better). The loss rank has a minimum for not too flexible regressors which fit D not too bad. We claim that minimizing the loss rank is a suitable model selection criterion, since it trades off the quality of fit with the flexibility of the model. Unlike PML, our Loss Rank Principle (LoRP) works without a noise (stochastic sampling) model, and is directly applicable to any non-parametric regressor, like kNN.

Related ideas. There are various other ideas that somehow count fictitious data. In normalized ML [Grü04], the complexity of a stochastic model class is defined as the log sum over all D' of maximum likelihood probabilities. In the luckiness framework for classification [Her02, Chp.4], the loss rank is related to the level of a hypothesis, if the empirical loss is used as an unluckiness function. The empirical Rademacher complexity [Kol01, BBL02] averages over all possible relabeled instances. Finally, instead of considering all D' one could consider only the set of all permutations of $\{y_1,...,y_n\}$, like in permutation tests [ET93]. The test statistic would here be the empirical loss.

Contents. In Section 2, after giving a brief introduction to regression, we formally state LoRP for model selection. Explicit expressions for the loss rank for the important class of linear regressors are derived in Section 3; this class includes kNN, polynomial, linear basis function (LBFR), kernel, projective regression, and some others. In Section 4, we establish optimality properties of LoRP for linear regression, namely model consistency and asymptotic mean efficiency. Experiments are presented in Section 5: We compare LoRP to other selection methods and demonstrate the use of LoRP for some specific problems like choosing tuning parameters in kNN and spline regression. In Section 6 we compare linear LoRP to Bayesian model selection for linear regression with Gaussian noise and prior, and in Section 7 to PML, in particular MDL, BIC, and AIC, and then discuss two trace formulas for the effective dimension. Sections 8-10 can be considered as extension sections. In Section 8 we show how to generalize linear LoRP to non-quadratic loss, in particular to other norms. We also discuss how LoRP can be used to select the loss function itself, in case it is not part of the problem specification. In Section 9 we briefly discuss interpolation. LoRP only depends on the regressor on data D and not on $x \notin \{x_1,...,x_n\}$. We construct canonical regressors for off-data interpolation from regressors given only on-data, in particular for kNN, Kernel, and LBFR, and show that they are canonical. In Section 10 we derive exact expressions for kNN when $\{x_1,...,x_n\}$ forms a discrete d-dimensional hypercube, and discuss the limits $n \to \infty$, $k \to \infty$, and $d \to \infty$. Section 11 contains the conclusions of our work and further considerations that could be elaborated on in the future.

The main idea of LoRP has already been presented at the COLT 2007 conference [Hut07]. In this paper we present LoRP more thoroughly, discover its theoretical properties and evaluate the method through some experiments.

2 The Loss Rank Principle

After giving a brief introduction to regression, classification, model selection, overfitting, and some reoccurring examples, we state our novel Loss Rank Principle for model selection. We first state it for classification (Principle 3 for discrete values), and then generalize it for regression (Principle 5 for continuous values), and exemplify it on two (over-simplistic) artificial Examples 4 and 6. Thereafter we show how to regularize LoRP for realistic regression problems.

Setup and notation. We assume data $D=(\boldsymbol{x},\boldsymbol{y}):=\{(x_1,y_1),...,(x_n,y_n)\}\in(\mathcal{X}\times\mathcal{Y})^n=:\mathcal{D}$ has been observed. We think of the y as having an approximate functional dependence on x, i.e., $y_i\approx f_{\text{true}}(x_i)$, where $\approx$ means that the y_i are distorted by noise from the unknown "true" values $f_{\text{true}}(x_i)$. We will write (x,y) for generic data points, use vector notation $\boldsymbol{x}=(x_1,...,x_n)^\top$ and $\boldsymbol{y}=(y_1,...,y_n)^\top$, and $D'=(\boldsymbol{x}',\boldsymbol{y}')$ for generic (fictitious) data of size n. A full list of abbreviations and notations used throughout the paper is placed in the appendix.

Regression and classification. In regression problems $\mathcal{Y}$ is typically (a subset of) the real set $\mathbb{R}$ or some more general measurable space like $\mathbb{R}^m$. In classification, $\mathcal{Y}$ is a finite set or at least discrete. We impose no restrictions on $\mathcal{X}$. Indeed, $\boldsymbol{x}$ will essentially be fixed and plays only a spectator role, so we will often notationally suppress dependencies on $\boldsymbol{x}$. The goal of regression/classification is to find a function $f_D\in\mathcal{F}\subset\mathcal{X}\to\mathcal{Y}$ "close" to f_{true} based on the past observations D. Or phrased in another way: we are interested in a mapping $r:\mathcal{D}\to\mathcal{F}$ such that $\hat{y}:=r(x|D)\equiv r(D)(x)\equiv f_D(x)\approx f_{\text{true}}(x)$ for all $x\in\mathcal{X}$.

Example 1 (polynomial regression) For $\mathcal{X}=\mathcal{Y}=\mathbb{R}$, consider the set $\mathcal{F}_d:=\{f_{\boldsymbol{w}}(x)=w_dx^{d-1}+...+w_2x+w_1:\boldsymbol{w}\in\mathbb{R}^d\}$ of polynomials of degree $d-1$. Fitting the polynomial to data D, e.g., by least squares regression, we estimate $\boldsymbol{w}$ with $\hat{\boldsymbol{w}}_D$. The regression function $\hat{y}=r_d(x|D)=f_{\hat{\boldsymbol{w}}_D}(x)$ can be written down in closed form (see Example 9). $\diamondsuit$

Example 2 (k nearest neighbors) Let $\mathcal{Y}$ be some vector space like $\mathbb{R}$ and $\mathcal{X}$ be a metric space like $\mathbb{R}^m$ with some (e.g., Euclidean) metric $d(\cdot,\cdot)$. kNN estimates $f_{\text{true}}(x)$ by averaging the y values of the k nearest neighbors $\mathcal{N}_k(x)$ of x in D, i.e., $r_k(x|D)=\frac{1}{k}\sum_{i\in\mathcal{N}_k(x)}y_i$ with $|\mathcal{N}_k(x)|=k$ such that $d(x,x_i)\leq d(x,x_j)$ for all $i\in\mathcal{N}_k(x)$ and $j\notin\mathcal{N}_k(x)$. $\diamondsuit$

Parametric versus non-parametric regression. Polynomial regression is an example of parametric regression in the sense that $r_d(D)$ is the optimal function from a family of functions $\mathcal{F}_d$ indexed by $d<\infty$ real parameters ($\boldsymbol{w}$). In contrast, the kNN regressor r_k is directly given and is not based on a finite-dimensional family of functions. In general, r may be given either directly or be the result of an optimization process.

Loss function. The quality of fit to the data is usually measured by a loss function $\text{Loss}(\boldsymbol{y},\hat{\boldsymbol{y}})$, where $\hat{y}_i=\hat{f}_D(x_i)$ is an estimate of y_i. Often the loss is additive:

$\text{Loss}(\boldsymbol{y},\hat{\boldsymbol{y}}) = \sum_{i=1}^n \text{Loss}(y_i,\hat{y}_i)$. If the class $\mathcal{F}$ is not too large, good regression functions $r(D)$ can be found by minimizing the loss w.r.t. all $f \in \mathcal{F}$. For instance, $r_d(D) = \text{argmin}_{f\in\mathcal{F}_d} \sum_{i=1}^n (y_i - f(x_i))^2$ and $\hat{y} = r_d(x|D)$ in Example 1.

Regression class and loss. In the following we assume a class of regressors $\mathcal{R}$ (whatever their origin), e.g., the kNN regressors $\{r_k : k \in \mathbb{N}\}$ or the least squares polynomial regressors $\{r_d : d \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}\}$. Each regressor r can be thought of as a model. Throughout the paper, we use the terms "regressor" and "model" interchangeably. Note that unlike $f \in \mathcal{F}$, regressors $r \in \mathcal{R}$ are not functions of x alone but depend on all observations D, in particular on $\boldsymbol{y}$. Like for functions f, we can compute the empirical loss of each regressor $r \in \mathcal{R}$:

$$\text{Loss}_r(D) \;\equiv\; \text{Loss}_r(\boldsymbol{y}|\boldsymbol{x}) \;:=\; \text{Loss}(\boldsymbol{y},\hat{\boldsymbol{y}}) \;=\; \sum_{i=1}^n \text{Loss}(y_i, r(x_i|\boldsymbol{x},\boldsymbol{y}))$$

where $\hat{y}_i = r(x_i|D)$ in the third expression, and the last expression holds in case of additive loss.

Overfitting. Unfortunately, minimizing Loss_r w.r.t. r will typically *not* select the "best" overall regressor. This is the well-known overfitting problem. In case of polynomials, the classes $\mathcal{F}_d \subset \mathcal{F}_{d+1}$ are nested, hence Loss_{r_d} is monotone decreasing in d with $\text{Loss}_{r_n} \equiv 0$ perfectly fitting the data. In case of kNN, Loss_{r_k} is more or less an increasing function in k with perfect regression on D for $k = 1$, since no averaging takes place. In general, $\mathcal{R}$ is often indexed by a "flexibility" or smoothness or complexity parameter, which has to be properly determined. The more flexible r is, the closer it can fit the data. Hence such r has smaller empirical loss, but is not necessarily better since it has higher variance. Clearly, too inflexible r also lead to a bad fit ("high bias").

Main goal. The main goal of the paper is to establish a selection criterion in order to specify the smallest model to which f_{true} belongs or is close to, and simultaneously determine the "best" fitting function $r(D)$. The criterion

- is based on r given as a black box that does not require insight into the origin or inner structure of r;
- does not depend on things often not given (like a stochastic noise model); and
- exploits what is or should be given (like the loss function).

Definition of loss rank. We first consider discrete $\mathcal{Y}$ (i.e., classification), fix $\boldsymbol{x}$, $\boldsymbol{y}$ is the observed data and $\boldsymbol{y}'$ are fictitious others. The key observation we exploit is that a more flexible r can fit more data $D' \in \mathcal{D}$ well than a more rigid one. The more flexible r is, the smaller the empirical loss $\text{Loss}_r(\boldsymbol{y}|\boldsymbol{x})$ is. Instead of minimizing the unsuitable $\text{Loss}_r(\boldsymbol{y}|\boldsymbol{x})$ w.r.t. r, we could ask how many $\boldsymbol{y}' \in \mathcal{Y}^n$ lead to smaller Loss_r than $\boldsymbol{y}$. We define the loss rank of r (w.r.t. $\boldsymbol{y}$) as the number of $\boldsymbol{y}' \in \mathcal{Y}^n$ with smaller or equal empirical loss than $\boldsymbol{y}$:

$$\text{Rank}_r(\boldsymbol{y}|\boldsymbol{x}) \equiv \text{Rank}_r(L) := \#\{\boldsymbol{y}' \in \mathcal{Y}^n : \text{Loss}_r(\boldsymbol{y}'|\boldsymbol{x}) \leq L\} \;\text{ with }\; L := \text{Loss}_r(\boldsymbol{y}|\boldsymbol{x}) \quad (1)$$

We claim that the loss rank of r is a suitable model selection measure. For (1) to make sense, we have to assume (and will later assure) that $\text{Rank}_r(L)<\infty$, i.e., there are only finitely many $\boldsymbol{y}'\in\mathcal{Y}^n$ having loss smaller than L.

Since the logarithm is a strictly monotone increasing function, we can also consider the logarithmic rank $\text{LR}_r(\boldsymbol{y}|\boldsymbol{x}) := \log\text{Rank}_r(\boldsymbol{y}|\boldsymbol{x})$, which will be more convenient.

Principle 3 (LoRP for classification) *For discrete $\mathcal{Y}$, the best classifier/regressor $r:\mathcal{D}\times\mathcal{X}\to\mathcal{Y}$ in some class $\mathcal{R}$ for data $D=(\boldsymbol{x},\boldsymbol{y})$ is the one with the smallest loss rank:*

$$r^{best} \;=\; \arg\min_{r\in\mathcal{R}} \text{LR}_r(\boldsymbol{y}|\boldsymbol{x}) \;\equiv\; \arg\min_{r\in\mathcal{R}} \text{Rank}_r(\boldsymbol{y}|\boldsymbol{x}) \tag{2}$$

where Rank_r *is defined in* (1).

We give a simple example for which we can compute all ranks by hand to help the reader better grasp how the principle works.

Example 4 (simple discrete) Consider $\mathcal{X}=\{1,2\}$, $\mathcal{Y}=\{0,1,2\}$, and two points $D=\{(1,1),(2,2)\}$ lying on the diagonal $x=y$, with polynomial (zero, constant, linear) least squares regressors $\mathcal{R}=\{r_0,r_1,r_2\}$ (see Ex.1). r_0 is simply 0, r_1 the y-average, and r_2 the line through points $(1,y_1)$ and $(2,y_2)$. This, together with the quadratic Loss for generic $\boldsymbol{y}'$ and observed $\boldsymbol{y}=(1,2)$ and fixed $\boldsymbol{x}=(1,2)$, is summarized in the following table

d	$r_d(x\|\boldsymbol{x},\boldsymbol{y}')$	$\text{Loss}_d(\boldsymbol{y}'\|\boldsymbol{x})$	$\text{Loss}_d(D)$
0	0	$y_1'^2+y_2'^2$	5
1	$\frac{1}{2}(y_1'+y_2')$	$\frac{1}{2}(y_2'-y_1')^2$	$\frac{1}{2}$
2	$(y_2'-y_1')(x-1)+y_1'$	0	0

From the Loss we can easily compute the Rank for all nine $\boldsymbol{y}'\in\{0,1,2\}^2$. Equal rank due to equal loss is indicated by a "=" in the table below. Whole equality groups are actually assigned the rank of their right-most member, e.g., for $d=1$ the ranks of $(y_1',y_2')=(0,1),(1,0),(2,1),(1,2)$ are all 7 (and not 4,5,6,7).

	$\text{Rank}_{r_d}(y_1'y_2'\|12)$									
d	1	2	3	4	5	6	7	8	9	$\text{Rank}_{r_d}(D)$
0	$y_1'y_2'=00<01=10<11<02=20<21=\mathbf{12}<22$									8
1	$y_1'y_2'=00=11=22<01=10=21=\mathbf{12}<02=20$									7
2	$y_1'y_2'=00=01=02=10=11=20=21=22=\mathbf{12}$									9

So LoRP selects r_1 as best regressor, since it has minimal rank on D. r_0 fits D too badly and r_2 is too flexible (perfectly fits all D'). $\diamondsuit$

LoRP for continuous $\mathcal{Y}$. We now consider the case of continuous or measurable spaces $\mathcal{Y}$, i.e., normal regression problems. We assume $\mathcal{Y}=\mathbb{R}$ in the following exposition, but the idea and resulting principle hold for more general measurable

spaces like $\mathbb{R}^m$. We simply reduce the model selection problem to the discrete case by considering the discretized space $\mathcal{Y}_\varepsilon = \varepsilon\mathbb{Z}$ for small $\varepsilon > 0$ and discretize $\boldsymbol{y} \rightsquigarrow \boldsymbol{y}_\varepsilon \in \varepsilon\mathbb{Z}^n$ ("$\rightsquigarrow$" means "is replaced by"). Then $\text{Rank}_r^\varepsilon(L) := \#\{\boldsymbol{y}'_\varepsilon \in \mathcal{Y}_\varepsilon^n : \text{Loss}_r(\boldsymbol{y}'_\varepsilon|\boldsymbol{x}) \leq L\}$ with $L = \text{Loss}_r(\boldsymbol{y}_\varepsilon|\boldsymbol{x})$ counting the number of ε-grid points in the set

$$V_r(L) \;:=\; \{\boldsymbol{y}' \in \mathcal{Y}^n : \text{Loss}_r(\boldsymbol{y}'|\boldsymbol{x}) \leq L\} \tag{3}$$

which we assume (and later assure) to be finite, analogous to the discrete case. Hence $\text{Rank}_r^\varepsilon(L) \cdot \varepsilon^n$ is an approximation of the *loss volume* $|V_r(L)|$ of set $V_r(L)$, and typically $\text{Rank}_r^\varepsilon(L) \cdot \varepsilon^n = |V_r(L)| \cdot (1 + O(\varepsilon)) \to |V_r(L)|$ for $\varepsilon \to 0$. Taking the logarithm we get $\text{LR}_r^\varepsilon(\boldsymbol{y}|\boldsymbol{x}) = \log\text{Rank}_r^\varepsilon(L) = \log|V_r(L)| - n\log\varepsilon + O(\varepsilon)$. Since $n\log\varepsilon$ is independent of r, we can drop it in comparisons like (2). So for $\varepsilon \to 0$ we can define the log-loss "rank" simply as the log-volume

$$\text{LR}_r(\boldsymbol{y}|\boldsymbol{x}) \;:=\; \log|V_r(L)|, \quad \text{where} \quad L := \text{Loss}_r(\boldsymbol{y}|\boldsymbol{x}) \tag{4}$$

Principle 5 (LoRP for regression) *For measurable $\mathcal{Y}$, the best regressor $r : \mathcal{D} \times \mathcal{X} \to \mathcal{Y}$ in some class $\mathcal{R}$ for data $D = (\boldsymbol{x}, \boldsymbol{y})$ is the one with the smallest loss volume:*

$$r^{best} \;=\; \arg\min_{r \in \mathcal{R}} \text{LR}_r(\boldsymbol{y}|\boldsymbol{x}) \;\equiv\; \arg\min_{r \in \mathcal{R}} |V_r(L)|$$

where LR, V_r, *and* L *are defined in* (3) *and* (4), *and* $|V_r(L)|$ *is the volume of* $V_r(L) \subseteq \mathcal{Y}^n$.

For discrete $\mathcal{Y}$ with counting measure we recover the discrete LoRP (Principle 3).

Example 6 (simple continuous) Consider Example 4 but with interval $\mathcal{Y} = [0,2]$. The first table remains unchanged, while the second table becomes

d	$V_d(L) = \{\boldsymbol{y}' \in [0,2]^2 : ...\}$	$\lvert V_d(L)\rvert$	$\text{Loss}_d(D)$	$\lvert V_d(\text{Loss}_d(D))\rvert$
0	${y'_1}^2 + {y'_2}^2 \leq L$	$\frac{\pi}{4}L$ if $L \leq 4$; 4 if $L \geq 8$; $2\sqrt{L-4} + L(\frac{\pi}{4} - \cos^{-1}(\frac{2}{\sqrt{L}}))$ else	5	$\doteq 3.6$
1	$\frac{1}{2}(y'_2 - y'_1)^2 \leq L$	$4\sqrt{2L} - 2L$ if $L \leq 2$; 4 if $L \geq 2$	$\frac{1}{2}$	3
2	$0 \leq L$	4	0	4

So LoRP again selects r_1 as best regressor, since it has smallest loss volume on D. $\diamondsuit$

Infinite rank or volume. Often the loss rank/volume will be infinite, e.g., if we had chosen $\mathcal{Y} = \mathbb{Z}$ in Ex.4 or $\mathcal{Y} = \mathbb{R}$ in Ex.6. There are various potential remedies. We could modify (a) the regressor r or (b) the Loss to make LR_r finite, (c) the Loss Rank Principle itself, or (d) find problem-specific solutions. Regressors r with infinite rank might be rejected for philosophical or pragmatic reasons. We will briefly consider (a) for linear regression later, but to fiddle around with r in a generic (blackbox way) seems difficult. We have no good idea how to tinker with LoRP (c), and also a patched LoRP may be less attractive. For kNN on a grid we

later use remedy (d). While in (decision) theory, the application's goal determines the loss, in practice the loss is often more determined by convenience or rules of thumb. So the Loss (b) seems the most inviting place to tinker with. A very simple modification is to add a small penalty term to the loss.

$$\text{Loss}_r(\boldsymbol{y}|\boldsymbol{x}) \;\rightsquigarrow\; \text{Loss}_r^\alpha(\boldsymbol{y}|\boldsymbol{x}) := \text{Loss}_r(\boldsymbol{y}|\boldsymbol{x}) + \alpha\|\boldsymbol{y}\|^2, \quad \alpha > 0 \text{ "small"} \tag{5}$$

The Euclidean norm $\|\boldsymbol{y}\|^2 := \sum_{i=1}^n y_i^2$ is default, but other (non)norm regularizations are possible. The regularized $\text{LR}_r^\alpha(\boldsymbol{y}|\boldsymbol{x})$ based on Loss_r^α is always finite, since $\{\boldsymbol{y}: \|\boldsymbol{y}\|^2 \leq L\}$ has finite volume. An alternative penalty $\alpha\hat{\boldsymbol{y}}^\top\hat{\boldsymbol{y}}$, quadratic in the regression estimates $\hat{y}_i = r(x_i|\boldsymbol{x},\boldsymbol{y})$ is possible if r is unbounded in every $\boldsymbol{y}\to\infty$ direction.

A scheme trying to determine a single (flexibility) parameter (like d and k in the above examples) would be of no use if it depended on one (or more) other unknown parameters (α), since varying through the unknown parameter leads to any (non)desired result. Since LoRP seeks the r of smallest rank, it is natural to also determine $\alpha = \alpha_{\min}$ by minimizing LR_r^α w.r.t. α. The good news is that this leads to meaningful results. Interestingly, as we will see later, a clever choice of α may also result in alternative optimalities of the selection procedure.

3 LoRP for y-Linear Models

In this section we consider the important class of *y-linear* regressions with quadratic loss function. By "y-linear regression", we mean the linearity is only assumed in y and the dependence on x can be arbitrary. This class is richer than it may appear. It includes the normal linear regression model, kNN (Example 7), kernel (Example 8), and many other regression models. For y-linear regression and $\mathcal{Y} = \mathbb{R}$, the loss rank is the volume of an n-dimensional ellipsoid, which can efficiently be computed in time $O(n^3)$ (Theorem 10). For the special case of projective regression, e.g., linear basis function regression (Example 9), we can even determine the regularization parameter α analytically (Theorem 11).

y-Linear regression. We assume $\mathcal{Y} = \mathbb{R}$ in this section; generalization to $\mathbb{R}^m$ is straightforward. A y-linear regressor r can be written in the form

$$\hat{y} \;=\; r(x|\boldsymbol{x},\boldsymbol{y}) \;=\; \sum_{j=1}^n m_j(x,\boldsymbol{x})y_j \quad \forall x \in \mathcal{X} \quad \text{and some} \quad m_j : \mathcal{X}\times\mathcal{X}^n \to \mathbb{R} \tag{6}$$

Particularly interesting is r for $x = x_1,...,x_n$.

$$\hat{y}_i \;=\; r(x_i|\boldsymbol{x},\boldsymbol{y}) \;=\; \sum_j M_{ij}(\boldsymbol{x})y_j \quad \text{with} \quad M : \mathcal{X}^n \to \mathbb{R}^{n\times n} \tag{7}$$

where matrix $M_{ij}(\boldsymbol{x}) = m_j(x_i,\boldsymbol{x})$. Since LoRP needs r only on the training data $\boldsymbol{x}$, we only need M.

Example 7 (kNN ctd.) For kNN of Ex.2 we have $m_j(x,\boldsymbol{x}) = \frac{1}{k}$ if $j \in \mathcal{N}_k(x)$ and 0 else, and $M_{ij}(\boldsymbol{x}) = \frac{1}{k}$ if $j \in \mathcal{N}_k(x_i)$ and 0 else. $\diamondsuit$

Example 8 (kernel regression) Kernel regression takes a weighted average over $\boldsymbol{y}$, where the weight of y_j to y is proportional to the similarity of x_j to x, measured by a kernel $K(x,x_j)$, i.e., $m_j(x,\boldsymbol{x})=K(x,x_j)/\sum_{j=1}^n K(x,x_j)$. For example the Gaussian kernel for $\mathcal{X}=\mathbb{R}^m$ is $K(x,x_j)=\mathrm{e}^{-\|x-x_j\|_2^2/2\sigma^2}$. The width σ controls the smoothness of the kernel regressor, and LoRP selects the real-valued "complexity" parameter σ. $\diamondsuit$

Example 9 (linear basis function regression, LBFR) Let $\phi_1(x),...,\phi_d(x)$ be a set or vector of "basis" functions often called "features". We place no restrictions on $\mathcal{X}$ or $\boldsymbol{\phi}:\mathcal{X}\to\mathbb{R}^d$. Consider the class of functions linear in $\boldsymbol{\phi}$:

$$\mathcal{F}_d \;=\; \{f_{\boldsymbol{w}}(x)=\textstyle\sum_{a=1}^d w_a\phi_a(x)=\boldsymbol{w}^\top\boldsymbol{\phi}(x):\boldsymbol{w}\in\mathbb{R}^d\}$$

For instance, for $\mathcal{X}=\mathbb{R}$ and $\phi_a(x)=x^{a-1}$ we would recover the polynomial regression Example 1. For quadratic loss function $\mathrm{Loss}(y_i,\hat{y}_i)=(y_i-\hat{y}_i)^2$ we have

$$\mathrm{Loss}_{\boldsymbol{w}}(\boldsymbol{y}|\boldsymbol{\phi}) \;:=\; \sum_{i=1}^n (y_i-f_{\boldsymbol{w}}(x_i))^2 \;=\; \boldsymbol{y}^\top\boldsymbol{y}-2\boldsymbol{y}^\top\Phi\boldsymbol{w}+\boldsymbol{w}^\top B\boldsymbol{w}$$

where matrix Φ is defined by $\Phi_{ia}=\phi_a(x_i)$ and B is a symmetric matrix with $B_{ab}=\sum_{i=1}^n\phi_a(x_i)\phi_b(x_i)=[\Phi^\top\Phi]_{ab}$. The loss is quadratic in $\boldsymbol{w}$ with minimum at $\boldsymbol{w}=B^{-1}\Phi^\top\boldsymbol{y}$. So the least squares regressor is $\hat{y}=\boldsymbol{y}^\top\Phi B^{-1}\boldsymbol{\phi}(x)$, hence $m_j(x,\boldsymbol{x})=(\Phi B^{-1}\boldsymbol{\phi}(x))_j$ and $M(\boldsymbol{x})=\Phi B^{-1}\Phi^\top$. $\diamondsuit$

Consider now a general linear regressor M with quadratic loss and quadratic penalty

$$\begin{aligned}\mathrm{Loss}_M^\alpha(\boldsymbol{y}|\boldsymbol{x}) \;&=\; \sum_{i=1}^n\Big(y_i-\textstyle\sum_{j=1}^n M_{ij}y_j\Big)^2+\alpha\|\boldsymbol{y}\|^2 \;=\; \boldsymbol{y}^\top S_\alpha\boldsymbol{y},\\ \text{where}^1\quad S_\alpha \;&=\; (I-M)^\top(I-M)+\alpha I \qquad (8)\end{aligned}$$

(I is the identity matrix). S_α is a symmetric matrix. For $\alpha>0$ it is positive definite and for $\alpha=0$ positive semidefinite. If $\lambda_1,...,\lambda_n\geq 0$ are the eigenvalues of S_0, then $\lambda_i+\alpha$ are the eigenvalues of S_α. $V(L)=\{\boldsymbol{y}'\in\mathbb{R}^n:\boldsymbol{y}'^\top S_\alpha\boldsymbol{y}'\leq L\}$ is an ellipsoid with the eigenvectors of S_α being the main axes and $\sqrt{L/(\lambda_i+\alpha)}$ being their length. Hence the volume is

$$|V(L)| \;=\; v_n\prod_{i=1}^n\sqrt{\frac{L}{\lambda_i+\alpha}} \;=\; \frac{v_n L^{n/2}}{\sqrt{\det S_\alpha}}$$

where $v_n=\pi^{n/2}/\frac{n}{2}!$ is the volume of the n-dimensional unit sphere, $z!:=\Gamma(z+1)$, and det is the determinant. Taking the logarithm we get

$$\mathrm{LR}_M^\alpha(\boldsymbol{y}|\boldsymbol{x}) \;=\; \log|V(\mathrm{Loss}_M^\alpha(\boldsymbol{y}|\boldsymbol{x}))| \;=\; \tfrac{n}{2}\log(\boldsymbol{y}^\top S_\alpha\boldsymbol{y})-\tfrac{1}{2}\log\det S_\alpha+\log v_n \qquad (9)$$

Since v_n is independent of α and M it is possible to drop v_n. Consider now a class of linear regressors $\mathcal{M}=\{M\}$, e.g., the kNN regressors $\{M_k:k\in\mathbb{N}\}$ or the d-dimensional linear basis function regressors $\{M_d:d\in\mathbb{N}_0\}$.

[1]The mentioned alternative penalty $\alpha\|\hat{\boldsymbol{y}}\|^2$ would lead to $S_\alpha=(I-M)^\top(I-M)+\alpha M^\top M$. For LBFR, penalty $\alpha\|\hat{\boldsymbol{w}}\|^2$ is popular (ridge regression). Apart from being limited to parametric regression, it has the disadvantage of not being reparametrization invariant. For instance, scaling $\phi_a(x)\rightsquigarrow\gamma_a\phi_a(x)$ does not change the class $\mathcal{F}_d$, but changes the ridge regressor.

Theorem 10 (LoRP for y-linear regression) *For $\mathcal{Y}=\mathbb{R}$, the best linear regressor $M:\mathcal{X}^n\to\mathbb{R}^{n\times n}$ in some class $\mathcal{M}$ for data $D=(\boldsymbol{x},\boldsymbol{y})$ is*

$$M^{best} \;=\; \mathop{\arg\min}_{M\in\mathcal{M},\alpha\geq 0}\{\tfrac{n}{2}\log(\boldsymbol{y}^\top S_\alpha\boldsymbol{y})-\tfrac{1}{2}\log\det S_\alpha\} \;=\; \mathop{\arg\min}_{M\in\mathcal{M}\;\alpha\geq 0}\Big\{\frac{\boldsymbol{y}^\top S_\alpha\boldsymbol{y}}{(\det S_\alpha)^{1/n}}\Big\} \quad (10)$$

where $S_\alpha=S_\alpha(M)$ is defined in (8).

The last expression shows that linear LoRP minimizes the Loss times the geometric average of the squared axes lengths of ellipsoid $V(1)$. Note that M^{best} depends on $\boldsymbol{y}$ unlike the $M\in\mathcal{M}$.

Nullspace of S_0. If M has an eigenvalue 1, then $S_0=(I-M)^\top(I-M)$ has a zero eigenvalue and $\alpha>0$ is necessary, since $\det S_0=0$. Actually this is true for most practical M. Most linear regressors are invariant under a constant shift of $\boldsymbol{y}$, i.e., $r(x|\boldsymbol{x},\boldsymbol{y}+c)=r(x|\boldsymbol{x},\boldsymbol{y})+c$, which implies that M has eigenvector $(1,...,1)^\top$ with eigenvalue 1. This can easily be checked for kNN (Ex.2), kernel (Ex.8), and LBFR (Ex.9). Such a generic 1-eigenvector effecting all $M\in\mathcal{M}$ could easily and maybe should be filtered out by considering only the orthogonal space or dropping these $\lambda_i=0$ when computing $\det S_0$. The 1-eigenvectors that depend on M are the ones where we really need a regularizer $\alpha>0$. For instance, M_d in LBFR has d eigenvalues 1, and M_{kNN} has as many eigenvalues 1 as there are disjoint components in the graph determined by the edges $M_{ij}>0$. In general we need to find the optimal α numerically. If M is a projection we can find α_m analytically.

Numerical approximation of $(\det S_\alpha)^{1/n}$ and the computational complexity of linear LoRP. For each α and candidate model, the determinant of S_α in the general case can be computed in time $O(n^3)$. Often M is a very sparse matrix (like in kNN) or can be well approximated by a sparse matrix (like for kernel regression), which allows us to approximate $\det S_\alpha$ sometimes in linear time [Reu02]. To search the optimal α and M, the computational cost depends on the range of α we search and the number of candidate models we have.

Projective regression. Consider a projection matrix $M=P=P^2$ with $d(=\text{tr}P)$ eigenvalues 1, and $n-d$ zero eigenvalues. For instance, $M=\Phi B^{-1}\Phi^\top$ of LBFR Ex.9 is such a matrix. This implies that S_α has d eigenvalues α and $n-d$ eigenvalues $1+\alpha$, thus $\det S_\alpha=\alpha^d(1+\alpha)^{n-d}$. Let $\rho=\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|^2/\|\boldsymbol{y}\|^2$, then $\boldsymbol{y}^\top S_\alpha\boldsymbol{y}=(\rho+\alpha)\boldsymbol{y}^\top\boldsymbol{y}$ and

$$\text{LR}_P^\alpha=\tfrac{n}{2}\log\boldsymbol{y}^\top\boldsymbol{y}+\tfrac{n}{2}\log(\rho+\alpha)-\tfrac{d}{2}\log\alpha-\tfrac{n-d}{2}\log(1+\alpha). \quad (11)$$

Solving $\partial\text{LR}_P^\alpha/\partial\alpha=0$ w.r.t. α we get a minimum at $\alpha=\alpha_m:=\frac{\rho d}{(1-\rho)n-d}$ provided that $1-\rho>d/n$. After some algebra we get

$$\text{LR}_P^{\alpha_m}=\tfrac{n}{2}\log\boldsymbol{y}^\top\boldsymbol{y}-\tfrac{n}{2}\text{KL}(\tfrac{d}{n}\|1-\rho),\quad\text{where}\quad \text{KL}(p\|q):=p\log\tfrac{p}{q}+(1-p)\log\tfrac{1-p}{1-q} \quad (12)$$

is the relative entropy or the Kullback-Leibler divergence. Note that (12) is still valid without the condition $1-\rho>d/n$ (the term $\log((1-\rho)n-d)$ has been canceled

in the derivation). What we need when using (12) is that $d<n$ and $\rho<1$, which are very reasonable in practice. Interestingly, if we use the penalty $\alpha\|\hat{\boldsymbol{y}}\|^2$ instead of $\alpha\|\boldsymbol{y}\|^2$, the loss rank then has the same expression as (12) without any condition[2].

Minimizing $\mathrm{LR}_P^{\alpha_m}$ w.r.t. P is equivalent to maximizing $\mathrm{KL}(\frac{d}{n}\|1-\rho)$. The term ρ is a measure of fit. If d increases, then ρ decreases and otherwise. We are seeking a tradeoff between the model complexity d and the measure of fit ρ, and LoRP suggests the optimal tradeoff by maximizing KL.

Theorem 11 (LoRP for projective regression) *The best projective regressor $P:\mathcal{X}^n\to\mathbb{R}^{n\times n}$ with $P=P^2$ in some projective class $\mathcal{P}$ for data $D=(\boldsymbol{x},\boldsymbol{y})$ is*

$$P^{best} \;=\; \arg\max_{P\in\mathcal{P}} \mathrm{KL}(\tfrac{\mathrm{tr}P(\boldsymbol{x})}{n}\|\tfrac{\boldsymbol{y}^\top P(\boldsymbol{x})\boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}}). \tag{13}$$

4 Optimality Properties of LoRP for Variable Selection

In the previous sections, LoRP was stated for general-purpose model selection. By restricting attention to linear regression models, we will point out in this section some theoretical properties of LoRP for variable (also called feature or attribute) selection.

Variable selection is probably the most fundamental and important topic in linear regression analysis. At the initial stage of modeling, a large number of potential covariates are often introduced; one then has to select a smaller subset of the covariates to fit/interpret the data. There are two main goals of variable selection, one is model identification, the other is regression estimation. The former aims at identifying the true subset generating the data, while the latter aims at estimating efficiently the regression function, i.e., selecting a subset that has the minimum mean squared error loss. Note that whether or not there is a selection criterion achieving simultaneously these two goals is still an open question [Yan05, Grü04]. We show that with the optimal parameter α (defined as α_m that minimizes the loss rank LR_M^α in α), LoRP satisfies the first goal, while with a suitable choice of α, LoRP satisfies the second goal.

Given $d+1$ potential covariates $X_0\equiv 1,X_1,...,X_d$ and a response variable Y, let $X=\boldsymbol{x}$ be a non-random design matrix of size $n\times(d+1)$ and $\boldsymbol{y}$ be a response vector respectively (if $\boldsymbol{y}$ and X are centered, then the covariate 1 can be omitted from the models). Denote by $\mathcal{S}=\{0,j_1,...j_{|\mathcal{S}|-1}\}$ the candidate model that has covariates $X_0,X_{j_1},...,X_{j_{|\mathcal{S}|-1}}$. Under a proposed model $\mathcal{S}$, we can write

$$\boldsymbol{y} \;=\; X_{\mathcal{S}}\boldsymbol{\beta}_{\mathcal{S}}+\sigma_{\mathcal{S}}\boldsymbol{\epsilon}$$

[2]Then $S_\alpha=(I_n-P)^\top(I_n-P)+\alpha P^\top P=I_n+(\alpha-1)P$ has d eigenvalues α and $n-d$ eigenvalues 1, thus $\det(S_\alpha)=\alpha^d$. The loss rank $\mathrm{LR}_P^\alpha=\frac{n}{2}\log\boldsymbol{y}^\top\boldsymbol{y}+\frac{n}{2}\log(1+(\alpha-1)(1-\rho))-\frac{d}{2}\log\alpha$ is minimized at $\alpha_m=\frac{\rho d}{(1-\rho)(n-d)}$. After some algebra we get the same expression of $\mathrm{LR}_P^{\alpha_m}$ as (12).

where ϵ is noise with expectation $\mathbf{E}[\boldsymbol{\epsilon}]=0$ and covariance $\mathrm{Cov}(\boldsymbol{\epsilon})=I_n$, $\sigma_{\mathcal{S}}>0$, $\boldsymbol{\beta}_{\mathcal{S}}=(\beta_0,\beta_{j_1},...,\beta_{j_{|\mathcal{S}|-1}})^\top$, and $X_{\mathcal{S}}$ is the $n\times|\mathcal{S}|$ design matrix obtained from X by removing the $(j+1)$st column for all $j\notin\mathcal{S}$.

Model consistency of LoRP for variable selection. The ordinary least squares (OLS) fitted vector under model $\mathcal{S}$ is $\hat{\boldsymbol{y}}_{\mathcal{S}}=M_{\mathcal{S}}\boldsymbol{y}$ with $M_{\mathcal{S}}=X_{\mathcal{S}}(X_{\mathcal{S}}^\top X_{\mathcal{S}})^{-1}X_{\mathcal{S}}^\top$ being a projection matrix. From Theorem 11 the best subset chosen by LoRP is

$$\hat{\mathcal{S}}_n \;=\; \arg\min_{\mathcal{S}} \mathrm{LR}_{\mathcal{S}}^{\alpha_m} \;=\; \arg\max_{\mathcal{S}}\{\mathrm{KL}(\tfrac{|\mathcal{S}|}{n}\|1-\rho_{\mathcal{S}})\},\quad \rho_{\mathcal{S}}=\tfrac{\|\boldsymbol{y}-\hat{\boldsymbol{y}}_{\mathcal{S}}\|^2}{\|\boldsymbol{y}\|^2}.$$

The term $\rho_{\mathcal{S}}$ is a measure of fit. It will be very close to 0 if model $\mathcal{S}$ is big, otherwise, it will be close to 1 if $\mathcal{S}$ is too small. Therefore, it is reasonable to consider only cases in which $\rho_{\mathcal{S}}$ is bounded away from 0 and 1. In order to prove the theoretical properties of LoRP, we need the following technical assumption.

(A) For each candidate model $\mathcal{S}$, $\rho_{\mathcal{S}}$ is bounded away from 0 and 1, i.e., there are constants c_1 and c_2 such that $0<c_1\leq\rho_{\mathcal{S}}\leq c_2<1$ with probability 1 (w.p.1).

Let $\hat{\sigma}_{\mathcal{S}}^2=\|\boldsymbol{y}-\hat{\boldsymbol{y}}_{\mathcal{S}}\|^2/n$ and $\mathcal{S}_{\text{null}}=\{0\}$. It is easy to see that for every $\mathcal{S}$

$$1-\rho_{\mathcal{S}}=\|\hat{\boldsymbol{y}}_{\mathcal{S}}\|^2/\|\boldsymbol{y}\|^2,\qquad n\hat{\sigma}_{\mathcal{S}}^2=\rho_{\mathcal{S}}\|\boldsymbol{y}\|^2,\qquad n\,\bar{\boldsymbol{y}}^2=\|\hat{\boldsymbol{y}}_{\mathcal{S}_{\text{null}}}\|^2\leq\|\hat{\boldsymbol{y}}_{\mathcal{S}}\|^2\leq\|\boldsymbol{y}\|^2 \quad (14)$$

where $\bar{\boldsymbol{y}}$ denotes the arithmetic mean $\sum_{i=1}^n y_i/n$. Assumption (A) follows from

(A') $0<\liminf_{n\to\infty}(\bar{\boldsymbol{y}})^2\leq\limsup_{n\to\infty}(\frac{1}{n}\|\boldsymbol{y}\|^2)<\infty$ and $\forall\mathcal{S}\colon\hat{\sigma}_{\mathcal{S}}^2\to\sigma_{\mathcal{S}}^2>0$ w.p.1.

The first condition of (A') is obviously very mild and satisfied in almost all cases in practice. The second one is routinely used to derive asymptotic properties of model selection criteria (e.g., Theorem 2 of [Sha97] and Condition 1 of [WLT07]).

Lemma 12 (LoRP for variable selection) *The loss rank of model $\mathcal{S}$ is*

$$\mathrm{LR}_{\mathcal{S}} \;\equiv\; \mathrm{LR}_{\mathcal{S}}^{\alpha_m} \;=\; \tfrac{n}{2}\log(n\hat{\sigma}_{\mathcal{S}}^2)+\tfrac{n}{2}H(\tfrac{|\mathcal{S}|}{n})+\tfrac{d}{2}\log\tfrac{1-\rho_{\mathcal{S}}}{\rho_{\mathcal{S}}} \quad (15)$$

where $\rho_{\mathcal{S}}$ and $\hat{\sigma}_{\mathcal{S}}^2$ are defined in (14), and $H(p):=-p\log p-(1-p)\log(1-p)$ is the entropy of p. Under Assumption (A) or (A'), after neglecting constants independent of $\mathcal{S}$, the loss rank of model $\mathcal{S}$ has the form

$$\mathrm{LR}_{\mathcal{S}} \;=\; \tfrac{n}{2}\log\hat{\sigma}_{\mathcal{S}}^2+\tfrac{|\mathcal{S}|}{2}\log n+O_{\mathrm{P}}(1), \quad (16)$$

where $O_{\mathrm{P}}(1)$ denotes a bounded random variable w.p.1.

Proof. Inserting $\boldsymbol{y}^\top\boldsymbol{y}=n\hat{\sigma}_{\mathcal{S}}^2/\rho_{\mathcal{S}}$ into (12) and rearranging terms gives (15). By Assumption (A) the last term in (15) is bounded w.p.1. Taylor expansion $\log(1-p)=-p+O(p^2)$ implies $H(p)/p+\log p\to 1$, hence $\frac{n}{2}H(\frac{|\mathcal{S}|}{n})=\frac{|\mathcal{S}|}{2}\log n+O(1)$. Finally, dropping the $\mathcal{S}$-independent term $\frac{n}{2}\log n$ from (15) gives (16). ■

This lemma implies that the loss rank $\mathrm{LR}_{\mathcal{S}}$ here is a BIC-type criterion, thus we immediately can state without proof the following theorem which is the well-known model consistency of BIC-type criteria (interested readers can find the routine proof in, for example, [Cha06]).

Theorem 13 (Model consistency) *Under Assumption (A) or (A'), LoRP is model consistent for variable selection in the sense that the probability of selecting the true model goes to 1 for data size $n\to\infty$.*

The optimal regression estimation of LoRP. The second goal of model selection is often measured by the (asymptotic) mean efficiency [Shi83] which is briefly defined as follows. Let $\mathcal{S}_T$ denote the true model (which may contain an infinite number of covariates). For a candidate model $\mathcal{S}$, let $L_n(\mathcal{S})=\|X_{\mathcal{S}_T}\boldsymbol{\beta}_{\mathcal{S}_T}-X_{\mathcal{S}}\hat{\boldsymbol{\beta}}_{\mathcal{S}}\|^2$ be the squared loss where $\hat{\boldsymbol{\beta}}_{\mathcal{S}}$ is the OLS estimate, and $R_n(\mathcal{S})=\mathbf{E}[L_n(\mathcal{S})]$ be the risk. The mean efficiency of a selection criterion δ is defined by the ratio

$$\mathrm{eff}(\delta)=\frac{\inf_{\mathcal{S}} R_n(\mathcal{S})}{\mathbf{E}[L_n(\mathcal{S}_\delta)]}\leq 1$$

where $\mathcal{S}_\delta$ is the model selected by δ. δ is said to be asymptotically mean efficient if $\liminf_{n\to\infty}\mathrm{eff}(\delta)=1$.

By minimizing the loss rank in α we have shown in the previous paragraph that LoRP satisfies the first goal of model selection. We now show that with a suitable choice of α, LoRP also satisfies the second goal.

From (11), we have

$$\mathrm{LR}^{\alpha}_{\mathcal{S}}(\boldsymbol{y}|\boldsymbol{x})=\tfrac{n}{2}\log(\hat{\sigma}^2_{\mathcal{S}}+\tfrac{\alpha}{n}\boldsymbol{y}^\top\boldsymbol{y})+\tfrac{n}{2}\log n-\tfrac{|\mathcal{S}|}{2}\log(\alpha)-\tfrac{n-|\mathcal{S}|}{2}\log(1+\alpha).$$

By choosing $\alpha=\tilde{\alpha}=\exp(-\frac{n(n+|\mathcal{S}|)}{|\mathcal{S}|(n-|\mathcal{S}|-2)})$, under Assumption (A), the loss rank of model $\mathcal{S}$ (neglecting the common constant $\frac{n}{2}\log n$) is proportional to

$$\mathrm{LR}^{\tilde{\alpha}}_{\mathcal{S}}(\boldsymbol{y}|\boldsymbol{x})=n\log\hat{\sigma}^2_{\mathcal{S}}+\tfrac{n(n+|\mathcal{S}|)}{n-|\mathcal{S}|-2}+o_{\mathrm{P}}(1),$$

which is the corrected AIC of [HT89]. As a result, LoRP($\tilde{\alpha}$) is optimal in terms of regression estimation, i.e., it is asymptotically mean efficient ([Shi83], 1983; [Sha97], 1997).

Theorem 14 (Asymptotic mean efficiency) *Under Assumption (A) or (A'), with a suitable choice of α, the loss rank is proportional to the corrected AIC. As a result, LoRP is asymptotically mean efficient.*

5 Experiments

In this section we present a simulation study for LoRP, compare it to other methods and also demonstrate how LoRP can be used for some specific problems like choosing tuning parameters for kNN and spline regression. All experiments are conducted by using MATLAB software and the source code is freely available at http://www.hutter1.net/ai/lorpcode.zip.

Comparison to AIC and BIC for model identification. Samples are generated from the model

$$y = \beta_0 + \beta_1 X_1 + ... + \beta_d X_d + \epsilon, \;\; \epsilon \sim N(0, \sigma^2) \tag{17}$$

where $\boldsymbol{\beta}$ is the vector of coefficients with some zero entries. Without loss of generality, we assume that $\beta_0 = 0$, otherwise, we can center the response vector $\boldsymbol{y}$ and standardize the design matrix X to exclude β_0 from the model. We shall compare the performance of LoRP to that of BIC and AIC with various factors n, d and signal-to-noise ratio (SNR) which is $\|\boldsymbol{\beta}\|^2/\sigma^2$ ($\|\boldsymbol{\beta}\|^2$ is often called the length of the signal).

For a given set of factors (n, d, SNR), the way we simulate a dataset from model (17) is as follows. Entries of X are sampled from a uniform distribution on $[-1,1]$. To generate $\boldsymbol{\beta}$, we first create a vector $\boldsymbol{u} = (u_1,...,u_d)^\top$ whose entries are sampled from a uniform distribution on $[-1,1]$. The number of true covariates d^* is randomly selected from $\{1,2,...,d\}$, the last $d-d^*$ entries of $\boldsymbol{u}$ are set to zero, then coefficient vector $\boldsymbol{\beta}$ is computed by $\beta_i = \{\text{length of signal}\} * u_i/||\boldsymbol{u}||$. In our simulation, the length of signal was fixed to be 10. n observation errors $\epsilon_1,...,\epsilon_n$ are sampled from a normal distribution with mean 0 and variance $\sigma^2 = ||\boldsymbol{\beta}||^2/\text{SNR}$. Finally, the response vector is computed by $\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$. For each set of factors (n, d, SNR), 1000 datasets are simulated in the same manner to assess the average performance of the methods. For simplicity, a candidate model is specified by its order, i.e., we search the best model among only d models $\{1\},\{1,2\}...,\{1,2,...,d\}$. For the general case, an efficient branch-and-bound algorithm [Mil02, Chp.3] can be used to exhaustively search for the best subsets.

Table 1 presents percentages of correctly-fitted models with various factors n, d and SNR. As shown, LoRP outperforms the others. The better performance of LoRP over BIC, which is the most popular criterion for model identification, is very encouraging. This is probably because LoRP is a selection criterion with a data-dependent penalty. This improvement needs a theoretical justification which we intend to do in the future.

Table 1: Percentage of correctly-fitted models over 1000 replications

n	d	SNR	AIC	BIC	LoRP	n	d	SNR	AIC	BIC	LoRP
100	5	1	62	62	69	300	5	1	74	82	83
		5	85	85	86			5	78	90	91
		10	80	90	91			10	81	94	94
	10	1	52	42	54		10	1	63	67	71
		5	63	77	77			5	70	85	86
		10	68	84	85			10	74	90	90
	20	1	32	22	36		20	1	54	45	61
		5	55	63	65			5	64	79	80
		10	56	73	74			10	67	85	85

Comparison to AIC and BIC for regression estimation. Consider the following model which is from [Shi83]

$$y = y(x) = \log \tfrac{1}{1-x} + \epsilon, \;\; \epsilon \sim N(0, \sigma^2), \; x \in [0, 1). \tag{18}$$

We approximate the true function by a Fourier series and consider the problem of choosing a good order among models

$$y = \beta_0 + \sum_{l=1}^{k-1} \tfrac{\cos(\pi l x/\delta)}{l+1} \beta_l + \epsilon, \;\; k = 1, ..., K.$$

In the present context, a model in Section 4 is completely specified by the order K of the Fourier series. Samples are created from (18) at the points $x_i = \delta \frac{i}{n+1}$, $i = 1,...,n$. As in [Shi83], we take $\delta = .99$, and $K = 163$ with various n and σ. The performance is measured by the estimate of mean efficiency over 1000 replications.

Table 2 represents the simulation results. In general, LoRP (with $\alpha = \tilde{\alpha}$ as in Section 4) outperforms the others, except for cases with unrealistically high noise level. For cases with high noise, mean efficiency of BIC is often larger than that of AIC and LoRP. This was also shown in the simulation study of [Shi83], Table 1. This phenomenon can be explained as follows.

The risk of model k (the model specified by its order k) is $R_n(k) = \|(I - M_k)\boldsymbol{y}_{\text{true}}\|^2 + k\sigma^2$ where M_k is the regression matrix under model k and $\boldsymbol{y}_{\text{true}}$ is the vector of true values $y(x_i)$. When $\sigma \to \infty$, the ideal $k^\star = \operatorname{arginf}_k R_n(k) \to 1$. Because BIC penalizes the model complexity more strongly than AIC and LoRP do, the order chosen by BIC is closer to $k^\star = 1$ than the ones chosen by AIC and LoRP. As a result, mean efficiency of BIC is larger than that of the others.

Table 2: Estimates of mean efficiency over 1000 replications

n	σ	AIC	BIC	LoRP	n	σ	AIC	BIC	LoRP
400	.001	1.00	.98	.99	600	.001	1.00	.98	1.00
	.01	.93	.68	.90		.01	.99	.67	.92
	.05	.88	.67	.95		.05	.90	.66	.94
	.1	.88	.67	.92		.1	.90	.67	.93
	.5	.81	.66	.85		.5	.82	.66	.83
	1	.79	.63	.82		1	.79	.65	.82
	5	.67	.65	.70		5	.65	.67	.66
	10	.54	.67	.59		10	.54	.59	.54
	100	.31	.89	.33		100	.40	.90	.41

LoRP for selecting a good number of neighbors in kNN. Let us now see how LoRP can be applied to select a good parameter k in kNN regression.

We created a dataset of $n = 100$ observations (x_i, y_i) from the model:

$$y = f(x) + \varepsilon, \text{ with } f(x) = \tfrac{\sin(12(x+0.2))}{x+0.2}, \; x \in [0, 1] \tag{19}$$

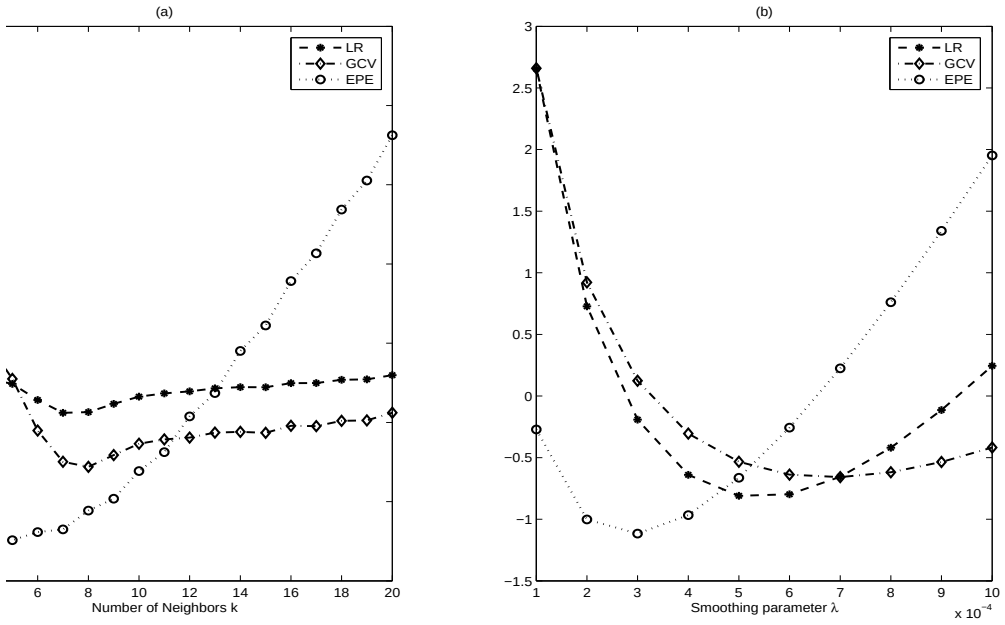

Figure 1: Choosing the tuning parameters in kNN and spline regression. The curves have been scaled by their standard deviations.

where $\varepsilon \sim N(0,\sigma^2)$ with $\sigma = 0.5$. The regression matrix $M^{(k)}$ for kNN regression is determined by $M_{ij}^{(k)} = \frac{1}{k}$ if $j \in \mathcal{N}_k(x_i)$ and 0 else. Then, the loss rank is

$$\mathrm{LR}(k) = \inf_{\alpha \geq 0} \{ \tfrac{n}{2} \log(\boldsymbol{y}^\top S_\alpha^{(k)} \boldsymbol{y}) - \tfrac{1}{2} \log\det S_\alpha^{(k)} \},$$

where $S_\alpha^{(k)} = (I - M^{(k)})^\top (I - M^{(k)}) + \alpha I$. The most widely-used method to select a good k is probably Generalized Cross-Validation (GCV) [CW79]: $\mathrm{GCV}(k) = n\|(I - M^{(k)})\boldsymbol{y}\|^2 / [\mathrm{tr}(I - M^{(k)})]^2$. To judge how well GCV and LoRP work, we compare them to the expected prediction error defined as

$$\mathrm{EPE}(k) \;=\; \sum_{i=1}^{n} \mathbf{E}(y_i - \hat{y}_i)^2 \;=\; \sum_{i=1}^{n} \Big[\sigma^2 + (f(x_i) - \tfrac{1}{k} \sum_{j \in \mathcal{N}_k(x_i)} f(x_j))^2 + \tfrac{\sigma^2}{k} \Big].$$

Figure 1(a) shows the curves $\mathrm{LR}(k)$, $\mathrm{GCV}(k)$, $\mathrm{EPE}(k)$ for $k = 2,...,20$ (the trivial case $k=1$ is omitted), in which $k=7$-nearest neighbors is chosen by LoRP and $k=8$ is chosen by GCV. The "ideal" k is 5. Both LoRP and GCV do a reasonable job. LoRP works slightly better than GCV.

LoRP for selecting a good smoothing parameter. We now further demonstrate the use of LoRP in selecting a good smoothing parameter for spline regression. Consider the following problem: find a function belonging to the class of functions with continuous 2nd derivative that minimizes the following penalized residual sum

of squares:

$$\mathrm{RSS}(f) = \sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt,$$

where λ is called the smoothing parameter. The second term penalizes the curvature of function f and the smoothing parameter λ controls the amount of penalty. Our goal is to choose a good λ.

It is well known (see, e.g., [HTF01], Section 5.4) that the solution is a natural spline $f(x) = \sum_{j=1}^{n} N_j(x)\theta_j$ where $N_1(x)$,...,$N_n(x)$ are the basis functions of the natural cubic spline:

$$N_1(x) = 1,\ N_2(x) = x,\ N_{k+2}(x) = d_k(x) - d_{n-1}(x) \ \text{ with } \ d_k(x) = \tfrac{(x-x_k)_+^3 - (x-x_n)_+^3}{x_n - x_k}.$$

The problem thus reduces to finding a vector $\boldsymbol{\theta} \in \mathbb{R}^n$ that minimizes

$$\mathrm{RSS}(\boldsymbol{\theta}) = (\boldsymbol{y} - N\boldsymbol{\theta})^\top (\boldsymbol{y} - N\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}^\top \Omega \boldsymbol{\theta}$$

where $N_{ij} = N_j(x_i)$ and $\Omega_{ij} = \int N_i''(x) N_j''(x) dx$. It is easy to see that the solution is $\hat{\boldsymbol{\theta}}_\lambda = (N^\top N + \lambda\Omega)^{-1} N^\top \boldsymbol{y}$, and the fitted vector is $\hat{\boldsymbol{y}} = N\hat{\boldsymbol{\theta}}_\lambda = M_\lambda \boldsymbol{y}$ with $M_\lambda = N(N^\top N + \lambda\Omega)^{-1} N^\top \boldsymbol{y}$. The fitted vector is linear in $\boldsymbol{y}$, thus the loss rank is

$$\mathrm{LR}(\lambda) = \arg\min_{\alpha \geq 0} \{ \tfrac{n}{2} \log(\boldsymbol{y}^\top S_\lambda^\alpha \boldsymbol{y}) - \tfrac{1}{2} \log\det S_\lambda^\alpha \}$$

where $S_\lambda^\alpha = (I - M_\lambda)^\top (I - M_\lambda) + \alpha I$.

Let us consider again the dataset generated from model (19). Figure 1(b) shows the curves $\mathrm{LR}(\lambda)$, $\mathrm{GCV}(\lambda)$ and $\mathrm{EPE}(\lambda)$. The derivation of expressions for $\mathrm{GCV}(\lambda)$ and $\mathrm{EPE}(\lambda)$ is similar to the previous example. $\lambda \approx 3 \times 10^{-4}$ is the optimal value selected by the "ideal" criterion EPE. $\lambda \approx 5 \times 10^{-4}$ and $\lambda \approx 7 \times 10^{-4}$ are selected by LoRP and GCV, respectively. One again, like the previous example, LoRP selects a better λ than GCV does.

6 Comparison to Gaussian Bayesian Linear Regression

We now consider LBFR from a Bayesian perspective with Gaussian noise and prior, and compare it to LoRP. In addition to the noise model as in PML, one also has to specify a prior. Bayesian model selection (BMS) proceeds by selecting the model that has largest evidence. In the special case of LBFR with Gaussian noise and prior and a type II maximum likelihood estimate for the noise variance, the expression for the evidence has a similar structure as the expression of the loss rank.

Gaussian Bayesian LBFR / MAP. Recall from Sec.3 Ex.9 that $\mathcal{F}_d$ is the class of functions $f_{\boldsymbol{w}}(x) = \boldsymbol{w}^\top \boldsymbol{\phi}(x)$ ($\boldsymbol{w} \in \mathbb{R}^d$) that are linear in feature vector $\boldsymbol{\phi}$. Let

$$\mathrm{Gauss}_N(\boldsymbol{z}|\boldsymbol{\mu}, \sigma) \ := \ \frac{\exp(-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu})^\top \sigma^{-1} (\boldsymbol{z} - \boldsymbol{\mu}))}{(2\pi)^{N/2} \sqrt{\det \sigma}} \tag{20}$$

denote a general N-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix σ. We assume that observations y are perturbed from $f_{\boldsymbol{w}}(x)$ by independent additive Gaussian noise with variance β^{-1} and zero mean, i.e., the likelihood of $\boldsymbol{y}$ under model $\boldsymbol{w}$ is $\mathrm{P}(\boldsymbol{y}|\boldsymbol{w}) = \mathrm{Gauss}_n(\boldsymbol{y}|\Phi\boldsymbol{w},\beta^{-1}I)$, where $\Phi_{ia} = \boldsymbol{\phi}_a(x_i)$. A Bayesian assumes a prior (before seeing $\boldsymbol{y}$) distribution on $\boldsymbol{w}$. We assume a centered Gaussian with covariance matrix $(\alpha C)^{-1}$, i.e., $\mathrm{P}(\boldsymbol{w}) = \mathrm{Gauss}_d(\boldsymbol{w}|\boldsymbol{0},\alpha^{-1}C^{-1})$. From the prior and the likelihood one can compute the evidence and the posterior

$$\begin{aligned} \text{Evidence:} \qquad \mathrm{P}(\boldsymbol{y}) &= \int \mathrm{P}(\boldsymbol{y}|\boldsymbol{w})\mathrm{P}(\boldsymbol{w})d\boldsymbol{w} = \mathrm{Gauss}_n(\boldsymbol{y}|\boldsymbol{0},\beta^{-1}S^{-1}) \quad (21) \\ \text{Posterior:} \qquad \mathrm{P}(\boldsymbol{w}|\boldsymbol{y}) &= \mathrm{P}(\boldsymbol{y}|\boldsymbol{w})\mathrm{P}(\boldsymbol{w})/P(\boldsymbol{y}) = \mathrm{Gauss}_d(\boldsymbol{w}|\hat{\boldsymbol{w}},A^{-1}) \end{aligned}$$

$$B := \Phi^\top\Phi, \quad A := \alpha C + \beta B, \quad M := \beta\Phi A^{-1}\Phi^\top, \quad S := I - M, \qquad (22)$$
$$\hat{\boldsymbol{w}} := \beta A^{-1}\Phi^\top\boldsymbol{y}, \quad \hat{\boldsymbol{y}} := \Phi\hat{\boldsymbol{w}} = M\boldsymbol{y}$$

A standard Bayesian point estimate for $\boldsymbol{w}$ for fixed d is the one that maximizes the posterior (MAP) (which in the Gaussian case coincides with the mean) $\hat{\boldsymbol{w}} = \mathrm{argmax}_{\boldsymbol{w}}\mathrm{P}(\boldsymbol{w}|\boldsymbol{y}) = \beta A^{-1}\Phi^\top\boldsymbol{y}$. For $\alpha \to 0$, MAP reduces to Maximum Likelihood (ML), which in the Gaussian case coincides with the least squares regression of Ex.9. For $\alpha > 0$, the regression matrix M is not a projection anymore.

Bayesian model selection. Consider now a family of models $\{\mathcal{F}_1,\mathcal{F}_2,...\}$. Here the $\mathcal{F}_d$ are the linear regressors with d basis functions, but in general they could be completely different model classes. All quantities in the previous paragraph implicitly depend on the choice of $\mathcal{F}$, which we now explicate with an index. In particular, the evidence for model class $\mathcal{F}$ is $\mathrm{P}_{\mathcal{F}}(\boldsymbol{y})$. BMS chooses the model class (here d) $\mathcal{F}$ of highest evidence:

$$\mathcal{F}^{\mathrm{BMS}} = \arg\max_{\mathcal{F}} \mathrm{P}_{\mathcal{F}}(\boldsymbol{y})$$

Once the model class $\mathcal{F}^{\mathrm{BMS}}$ is determined, the MAP (or other) regression function $f_{\boldsymbol{w}_{\mathcal{F}^{\mathrm{BMS}}}}$ or $M_{\mathcal{F}^{\mathrm{BMS}}}$ are chosen. The data variance β^{-1} may be known or estimated from the data, C is often chosen I, and α has to be chosen somehow. Note that while $\alpha \to 0$ leads to a reasonable MAP=ML regressor for fixed d, this limit cannot be used for BMS.

Comparison to LoRP. Inserting (20) into (21) and taking the logarithm we see that BMS minimizes

$$-\log \mathrm{P}_{\mathcal{F}}(\boldsymbol{y}) = \tfrac{\beta}{2}\boldsymbol{y}^\top S\boldsymbol{y} - \tfrac{1}{2}\log\det S - \tfrac{n}{2}\log\tfrac{\beta}{2\pi} \qquad (23)$$

w.r.t. $\mathcal{F}$. Let us estimate β by ML: We assume a broad prior $\alpha \ll \beta$ so that $\beta\frac{\partial S}{\partial\beta} = O(\frac{\alpha}{\beta})$ can be neglected. Then $-\frac{\partial\log \mathrm{P}_{\mathcal{F}}(\boldsymbol{y})}{\partial\beta} = \frac{1}{2}\boldsymbol{y}^\top S\boldsymbol{y} - \frac{n}{2\beta} + O(\frac{\alpha}{\beta}n) = 0 \Leftrightarrow \beta \approx \hat{\beta} := n/(\boldsymbol{y}^\top S\boldsymbol{y})$. Inserting $\hat{\beta}$ into (23) we get

$$-\log \mathrm{P}_{\mathcal{F}}(\boldsymbol{y}) = \tfrac{n}{2}\log \boldsymbol{y}^\top S\boldsymbol{y} - \tfrac{1}{2}\log\det S - \tfrac{n}{2}\log\tfrac{n}{2\pi\mathrm{e}} \qquad (24)$$

Taking an improper prior $\mathrm{P}(\beta) \propto \beta^{-1}$ and integrating out β leads for small α to a similar result. The last term in (24) is a constant independent of $\mathcal{F}$ and can be ignored. The first two terms have the same structure as in linear LoRP (10), but the matrix S is different. In both cases, α act as regularizers, so we may minimize over α in BMS like in LoRP. For $\alpha = 0$ (which neither makes sense in BMS nor in LoRP), M in BMS coincides with M of Ex.9, but still the S_0 in LoRP is the square of the S in BMS. For $\alpha > 0$, M of BMS may be regarded as a regularized regressor as suggested in Sec.2 (a), rather than a regularized loss function (b) used in LoRP. Note also that BMS is limited to (semi)parametric regression, i.e., does not cover the non-parametric kNN Ex.2 and kernel Ex.8, unlike LoRP.

Since B only depends on $\boldsymbol{x}$ (and not on $\boldsymbol{y}$), and all P are implicitly conditioned on $\boldsymbol{x}$, one could choose $C = B$. In this case, $M = \gamma \Phi B^{-1} \Phi^\top$, with $\gamma = \frac{\beta}{\alpha+\beta} < 1$ for $\alpha > 0$, is a simple multiplicative regularization of projection $\Phi B^{-1} \Phi^\top$, and (24) coincides with (11) for suitable α, apart from an irrelevant additive constant, hence minimizing (24) over α also leads to (12).

7 Comparison to other Model Selection Schemes

In this section we give a brief introduction to PML for (semi)parametric regression, and its major instantiations, AIC, BIC, and MDL principle, whose penalty terms are all proportional to the number of parameters d. The *effective number of parameters* is often much smaller than d, e.g., if there are soft constraints like in ridge regression. We compare MacKay's trace formula [Mac92] for Gaussian Bayesian LBFR and Hastie's et al. trace formula [HTF01] for general linear regression with LoRP.

Penalized ML (AIC, BIC, MDL). Consider a d-dimensional stochastic model class like the Gaussian Bayesian linear regression example of Section 6. Let $\mathrm{P}_d(\boldsymbol{y}|\boldsymbol{w})$ be the data likelihood under d-dimensional model $\boldsymbol{w} \in \mathbb{R}^d$. The maximum likelihood (ML) estimator for fixed d is

$$\hat{\boldsymbol{w}} \;=\; \arg\max_{\boldsymbol{w}} \mathrm{P}_d(\boldsymbol{y}|\boldsymbol{w}) \;=\; \arg\min_{\boldsymbol{w}} \{-\log \mathrm{P}_d(\boldsymbol{y}|\boldsymbol{w})\} \tag{25}$$

Since $-\log \mathrm{P}_d(\boldsymbol{y}|\boldsymbol{w})$ decreases with d, we cannot find the model dimension by simply minimizing over d (overfitting). Penalized ML adds a complexity term to get reasonable results

$$\hat{d} \;=\; \arg\min_{d} \{-\log \mathrm{P}_d(\boldsymbol{y}|\hat{\boldsymbol{w}}) + \mathrm{Penalty}(d)\} \tag{26}$$

The penalty introduces a tradeoff between the first and second term with a minimum at $\hat{d} < \infty$. Various penalties have been suggested: AIC [Aka73] uses d, BIC [Sch78] and the (crude) MDL [Ris78, Grü04] use $\frac{d}{2}\log n$ for $\mathrm{Penalty}(d)$. There are at least *three important conceptual differences* to LoRP:

- In order to apply PML one needs to specify not only a class of regression functions, but a full probabilistic model $\mathrm{P}_d(\boldsymbol{y}|\boldsymbol{w})$,

- PML ignores or at least does not tell how to incorporate a potentially given loss-function,
- PML is mostly limited to selecting between (semi)parametric models.

We discuss two approaches to the last item in the remainder of this section (where AIC, BIC, and MDL are not directly applicable): (a) for non-parametric models like kNN or kernel regression, or (b) if d does not reflect the "true" complexity of the model. [Mac92] suggests an expression for the effective number of parameters d_{eff} as a substitute for d in case of (b), while [HTF01] introduce another expression which is applicable for both (a) and (b).

The trace penalty for parametric Gaussian LBFR. We continue with the Gaussian Bayesian linear regression example (see Section 6 for details and notation). Performing the integration in (21), [Mac92, Eq.(21)] derives the following expression for the Bayesian evidence for $C=I$

$$\begin{aligned} -\log \mathrm{P}(\boldsymbol{y}) &= (\alpha\hat{E}_W + \beta\hat{E}_D) + (\tfrac{1}{2}\log\det A - \tfrac{d}{2}\log\alpha) - \tfrac{n}{2}\log\tfrac{\beta}{2\pi} \qquad (27) \\ \hat{E}_D &= \tfrac{1}{2}\|\Phi\hat{\boldsymbol{w}} - \boldsymbol{y}\|_2^2, \quad \hat{E}_W = \tfrac{1}{2}\|\hat{\boldsymbol{w}}\|_2^2 \end{aligned}$$

(the first bracket in (27) equals $\frac{\beta}{2}\boldsymbol{y}^\top S\boldsymbol{y}$ and the second equals $-\frac{1}{2}\mathrm{logdet}S$, cf. (23)). Minimizing (27) w.r.t. α leads to the following relation:

$$0 \;=\; \tfrac{-\partial\log \mathrm{P}(\boldsymbol{y})}{\partial\alpha} \;=\; \hat{E}_W + \tfrac{1}{2}\mathrm{tr}A^{-1} - \tfrac{d}{2\alpha} \qquad (\tfrac{\partial}{\partial\alpha}\log\det A = \mathrm{tr}A^{-1})$$

He argues that $\alpha\|\hat{\boldsymbol{w}}\|_2^2$ corresponds to the effective number of parameters, hence

$$d_{eff}^{\mathrm{McK}} \;:=\; \alpha\|\hat{\boldsymbol{w}}\|_2^2 \;=\; 2\alpha\hat{E}_W \;=\; d - \alpha\mathrm{tr}A^{-1} \qquad (28)$$

The trace penalty for general linear models. We now return to general linear regression $\hat{\boldsymbol{y}} = M(\boldsymbol{x})\boldsymbol{y}$ (7). LBFR is a special case of a projection matrix $M=M^2$ with rank $d=\mathrm{tr}M$ being the number of basis functions. M leaves d directions untouched and projects all other $n-d$ directions to zero. For general M, [HTF01, Sec.5.4.1] argue to regard a direction that is only somewhat shrunken, say by a factor of $0<\beta<1$, as a fractional parameter (β degrees of freedom). If β_1,...,β_n are the shrinkages = eigenvalues of M, the effective number of parameters could be defined as [HTF01, Sec.7.6]

$$d_{eff}^{\mathrm{HTF}} \;:=\; \sum_{i=1}^{n}\beta_i \;=\; \mathrm{tr}M,$$

where HTF stands for Hastie-Tibshirani-Friedman, which generalizes the relation $d=\mathrm{tr}M$ beyond projections. For MacKay's M (22), $\mathrm{tr}M = d-\alpha\mathrm{tr}A^{-1}$, i.e., d_{eff}^{HTF} is consistent with and generalizes d_{eff}^{McK}.

Problems. Though nicely motivated, the trace formula is not without problems. First, since for projections, $M=M^2$, one could have argued equally well for $d_{eff}^{\mathrm{HTF}} =$

$\mathrm{tr}M^2$. Second, for kNN we have $\mathrm{tr}M=\frac{n}{k}$ (since M is $\frac{1}{k}$ on the diagonal), which does not look unreasonable. Consider now kNN', which is defined as follows: we average over the k nearest neighbors *excluding* the closest neighbor. For sufficiently smooth functions, kNN' for suitable k is still a reasonable regressor, but $\mathrm{tr}M=0$ (since M is zero on the diagonal). So $d_{eff}^{\mathrm{HTF}}=0$ for kNN', which makes no sense and would lead one to always select the $k=1$ model.

Relation to LoRP. In the case of kNN', $\mathrm{tr}M^2$ would be a better estimate for the effective dimension. In linear LoRP, $-\mathrm{logdet}S_\alpha$ serves as complexity penalty. Ignoring the nullspace of $S_0=(I-M)^\top(I-M)$ (8), we can Taylor expand $-\frac{1}{2}\mathrm{logdet}S_0$ in M

$$-\tfrac{1}{2}\log\det S_0 \;=\; -\mathrm{tr}\log(I-M) \;=\; \sum_{s=1}^{\infty}\tfrac{1}{s}\mathrm{tr}(M^s) \;=\; \mathrm{tr}M+\tfrac{1}{2}\mathrm{tr}M^2+...$$

For BMS (24) with $S=I-M$ (22) we get half of this value. So the trace penalty may be regarded as a leading order approximation to LoRP. The higher order terms prevent peculiarities like in kNN'.

Coding/MDL interpretation of LoRP. The basic idea of MDL is as follows [Grü04]: "The goal of statistical inferences may be cast as trying to find regularity in the data. 'Regularity' may be identified with 'ability to compress'. MDL combines these two insights by *viewing learning as data compression*: it tells us that, for a given set of hypotheses $\mathcal{H}$ and data set D, we should try to find the hypothesis or combination of hypotheses in $\mathcal{H}$ that compress D most."

The standard incarnation of (crude) MDL is as follows: If H is a stochastic model of (discrete) data D, we can code D (by Shannon-Fano) in $\lceil-\log_2\mathrm{P}(D|H)\rceil$ bits. If we have a class of models $\mathcal{H}$, we also have to code H (somehow in, say, $L(H)$ bits) in order to be able to decode D. MDL chooses the hypothesis $H^{\mathrm{MDL}}=\mathrm{argmin}_{H\in\mathcal{H}}\{-\log_2\mathrm{P}(D|H)+L(H)\}$ of minimal two-part code. For instance, if $\mathcal{H}$ is the class of all polynomials of all degrees with each coefficient coded to $\frac{1}{2}\log_2 n$ bits (i.e., $O(n^{-1/2})$ accuracy) and we condition on x, i.e., $D\rightsquigarrow\boldsymbol{y}|\boldsymbol{x}$, MDL takes the form (25) and (26), i.e., $H^{\mathrm{MDL}}=(\hat{\boldsymbol{w}},\hat{d})$.

We now give LoRP (for discrete D) a data compression/MDL interpretation. For simplicity, we will first assume that all loss values are different, i.e., if $\mathrm{Loss}_r(\boldsymbol{y}'|\boldsymbol{x})\neq\mathrm{Loss}_r(\boldsymbol{y}''|\boldsymbol{x})$ for $\boldsymbol{y}'\neq\boldsymbol{y}''$ (adding infinitesimal random noise to Loss_r easily ensures this). In this case, $\mathrm{Rank}_r(\cdot|\boldsymbol{x}):\mathcal{Y}^n\to\mathbb{N}$ is an order preserving bijection, i.e., $\mathrm{Rank}_r(\boldsymbol{y}'|\boldsymbol{x})<\mathrm{Rank}_r(\boldsymbol{y}''|\boldsymbol{x})$ iff $\mathrm{Loss}_r(\boldsymbol{y}'|\boldsymbol{x})<\mathrm{Loss}_r(\boldsymbol{y}''|\boldsymbol{x})$ with no gaps in the range of $\mathrm{Rank}_r(\cdot|\boldsymbol{x})$.

Phrased differently, $\mathrm{Rank}_r(\cdot|\boldsymbol{x})$ codes each $\boldsymbol{y}'\in\mathcal{Y}^n$ as a natural number m in increasing loss-order. The natural number m can itself be coded in $\lceil\log_2 m\rceil$ bits (using plain not prefix coding). Let us call this code of $\boldsymbol{y}'$ the *Loss Rank Code* (LRC). LRC has a nice characterization: LRC is the shortest loss-order preserving code. Ignoring the rounding, the *Length* of $\mathrm{LRC}_r(\boldsymbol{y}'|\boldsymbol{x})$ is $\mathrm{LR}_r(\boldsymbol{y}'|\boldsymbol{x})$:

Proposition 15 (Minimality property) *If all loss values are different, i.e., if*

$$\text{Loss}_r(\boldsymbol{y}'|\boldsymbol{x}) \neq \text{Loss}_r(\boldsymbol{y}''|\boldsymbol{x}) \text{ for all } \boldsymbol{y}' \neq \boldsymbol{y}''$$

then the loss rank (code) of $\boldsymbol{y}$ is the smallest/shortest among all loss-order preserving rankings/codes C in the sense that

$$\begin{aligned} \text{Rank}(\boldsymbol{y}) &= \min\{C(\boldsymbol{y}) : C \in \mathcal{Y}^n \to \mathbb{N} \wedge (\star)\} \\ \lfloor \text{LR}(\boldsymbol{y})/\log 2 \rfloor &= \min\{\text{Length}(C(\boldsymbol{y})) : C \in \mathcal{Y}^n \to \{0,1\}^* \wedge (\star)\} \\ (\star) &:= [\text{Loss}(\boldsymbol{y}') < \text{Loss}(\boldsymbol{y}'') \Leftrightarrow C(\boldsymbol{y}') < C(\boldsymbol{y}''), \forall \boldsymbol{y}', \boldsymbol{y}''] \end{aligned}$$

The proof follows from the fact that if a discrete injection (code) is order preserving, there exists a "smallest" one without gaps in the range. So LoRP minimizes the Loss Rank Code, where LRC itself is the shortest among all loss-order preserving codes. From this perspective, LoRP is just a different (non-stochastic, non-parametric, loss-based) incarnation of MDL. The MDL philosophy provides a justification of LoRP (2), its regularization (5), and loss function selection (Section 8). This identification should also allow to apply or adapt the various consistency results of MDL, implying that LoRP is consistent under some mild conditions.

If some losses are equal, $\text{Rank}_r(\cdot|\boldsymbol{x}):\mathcal{Y}^n \to \mathbb{N}$ still preserves the order $\leq$, but the mapping is neither surjective nor injective anymore.

Large regression classes $\mathcal{R}$. The classes $\mathcal{R}$ of regressors we considered so far were discrete and "small", often indexed by an integer complexity index (like k in kNN or d in LBFR). But large classes are also thinkable.

As an extreme case, consider the class of *all* regressors. Clearly, there is an $r=r_D$ which "knows" D and perfectly fits D ($r(x_i|D)=y_i$, $\forall i$), but is the worst possible on all other D' ($r(x_i|D')=\infty$, $\forall i$, $\forall D' \neq D$). This r has (discrete) Rank 1, so is best according to LoRP. So if $\mathcal{R}$ is too large, LoRP can overfit too.

Consider a more realistic example by not taking *all* of the first d basis functions in LBFR, but selecting *some* basis functions $\phi_{i_1},...,\phi_{i_d}$, i.e., $\mathcal{R}$ is indexed by d integers, and d may be variable too.

One solution approach is to group more regressors in $\mathcal{R}$ into one function class $\mathcal{F}$, e.g., the class of functions $\mathcal{F}_{k,d}=\{w_1\phi_{i_1}+...w_d\phi_{i_d}:\boldsymbol{w}\in\mathbb{R}^d, 1\leq i_1<...<i_d\leq k\}$ that are linear in d of the first k bases. Now $\mathcal{R}$ is a small class indexed by d and k only.

Looking at the coding interpretation of LR_r and the MDL philosophy, suggests to assign a code to $r\in\mathbb{R}$ in order to get a complete code for D:

$$r^{best} = \arg\min_r\{\text{LR}_r(\boldsymbol{y}|\boldsymbol{x}) + L(r)\}$$

where r is the length of a code for r (given $\mathcal{R}$). For $\mathcal{R} \simeq \mathbb{N}$ a single integer has to be coded, e.g., k in $L(r)=L(k)\approx\log_2 k$ bits, which can usually be safely dropped/ignored. For more complex classes like the (ungrouped) LBFR subset selection above, $L(r)=L(i_1,...,i_d,d)\approx d\log_2 k+\log_2 d$ can become important.

8 Loss Functions and their Selection

General additive loss. Linear LoRP $\hat{\boldsymbol{y}}=M(\boldsymbol{x})\boldsymbol{y}$ of Section 3 can easily be generalized to non-quadratic loss. Let us consider the $\rho>0$ loss

$$\begin{aligned}\text{Loss}_M(\boldsymbol{y}|\boldsymbol{x}) &:= (\textstyle\sum_{i=1}^n(y_i-\hat{y}_i)^\rho)^{1/\rho} = \|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_\rho = \|(I-M)\boldsymbol{y}\|_\rho \\ V(L) &= \{\boldsymbol{y}'\in\mathbb{R}^n : \|(I-M)\boldsymbol{y}'\|_\rho\leq L\} = \{(I-M)^{-1}\boldsymbol{z}\in\mathbb{R}^n : \|\boldsymbol{z}\|_\rho\leq L\} \\ \text{Let}\quad v_n^\rho &:= |\{\boldsymbol{z}\in\mathbb{R}^n : \|\boldsymbol{z}\|_\rho\leq 1\}| = 2^n\textstyle\prod_{i=1}^{n-1}\frac{i}{\rho}!\frac{1}{\rho}!/\frac{i+1}{\rho}!,\end{aligned}$$

where $\frac{i}{\rho}!:=\Gamma(\frac{i}{\rho}+1)$, be the volume of the unit d-dimensional ρ-norm "ball". Since $V(L)$ is a linear transformation of this ball with transformation matrix $(I-M)^{-1}$ and scaling L, we have $|V(L)|=v_n^\rho L^n/\det(I-M)$, hence

$$\text{LR}_M(\boldsymbol{y}|\boldsymbol{x}) = \log|V(\text{Loss}_M(\boldsymbol{y}|\boldsymbol{x}))| = n\log\|(I-M)\boldsymbol{y}\|_\rho - \log\det(I-M) + \log v_n^\rho \tag{29}$$

For the $\rho=2$ norm, (29) reduces to LR_M^0 (9). Note that $\text{Loss}_M:=g(\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_\rho)$ leads to the same result (29) for any monotone increasing g, i.e., only the order of the loss matters, not its absolute value. More generally $\text{Loss}_M=g(\sum_i h(y_i-\hat{y}_i))$ for any h implies

$$\begin{aligned}\text{LR}_M(\boldsymbol{y}|\boldsymbol{x}) &= n\log v_n^h(\textstyle\sum_i h(y_i-\hat{y}_i)) - \log\det(I-M), \quad \text{where} \\ v_n^h(l) &:= |\{\boldsymbol{z}\in\mathbb{R}^n : \textstyle\sum_i h(z_i)\leq l\}|^{1/n}\end{aligned}$$

is a one-dimensional function of l (independent D and M), once to be determined (e.g., $v_n^h(l)=l\cdot(v_n^\rho)^{1/n}\propto l$ for ρ-norm loss). Regularization may be performed by $M\rightsquigarrow\gamma M$ with optimization over $\gamma<1$.

Loss-function selection. In principle, the loss function should be part of the problem specification, since it characterizes the ultimate goal. For instance, whether a test should more likely classify a healthy person as sick than a sick person as healthy, depends on the severity of a misclassification (loss) in each direction. In reality, though, having to specify the loss function can be a nuisance. Sure, the loss has to respect some general features, e.g., that it increases with the deviation of $\hat{y}_i$ from y_i. Otherwise it is chosen by convenience or rules of thumb, rather than by elicitation of the real goal, for instance preferring the Euclidean norm over $\rho\neq 2$ norms. If we subscribe to the procedure of *choosing* the loss function, we could ask whether this may be done in a more principled way. Consider a (not too large) class of loss functions Loss^α, indexed by some parameter α. For instance, $\text{Loss}^\alpha=\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_\alpha$ from the previous paragraph. The regularized loss (5) also constitutes a class of losses. In this case we minimized over the regularization parameter α. This suggests to choose in general the loss function that has minimal loss rank LR_r^α. The justifications are similar to the ones for minimizing LR_r^α w.r.t. r. Note that the term $\log v_n^\rho$ cannot be dropped anymore, unlike in (10).

9 Self-Consistent Regression

So far we have considered only "on-data" regression. LoRP only depends on the regressor r on data D and not on $x \notin \{x_1,...,x_n\}$. We now construct canonical regressors for off-data x from regressors given only on-data. First, this may ease the specification of the regression functions, second, it is a canonical way for interpolation (LoRP can't distinguish between r that are identical on D), and third, we show that many standard regressors (kNN, Kernel, LBFR) are self-consistent in the sense that they are canonical. We limit our exposition to linear regression.

Off-data regression. A linear regressor is completely determined by the n functions m_j (6), but not by the matrix function M (7). Indeed, two sets $\{m_j\}$ and $\{m'_j\}$ that coincide on $D=(\boldsymbol{x},\boldsymbol{y})$, i.e. $m_j(x_i|\boldsymbol{x})=m'_j(x_i|\boldsymbol{x})\,\forall i,j$ but possibly differ for $x\notin\boldsymbol{x}$, lead to the same matrix $M_{ij}(\boldsymbol{x})=m_j(x_i|\boldsymbol{x})=m'_j(x_i|\boldsymbol{x})$. LoRP has the advantage of only depending on M, but this also means that it cannot distinguish between an m_j that behaves well on $x\notin\boldsymbol{x}$ and one that, e.g., wildly oscillates outside $\boldsymbol{x}$.

Typically, the m_j are given and, provided the model complexity is chosen appropriately e.g. by LoRP, they properly interpolate $\boldsymbol{x}$. Nevertheless, a canonical extension from M to m_j would be nice. In this way LoRP would not be vulnerable to bad m_j, and we could interpolate D (predict y for any $x\in\mathcal{X}$) even without m_j given a-priori.

We define a self-consistent regression scheme based only on M (for all n). We ask for an estimate $\hat{y}$ of y for $x\notin\boldsymbol{x}$. We add a virtual data point (x_0,y_0) to D, where $x_0=x$. If we knew $y_0=y$ we could estimate $\hat{y}_0=r(x_0|\{(x_0,y_0)\}\cup D)$, but we don't know y_0. But we could require a self-consistency condition, namely that $\hat{y}_0=y_0$ for $x_0\notin\boldsymbol{x}$.

Definition 16 (canonical and self-consistent regressors) *Let $M'_{ij}(\boldsymbol{x}')_{0\le i,j\le n}$ be the regression matrix for the data set $D'=\{(x_0,y_0)\}\cup D=((x_0,\boldsymbol{x}),(y_0,\boldsymbol{y}))=(\boldsymbol{x}',\boldsymbol{y}')$ of size $n+1$.*

(i) A linear regressor $\tilde{y}_0=\tilde{r}(x_0|D)$ is called a canonical regressor for M' if the consistency condition $\tilde{y}_0=r(x_0|D')\equiv\sum_{j=0}^n M'_{0j}y_j$ holds $\forall x_0,D$.
(ii) A regressor r is called self-consistent if $\tilde{r}=r$, i.e. if $r(x_0|\{(x_0,r(x_0|D))\}\cup D)=r(x_0|D)\ \forall x_0,D$.
(iii) A class of regressors $\mathcal{R}=\{r\}$ is called self-consistent if $\tilde{R}=\{\tilde{r}\}\subseteq\mathcal{R}$.

We denote the solution of the self-consistency condition $y_0=\sum_{j=0}^n M'_{0j}y_j$ by $\tilde{y}_0$. So we have to solve

$$\tilde{y}_0 \;=\; \sum_{j=1}^n M'_{0j}y_j + M'_{00}\tilde{y}_0 \quad\Longrightarrow\quad \tilde{y}_0 \;=\; \frac{\sum_{j=1}^n M'_{0j}y_j}{1-M'_{00}} \;=\; \frac{\sum_{j=1}^n M'_{0j}y_j}{\sum_{j=1}^n M'_{0j}}$$

where the last equality only holds if $\sum_{j=0}^n M'_{0j}=1$, which is often the case, in particular for kNN and Kernel regression, but not necessarily for LBFR.

Proposition 17 (canonical regressor) *The linear regressor*

$$y_0 \;=\; \tilde r(x_0|D) \;:=\; \sum_{j=1}^n \tilde m_j(x_0|\boldsymbol{x}) y_j, \quad \textit{where} \quad \tilde m_j(x_0|\boldsymbol{x}) := \frac{M'_{0j}(\boldsymbol{x}')}{1-M'_{00}(\boldsymbol{x}')}$$

is the unique canonical regressor for M' (if $M'_{00}<1$).

Example 18 (self-consistent kNN, ↑Ex.2) $M'_{0j}(\boldsymbol{x}')=\frac{1}{k}$ for $x_j\in\mathcal{N}'_k(x_0)$ and 0 else. The k nearest neighbors $\mathcal{N}'_k(x_0)$ of x_0 among $\boldsymbol{x}'$ consist of x_0 and the $k-1$ nearest neighbors $\mathcal{N}_{k-1}(x_0)=:J$ of x_0 among $\boldsymbol{x}$, i.e. $\mathcal{N}'_k(x_0)=\{x_0\}\cup\mathcal{N}_{k-1}(x_0)$. Hence

$$\tilde y_0 \;=\; \frac{\sum_{j=1}^n M'_{0j}y_j}{\sum_{j=1}^n M'_{0j}} \;=\; \frac{\sum_{j\in J}\frac{1}{k}y_j}{\sum_{j\in J}\frac{1}{k}} \;=\; \sum_{j\in J}\tfrac{1}{k-1}y_j \;=\; \sum_{j=1}^n M_{0j}^{(k-1)}y_j \;=\; r_{k-1}(x_0|D)=\hat y_0$$

Canonical kNN is equivalent to standard (k–1)NN, so the class of canonical kNN regressors coincides with the standard kNN class. ♢

Example 19 (self-consistent kernel)

$$M'_{0j}(\boldsymbol{x}') \;=\; \frac{K(x_0,x_j)}{\sum_{j=0}^n K(x_0,x_j)} \quad\Longrightarrow\quad \tilde y_0 \;=\; \frac{\sum_{j=1}^n K(x_0,x_j)y_j}{\sum_{j=1}^n K(x_0,x_j)} \;=\; r(x_0|D) \;=\; \hat y_0$$

Canonical kernel regression coincides with the standard kernel smoother. ♢

Example 20 (self-consistent LBFR)

$$\begin{aligned} B' \;&=\; \sum_{i=0}^n \boldsymbol{\phi}(x_i)\boldsymbol{\phi}(x_i)^\top = B+\boldsymbol{\phi}(x_0)\boldsymbol{\phi}(x_0)^\top \\ \Rightarrow\quad M'_{0j} \;&=\; \boldsymbol{\phi}(x_0)^\top B'^{-1}\boldsymbol{\phi}(x_j) \;=\; \boldsymbol{\phi}(x_0)^\top\left[B^{-1}-\frac{B^{-1}\boldsymbol{\phi}(x_0)\boldsymbol{\phi}(x_0)^\top B^{-1}}{1+\boldsymbol{\phi}(x_0)^\top B^{-1}\boldsymbol{\phi}(x_0)}\right]\boldsymbol{\phi}(x_j) \\ &=\; M_{0j}-\frac{M_{00}M_{0j}}{1+M_{00}} \;=\; \frac{M_{0j}}{1+M_{00}} \quad\Rightarrow\quad 1-M'_{00}=\frac{1}{1+M_{00}} \end{aligned}$$

In the first line we used the Sherman-Morrison formula for inverting B'. In the second line we defined $M_{0j}=\boldsymbol{\phi}(x_0)^\top B^{-1}\boldsymbol{\phi}(x_j)$, extending M.

$$\Rightarrow\quad \tilde y_0 \;=\; \frac{\sum_{j=1}^n M'_{0j}y_j}{1-M'_{00}} \;=\; \sum_{j=1}^n M_{0j}y_j \;=\; \sum_{j=1}^n m_j(x_0,\boldsymbol{x})y_j \;=\; \hat y_0$$

Canonical LBFR coincides with standard LBFR. ♢

Proposition 21 (self-consistent regressors) *Kernel regression and linear basis function regression are self-consistent. kNN is not self-consistent but the class of kNN regressors $\mathcal{R}=\{r_{kNN}:k\in\mathbb{N}\}$ is self-consistent.*

To summarize, we expect LoRP to select good regressors with proper interpolation behavior for canonical and self-consistent regressors.

10 Nearest Neighbors Classification

We now consider k-nearest neighbors classification in more detail. In order to get more insight into LoRP we seek a case that allows analytic solution. In general, the determinant $\det S_\alpha$ cannot be computed analytically, but for $\boldsymbol{x}$ lying on a hypercube of the regular grid $\mathcal{X}=\mathbb{Z}^d$ we can. We derive exact expressions, and consider the limits $n\to\infty$, $k\to\infty$, and $d\to\infty$.

kNN on one-dimensional grid. We consider the $d=1$ dimensional case first. We assume $\boldsymbol{x}=(1,2,3,...,n)$, a circular metric $d(x_i,x_j)=d(i,j)=\min\{|i-j|,n-|i-j|\}$, and odd $k\leq n$. The kNN regression matrix

$$M_{ij}=b_{i-j} \quad \text{with} \quad b_{i-j}=\tfrac{1}{k} \quad \text{if} \quad d(i,j)\leq\tfrac{k-1}{2} \quad \text{and} \quad 0 \quad \text{otherwise}$$

is a diagonal-constant (Toeplitz) matrix with circularity property $b_{i-j}=b_{i-j+n}$. For instance, for $k=3$ and $n=5$

$$M=\frac{1}{3}\begin{pmatrix}1&1&0&0&1\\1&1&1&0&0\\0&1&1&1&0\\0&0&1&1&1\\1&0&0&1&1\end{pmatrix}$$

For every circulant matrix, the eigenvectors $\boldsymbol{v}^1,...,\boldsymbol{v}^n$ are waves $v^l_j=\theta^{jl}$ with $\theta=\mathrm{e}^{2\pi\sqrt{-1}/n}$. The eigenvalues are the fourier transform $\hat{b}_l=\sum_{j=1}^n b_j\theta^{-jl}$ of $\boldsymbol{b}$, since $\sum_j M_{ij}v^l_j=\sum_j b_{i-j}\theta^{jl}=\sum_j b_j\theta^{(i-j)l}=v^l_i\sum_j b_j\theta^{-jl}=\hat{b}_l v^l_i$, where we exploited circularity of $\boldsymbol{b}$ and θ^{jl}. For M_{kNN} in particular we get

$$\underset{\text{circularity}}{\hat{b}_l \;=\; \frac{1}{k}\sum_{j=-\frac{k-1}{2}}^{\frac{k-1}{2}}\theta^{-jl}} \;=\; \underset{\text{geometric sum}}{\frac{1}{k}\frac{\theta^{lk/2}-\theta^{-lk/2}}{\theta^{l/2}-\theta^{-l/2}}} \;=\; \underset{\text{insert }\theta}{\frac{\sin(\pi lk/n)}{k\sin(\pi l/n)}} \;<\; 1 \quad \text{for} \quad l\neq n$$

and $\hat{b}_n=1$. The only 1-vector $\boldsymbol{v}^n=\mathbf{1}$ corresponds to a constant shift $y_i\rightsquigarrow y_i+c$ under which kNN (like many other regressors) is invariant. Instead of regularizing LoRP with $\alpha>0$ we can restrict $V(L)\subset\mathbb{R}^n$ to the space orthogonal to $\boldsymbol{v}^n$, which means dropping $\hat{b}_n=1$ in the determinant. Intuitively, since this invariant direction is the same for all k, we can drop the same additive infinite constant from LR for every k, which is irrelevant for comparisons (formally we should compute $\lim_{\alpha\to0}\{\text{LR}^\alpha_{k_1}-LR^\alpha_{k_2}\}$). The exact expression for the restricted log-determinant (denoted by a prime) is

$$-\tfrac{1}{2}\log\det{}'S_0 \;=\; -\log\det{}'(\mathbb{1}-M) \;=\; -\sum_{l=1}^{n-1}\log(1-\hat{b}_l) \;=:\; \tfrac{n}{k}c^1_{nk} \;=\; c^1_{nk}\mathrm{tr}M$$

For large n (and large k) the expression can be simplified. The exact, large n, and

large $k \ll n$ expressions are

$$\begin{aligned} c^1_{nk} &= -\frac{k}{n}\sum_{l=1}^{n-1}\log\Big(1-\frac{\sin(\pi lk/n)}{k\sin(\pi l/n)}\Big) \\ c^1_{\infty k} &= -\frac{k}{\pi}\int_{-\pi/2}^{\pi/2}\log\Big(1-\frac{\sin(kz)}{k\sin(z)}\Big)dz \qquad \begin{pmatrix} z=\pi l/n \text{ for } l<\frac{n}{2} \\ z=\pi l/n-\pi \text{ else}\end{pmatrix} \\ c^1_{\infty\infty} &= -\frac{1}{\pi}\int_{-\infty}^{\infty}\log\Big(1-\frac{\sin t}{t}\Big)\,dt \doteq 3.202 \qquad (t=kz, \sin(z)\sim z) \end{aligned}$$

Further, $c^1_{\infty 3}=3\log 3\doteq 3.295$. Since $c^1_{\infty k}$ is decreasing in k, $c^1_{\infty k}$ equals 3.2 within 3% for all k.

kNN on d-dimensional grid. We now consider $\boldsymbol{x}=\mathcal{X}^d=\{1,...,n_1\}^d$ on a d-dimensional complete hypercube grid with $n=n_1^d$ points and Manhattan distance $d(x_{\boldsymbol{i}},x_{\boldsymbol{j}})=d(\boldsymbol{i},\boldsymbol{j})=\sum_{a=1}^d d_1(i_a,j_a)$ for all $x_{\boldsymbol{i}}=\boldsymbol{i}\in\mathcal{X}^d$ and $x_{\boldsymbol{j}}=\boldsymbol{j}\in\mathcal{X}^d$, where d_1 is the one-dimensional circular distance defined above (so actually $\mathcal{X}^d$ is a discrete torus). For $k=k_1^d$, the neighborhood $\mathcal{N}_k(x)$ of x is a cube of side-length k_1. In this case, $M=M_1\otimes...\otimes M_1$ is a d-fold tensor product of the 1d k$_1$NN matrices M_1 of sample size n_1. The eigenvectors of M are $\boldsymbol{v}^{l_1}\otimes...\otimes\boldsymbol{v}^{l_d}$ with eigenvalues $\hat{b}_{l_1}\cdot....\cdot\hat{b}_{l_d}$. We get

$$\begin{aligned} -\log\det{}'(\mathbb{1}-M) &= -\sum_{l_1=1}^{n_1-1}...\sum_{l_d=1}^{n_d-1}\log(1-\hat{b}_{l_1}\cdot...\cdot\hat{b}_{l_d}) \qquad (30) \\ &\overset{n\gg k\to\infty}{\longrightarrow} -\frac{1}{\pi^d}\int_{\mathbb{R}^d}\log\Big(1-\prod_{a=1}^d\frac{\sin t_a}{t_a}\Big)d^d\boldsymbol{t} \;=:\; \tfrac{n}{k}c^d_{\infty\infty} \end{aligned}$$

For instance, for $d=2$, numerical integration gives $c^2_{\infty\infty}\doteq 2.2$ compared to 3.2 in one dimension. For higher dimensions, evaluation of the d-dimensional integral becomes cumbersome, and we resort to a different approximation.

Taylor series in M. We can also (not only for kNN) expand logdetS_0 in a Taylor series in M:

$$\begin{aligned} -\log\det{}'(\mathbb{1}-M) &= -\mathrm{tr}'\log(\mathbb{1}-M) \;=\; \sum_{s=1}^{\infty}\tfrac{1}{s}\mathrm{tr}'(M^s) \\ &= \sum_{s=1}^{\infty}\tfrac{1}{s}(\mathrm{tr}'M_1^s)^d \;=\; \tfrac{n}{k}\sum_{s=1}^{\infty}\tfrac{1}{s}(A_{n_1k_1s})^d \;=:\; \tfrac{n}{k}c^d_{nk} \end{aligned}$$

where we used $\mathrm{tr}(A\otimes B)=\mathrm{tr}(A)\cdot\mathrm{tr}(B)$ and $(A\otimes B)^s=A^s\otimes B^s$ and defined

$$A_{n_1k_1s} \;:=\; \frac{k_1}{n_1}\mathrm{tr}'(M_1^s) \;=\; \frac{k_1}{n_1}\sum_{l=1}^{n_1-1}(\hat{b}_l)^s \;\overset{n\gg k\to\infty}{\longrightarrow}\; \frac{1}{\pi}\int_{-\infty}^{\infty}\Big(\frac{\sin t}{t}\Big)^s dt$$

The one-dimensional integral can be expressed as a finite sum with s terms or evaluated numerically. For any n and k one can show that $A_{nk1}=A_{nk2}=1>A_{nks}$ for

$s>2$. So the expansion above is useful for large d. Note also that c_{nk}^d is monotone decreasing in d. For $d\to\infty$ we have

$$c_{nk}^\infty \;=\; \sum_{s=1}^{\infty} \tfrac{1}{s}(A_{nks})^\infty \;=\; 1+\tfrac{1}{2}+0+... \;=\; \tfrac{3}{2}$$

i.e. c_{nk}^d decreases monotone in d from about 3.2 to $\frac{3}{2}$.

The practical implication of this observation, though, is limited, since $k=k_1^d\to\infty$ is actually not fixed for $d\to\infty$. Indeed, in practical high-dimensional problems, $k\ll n\ll 3^d$, but in our grid example $k=k_1^d\geq 3^d$. Real data do not form full grids but sparse neighborhoods if d is large.

11 Conclusion and Outlook

We introduced a new method, the Loss Rank Principle, for model selection. The loss rank of a model is defined as the number of other data that fit the model better than the training data. The model chosen by LoRP is the one of smallest loss rank. The loss rank has an explicit expression in case of linear models. Model consistency and asymptotic efficiency of LoRP were considered. The numerical experiments suggest that LoRP works well in practice. A comparison between LoRP and other methods for model selection was also presented.

In this paper, we have only scratched at the surface of LoRP. LoRP seems to be a promising principle with a lot of potential, leading to a rich field. In the following we briefly summarize miscellaneous considerations.

Comparison to Rademacher complexities. For a (binary) classification problem, the rank (1) of classifier r can be re-formulated as the probability that a randomly relabeled sample $\boldsymbol{y}'$ behaves better than the actual $\boldsymbol{y}$. The more flexible r is, the larger its rank is. The Rademacher complexity [Kol01, BBL02] of r is the expectation of the difference between the misclassifying loss under the actual $\boldsymbol{y}$ and the misclassifying loss under a randomly relabeled sample $\boldsymbol{y}'$. The more flexible r is, the larger its Rademacher complexity is. Therefore, there is a close connection between LoRP and Rademacher complexities. Model selection based on Rademacher complexities has a number of attractive properties and has been attracting many researchers, thus it's worth discovering this connection. Some results have been recently already obtained, however, to keep the present paper not so long, we decide to present the results in another paper.

Monte Carlo estimates for non-linear LoRP. For non-linear regression we did not present an efficient algorithm for the loss rank/volume $\mathrm{LR}_r(\boldsymbol{y}|\boldsymbol{x})$. The high-dimensional volume $|V_r(L)|$ (3) may be computed by Monte Carlo algorithms. Normally $V_r(L)$ constitutes a small part of $\mathcal{Y}^n$, and uniform sampling over $\mathcal{Y}^n$ is not feasible. Instead one should consider two competing regressors r and r' and compute $|V\cap V'|/|V|$ and $|V\cap V'|/|V'|$ by uniformly sampling from V and V' respectively e.g., with a Metropolis-type algorithm. Taking the ratio we get $|V'|/|V|$ and hence the

loss rank difference $\mathrm{LR}_r - \mathrm{LR}_{r'}$, which is sufficient for LoRP. The usual tricks and problems with sampling apply here too.

LoRP for hybrid model classes. LoRP is not restricted to model classes indexed by a single integral "complexity" parameter, but may be applied more generally to selecting among some (typically discrete) class of models/regressors. For instance, the class could contain kNN *and* polynomial regressors, and LoRP selects the complexity *and* type of regressor (non-parametric kNN versus parametric polynomials).

Generative versus discriminative LoRP. We have concentrated on counting y's given fixed x, which corresponds to discriminative learning. LoRP might equally well be used for counting (x,y), as alluded to in the introduction. This would correspond to generative learning. Both regimes are used in practice. See [LJ08] for some recent results on their relative benefit, and further references.

Acknowledgement. We would like to thank two anonymous reviewers for their detailed and helpful comments. The second author would like to thank the SML@NICTA for supporting a visit which led to the present paper.

Appendix: List of Abbreviations and Notations

AIC= Akaike Information Criterion.
BIC= Bayesian Information Criterion.
BMS= Bayesian Model Selection
kNN= k Nearest Neighbors.
LBFR= Linear Basis Function Regression.
LoRP= Loss Rank Principle.
LRC = Loss Rank Code.
MAP= Maximum a Posterior.
MDL= Minimum Description Length.
ML= Maximum Likelihood.
PML= Penalized Maximum Likelihood.
$D=\{(x_1,y_1),...,(x_n,y_n)\}$= observed data.
$\mathcal{D}=\{D\}$= set of all possible data D.
$\mathcal{X}\times\mathcal{Y}$=observation space.
$\boldsymbol{x}=(x_1,...,x_n)$= vector of x-observations, similarly $\boldsymbol{y}$.
$f:\mathcal{X}\to\mathcal{Y}$= functional dependence between x and y.
$\mathcal{F}$= ("small") class of functions f.
$\mathcal{H}$= class of stochastic hypotheses/models.
$r:\mathcal{D}\to\mathcal{F}$= regressor/model.
$\hat{y}_i=r(x_i|D)$= r-estimate of y_i.
$\mathcal{R}$= ("small") class of regressors/models.
$\boldsymbol{w}\in\mathbb{R}^d$= parametrization of $\mathcal{F}_d$.
$\mathcal{N}_k(x)$= set of indices of the k nearest neighbors of x in D.
$L=\mathrm{Loss}_r(D)=\mathrm{Loss}(\boldsymbol{y},\hat{\boldsymbol{y}})$= empirical loss of r on D.

$\text{Rank}_r(L) = \#\{\boldsymbol{y}' \in \mathcal{Y}^n : \text{Loss}_r(\boldsymbol{y}'|\boldsymbol{x}) \leq L\}$= loss rank of r.
$V(L)$= volume of D under r.
$\text{LR}_r(\boldsymbol{y}|\boldsymbol{x})$= log rank/volume of D.
LR_r^α= regularized LR_r.
d_{eff}= effective dimension.
$m_j(x,\boldsymbol{x})$= coefficients of linear regressor.
$M(\boldsymbol{x})$= linear regression matrix or "hat" matrix.
log= natural logarithm.
$a \rightsquigarrow b$: a is replaced by b.

References

[Aka73] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proc. 2nd International Symposium on Information Theory*, pages 267–281, Budapest, Hungary, 1973. Akademiai Kaidó.

[All74] D. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127, 1974.

[BBL02] P. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. *Machine Learning*, 48:85–113, 2002.

[Cha06] A. Chambaz. Testing the order of a model. *Ann. Stat.*, 34(3):1166–1203, 2006.

[CW79] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the methods of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

[ET93] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, New York, 1993.

[Grü04] P. D. Grünwald. Tutorial on minimum description length. In *Minimum Description Length: recent advances in theory and practice*, page Chapters 1 and 2. MIT Press, 2004. http://www.cwi.nl/~pdg/ftp/mdlintro.pdf.

[Her02] R. Herbrich. *Learning Kernel Classifiers*. The MIT Press, 2002.

[HT89] C. M. Hurvich and C. L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

[HTF01] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[Hut07] M. Hutter. The loss rank principle for model selection. In *Proc. 20th Annual Conf. on Learning Theory (COLT'07)*, volume 4539 of *LNAI*, pages 589–603, San Diego, 2007. Springer, Berlin.

[Kol01] V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Trans. Inform. Theory*, 47:1902–1914, 2001.

[LJ08] P. Liang and M. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proc. 25th International Conf. on Machine Learning (ICML-2008)*, volume 307, pages 584–591. ACM, 2008.

[Mac92] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[Mil02] A. Miller. *Subset Selection in Regression*. Chapman & Hall/CRC, 2002.

[Reu02] A. Reusken. Approximation of the determinant of large sparse symmetric positive definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 23(3):799–818, 2002.

[Ris78] J. J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[Sha97] J. Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, 7:221–264, 1997.

[Shi83] R. Shibata. Asymptotic mean efficiency of a selection of regression variables. *Annals of the Institute of Statistical Mathematics*, 35:415–423, 1983.

[WLT07] H. Wang, R. Li, and C. L. Tsai. Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, 3(94):553–568, 2007.

[Yam99] K. Yamanishi. Extended stochastic complexity and minimax relative loss analysis. In *In Proc. 10th International Conference on Algorithmic Learning Theory - ALT' 99*, pages 26–38. Springer-Verlag, 1999.

[Yan05] Y. Yang. Can the strengths of aic and bic be shared? a conflict between model identification and regression estimation. *Biometrika*, 92(4):937–950, 2005.