The Loss Rank Principle for Model Selection

Marcus Hutter

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

February 2006

Abstract

We introduce a new principle for model selection in regression and classification. Many regression models are controlled by some smoothness or flexibility or complexity parameter c, e.g. the number of neighbors to be averaged over in k nearest neighbor (kNN) regression or the polynomial degree in regression with polynomials. Let $\hat{f}_D^c$ be the (best) regressor of complexity c on data D. A more flexible regressor can fit more data D' well than a more rigid one. If something (here small loss) is easy to achieve it's typically worth less. We define the loss rank of $\hat{f}_D^c$ as the number of other (fictitious) data D' that are fitted better by $\hat{f}_{D'}^c$ than D is fitted by $\hat{f}_D^c$. We suggest selecting the model complexity c that has minimal loss rank (LoRP). Unlike most penalized maximum likelihood variants (AIC,BIC,MDL), LoRP only depends 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. In this paper we formalize, discuss, and motivate LoRP, study it for specific regression problems, in particular linear ones, and compare it to other model selection schemes.

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 $y_i \approx f_{true}(x_i)$ from data $D=\{(x_1,y_1),...,(x_n,y_n)\} \in \mathcal{D}$, i.e. we seek a function f_D such that $f_D(x)$ is close to the unknown $f_{true}(x)$ for all x. One may define regressor f_D directly, e.g. 'average the y values of the k nearest neighbors (kNN) of x in D', or select the f 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:

We will not discuss empirical test set methods like cross-validation, but only training set based methods. See e.g. [Mac92] for a comparison of cross-validation with Bayesian model selection. Training set based model selection methods allow using 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 $\mathcal{F}$, one has to specify a sampling model $\mathrm{P}(D|f)$, e.g. that the y_i have independent Gaussian distribution with mean $f(x_i)$. ML chooses $\hat{f}_D^c = \mathrm{argmax}_{f\in\mathcal{F}_c}\mathrm{P}(D|f)$, Penalized ML (PML) then chooses $\hat{c} = \mathrm{argmin}_c\{-\log\mathrm{P}(D|\hat{f}_D^c)+\mathrm{Penalty}(c)\}$, where the penalty depends on the used approach (MDL [Ris78], BIC [Sch78], AIC [Aka73]). In particular, modern MDL [Grü04] has sound exact foundations and works very well in practice. 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, 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 $\hat{f}_D^c$ given as a black box without insight into the origin or inner structure of $\hat{f}_D^c$, that does not depend on things often not given (like a stochastic noise model), and that exploits what is given (like the loss function). The key observation we exploit is that large classes $\mathcal{F}_c$ or more flexible regressors $\hat{f}_D^c$ can fit more data $D' \in \mathcal{D}$ well than more rigid ones, e.g. many D' can be fit well with high order polynomials. We define the *loss rank* of $\hat{f}_D^c$ as the number of other (fictitious) data $D' \in \mathcal{D}$ that are fitted better by $\hat{f}_{D'}^c$ than D is fitted by $\hat{f}_D^c$, 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 new Loss Rank Principle (LoRP) works without a noise (stochastic sampling) model, and is directly applicable to any non-parametric regressor, like kNN.

Contents. In Section 2, after giving a brief introduction to regression, we formally state LoRP for model selection. To make it applicable to real problems, we have to generalize it to continuous spaces and regularize infinite loss ranks. In Section 3 we derive explicit expressions for the loss rank for the important class of linear regressors, which includes kNN, polynomial, linear basis function (LBFR), Kernel, and projective regression. In Section 4 we compare linear LoRP to Bayesian model selection for linear regression with Gaussian noise and prior, and in Section 5 to PML, in particular MDL, BIC, AIC, and MacKay's [Mac92] and Hastie's et al. [HTF01] trace formulas for the effective dimension. In this paper we just scratch at the surface of LoRP. Section 6 contains further considerations, to be elaborated on in the future.

2 The Loss Rank Principle (LoRP)

After giving a brief introduction to regression, classification, model selection, overfitting, and some reoccurring examples (polynomial regression Example 1 and kNN Example 2), 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 (oversimplistic) artificial Examples 4 and 6. Thereafter we show how to regularize LoRP for realistic regression problems.

Setup. 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. $y_i\approx f_{true}(x_i)$, where $\approx$ means that the y_i are distorted by noise or otherwise from the unknown "true" values $f_{true}(x_i)$.

Regression and classification. In regression problems $\mathcal{Y}$ is typically (a subset of) the real numbers $I\!R$ or some more general measurable space like $I\!R^m$. 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 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 regression function $r:\mathcal{D}\to\mathcal{F}$ such that $\hat{y}:=r(x|D)\equiv r(D)(x)\equiv f_D(x)\approx f_{true}(x)$ for all $x\in\mathcal{X}$.

Notation. We will write (x,y) or (x_0,y_0) 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.

Example 1 (polynomial regression) For $\mathcal{X}=\mathcal{Y}=I\!R$, consider the set $\mathcal{F}_d:=\{f_{\boldsymbol{w}}(x)=w_dx^{d-1}+...w_2x+w_1:\boldsymbol{w}\in I\!R^d\}$ 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, kNN) Let $\mathcal{Y}$ be some vector space like $I\!R$ and $\mathcal{X}$ be a metric space like $I\!R^m$ with some (e.g. Euclidian) metric $d(\cdot,\cdot)$. kNN estimates $f_{true}(x)$ 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 regressors r 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 (typically countable) class of regressors $\mathcal{R}$ (whatever their origin), e.g. the kNN regressors $\{r_k:k\in I\!N\}$ or the least squares polynomial regressors $\{r_d:d\in I\!N_0\}$. 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 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. More flexible r can closer fit the data and hence have smaller loss, but are not necessarily better, since they have 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 for the "best" regressor $r\in\mathcal{R}$

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

While for parametric (e.g. polynomial) regression, MDL and Bayesian methods work well (effectively the number of parameters serve as complexity penalty), their use is seriously limited for non-parametric black box r like kNN or if a stochastic/coding model is hard to establish (see Section 4 for a detailed comparison).

Main idea: loss rank. 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. For instance, r_d can perfectly fit all D' for $d = n$, all D' that lie on a parabola for $d = 3$, but only linear D' for $d = 2$. We consider discrete $\mathcal{Y}$ i.e. classification first, and fix $\boldsymbol{x}$. $\boldsymbol{y}$ is the observed data and $\boldsymbol{y}'$ are fictitious others.

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 $\boldsymbol{y}$. Many $\boldsymbol{y}'$ have small loss for flexible r, and so smallness of Loss_r is less significant than if $\boldsymbol{y}$ is among very few other $\boldsymbol{y}'$ with small Loss_r. We claim that the loss rank of $\boldsymbol{y}$ among all $\boldsymbol{y}' \in \mathcal{Y}^n$ is a suitable measure of fit. We define the rank of $\boldsymbol{y}$ under r as the number of $\boldsymbol{y}' \in \mathcal{Y}^n$ with smaller or equal loss than $\boldsymbol{y}$:

$$\begin{aligned} \text{Rank}_r(\boldsymbol{y}|\boldsymbol{x}) \;\equiv\; \text{Rank}_r(L) \;&:=\; \#\{\boldsymbol{y}' \in \mathcal{Y}^n : \text{Loss}_r(\boldsymbol{y}'|\boldsymbol{x}) \leq L\}, \\ \text{where}\quad L \;&:=\; \text{Loss}_r(\boldsymbol{y}|\boldsymbol{x}) \end{aligned} \tag{1}$$

For this 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. In a sense, $\rho = \text{Rank}_r(\boldsymbol{y}|\boldsymbol{x})$ measures how compatible $\boldsymbol{y}$ is with r; $\boldsymbol{y}$ is the ρth most compatible with r.

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 (loss rank principle (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 of 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 better grasping how the principle works, but the example is too simplistic to allow any conclusion on whether the principle is appropriate.

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 regression $\mathcal{R}=\{r_0,r_1,r_2\}$ (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	$r_d(x\vert\boldsymbol{x},\boldsymbol{y}')$	$\text{Loss}_d(\boldsymbol{y}'\vert\boldsymbol{x})$	$\text{Loss}_d(D)$
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).

	$\text{Rank}_{r_d}(y'_1y'_2\vert 12)$	
d	1 2 3 4 5 6 7 8 9	$\text{Rank}_{r_d}(D)$
0	$y'_1y'_2 = 00 < 01 = 10 < 11 < 02 = 20 < 21 = \mathbf{12} < 22$	8
1	$y'_1y'_2 = 00 = 11 = 22 < 01 = 10 = 21 = \mathbf{12} < 02 = 20$	7
2	$y'_1y'_2 = 00 = 01 = 02 = 10 = 11 = 20 = 21 = 22 = \mathbf{12}$	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 $\mathcal{Y}=I\!R$ in the following exposition, but the idea and resulting principle hold for more general measurable spaces like $I\!R^m$. We simply reduce the model selection problem to the discrete case by considering the discretized space $\mathcal{Y}_\varepsilon=\varepsilon Z\!\!Z$ for small $\varepsilon>0$ and discretize $\boldsymbol{y}\rightsquigarrow\boldsymbol{y}_\varepsilon\in\varepsilon Z\!\!Z^n$. 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 $L=\text{Loss}_r(\boldsymbol{y}_\varepsilon|\boldsymbol{x})$ counts 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 (loss rank principle 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 of smallest loss volume:*

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

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 Loss Rank 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	$V_d(L)=\{\boldsymbol{y}'\in[0,2]^2:...\}$	$\lvert V_d(L)\rvert$	$\lvert V_d(\mathrm{Loss}_d(D))\rvert$
0	${y'_1}^2+{y'_2}^2\leq L$	$2\sqrt{\max\{L-4,0\}}+$ $L(\frac{\pi}{4}-\cos^{-1}(\min\{\frac{2}{\sqrt{L}},1\}))$	≈ 3.6
1	$\frac{1}{2}(y'_2-y'_1)^2\leq L$	$4\sqrt{2L}-2L$	3
2	$0\leq L$	4	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. We will encounter such infinities in Section 3. 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.

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

The Euclidian norm $||\boldsymbol{y}||^2:=\sum_{i=1}^n y_i^2$ is default, but other (non)norm regularizes are possible. The regularized $\mathrm{LR}_r^\alpha(\boldsymbol{y}|\boldsymbol{x})$ 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 α by minimizing LR_r^α w.r.t. α. The good news is that this leads to meaningful results.

3 LoRP for Linear Models

In this section we consider the important class of linear regressors with quadratic loss function. Since linearity is only assumed in y and the dependence on x can be arbitrary, this class is richer than it may appear. It includes kNN (Example 7), kernel (Example 8), and many other regressors. For linear regression and $\mathcal{Y}=I\!R$, 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).

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

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

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

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

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. ♢

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 $\mathcal{X}=I\!R^m$ is $K(x,x_j)=\mathrm{e}^{-||x-x_j||_2^2/2\sigma^2}$. ♢

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 $\boldsymbol{\phi}:\mathcal{X}\to I\!R^d$. Consider the class of functions linear in $\boldsymbol{\phi}$:

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

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

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

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

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

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

($1\!\!1$ 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. $V(L)=\{\boldsymbol{y}'\in I\!R^n : \boldsymbol{y}'^\top S_\alpha \boldsymbol{y}' \leq L\}$ 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

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

Consider now a *class* of linear regressors $\mathcal{R}=\{M\}$, e.g. the kNN regressors $\{M_k : k\in I\!N\}$ or the d-dimensional linear basis function regressors $\{M_d : d\in I\!N_0\}$.

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

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

where S_α is defined in (8).

Since v_n is independent of α and M it was possible to drop v_n. 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 $\boldsymbol{S_0}$. If M has an eigenvalue 1, then $S_0 = (1\!\!1 - M)^\top(1\!\!1 - M)$ has a zero eigenvalue and $\alpha>0$ is necessary, since $\det S_0 = 0$. Actually this is true for most practical M. Nearly all linear regressors are invariant under a constant shift of $\boldsymbol{y}$, i.e. $r(y_i+c|D) = r(y_i|D)+c$, 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

[1] The mentioned alternative penalty $\alpha||\hat{\boldsymbol{y}}||^2$ would lead to $S_\alpha = (1\!\!1 - M)^\top(1\!\!1 - M) + \alpha M^\top M$. 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 $\phi_a(x) \rightsquigarrow \gamma_a\phi_a(x)$ doesn't change the class $\mathcal{F}_d$, but changes the ridge regressor.

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. 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_{min} analytically.

Projective regression. Consider a projection matrix $M = P = P^2$ with $d = \text{tr}P$ eigenvalues 1, and $n-d$ zero eigenvalues. For instance, $M = \mathbf{\Phi} B^{-1} \mathbf{\Phi}^\top$ of LBFR Ex.9 is such a matrix, since $M\mathbf{\Phi} = \mathbf{\Phi}$ and $M\mathbf{\Psi} = 0$ for $\mathbf{\Psi}$ such that $\mathbf{\Phi}^\top \mathbf{\Psi} = 0$. This implies that S_α has d eigenvalues α and $n-d$ eigenvalues $1+\alpha$. Hence

$$\begin{aligned} \det S_\alpha &= \alpha^d (1+\alpha)^{n-d}, \quad \text{where} \quad S_\alpha = S_0 + \alpha\mathbb{1} = \mathbb{1} - P + \alpha\mathbb{1} \\ \boldsymbol{y}^\top S_\alpha \boldsymbol{y} &= (\rho+\alpha)\boldsymbol{y}^\top\boldsymbol{y}, \quad \text{where} \quad \rho := \frac{\boldsymbol{y}^\top S_0 \boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}} = 1 - \frac{\boldsymbol{y}^\top P \boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}} \\ \Rightarrow \quad \text{LR}_P^\alpha &= \tfrac{n}{2}\log \boldsymbol{y}^\top\boldsymbol{y} + \tfrac{n}{2}\log(\rho+\alpha) - \tfrac{d}{2}\log\alpha - \tfrac{n-d}{2}\log(1+\alpha) \end{aligned} \tag{11}$$

The first term is independent of α. Consider $1-\rho > \frac{d}{n}$, the reasonable region in practice. Solving $\partial\text{LR}_P^\alpha / \partial\alpha = 0$ w.r.t. α we get a minimum at $\alpha = \alpha_{min} := \frac{\rho d}{(1-\rho)n - d}$. After some algebra we get

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

is the relative entropy or Kullback-Leibler divergence. Minimizing $\text{LR}_P^{\alpha_{min}}$ w.r.t. M is equivalent to maximizing $\text{KL}(\frac{d}{n}||1-\rho)$. This is an unusual task, since one mostly encounters D minimizations. For fixed d, $\text{LR}_P^{\alpha_{min}}$ is monotone increasing in ρ. Since $\text{Loss}_P^\alpha \propto \rho + \alpha$, LoRP suggests to minimize Loss for fixed model dimension d. For fixed ρ, $\text{LR}_P^{\alpha_{min}}$ is monotone increasing in d, i.e. LoRP suggests to minimize model dimension d for fixed Loss. Normally there is a tradeoff between minimizing d and ρ, and LoRP suggests that the optimal choice is the one that maximizes 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*

$$P^{best} \;=\; \arg\max_{P\in\mathcal{P}} \text{KL}\Big(\tfrac{\text{tr}P(x)}{n} \Big|\Big| \tfrac{\boldsymbol{y}^\top P(x)\boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}}\Big), \quad \textit{provided} \quad \tfrac{\text{tr}P}{n} < \tfrac{\boldsymbol{y}^\top P\boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}}$$

4 Comparison to Gaussian Bayesian Linear Regression

We now consider linear basis function regression (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 an ML-II 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 $f_{\boldsymbol{w}}(x)=\boldsymbol{w}^\top\boldsymbol{\phi}(x)$ ($\boldsymbol{w}\in I\!R^d$) that are linear in feature vector $\boldsymbol{\phi}$. Let

$$\text{Gauss}_N(\boldsymbol{z}|\boldsymbol{\mu},\Sigma) \;:=\; \frac{\exp(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^\top\Sigma^{-1}(\boldsymbol{z}-\boldsymbol{\mu}))}{(2\pi)^{N/2}\sqrt{\det\Sigma}} \tag{13}$$

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 $\text{P}(\boldsymbol{y}|\boldsymbol{w})=\text{Gauss}_n(\boldsymbol{y}|\boldsymbol{\Phi w},\beta^{-1}1\!\!1)$, where $\boldsymbol{\Phi}_{ia}=\boldsymbol{\phi}_a(x_i)$. 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. $\text{P}(\boldsymbol{w})=\text{Gauss}_d(\boldsymbol{w}|\boldsymbol{0},\alpha^{-1}C^{-1})$. From the prior and the likelihood one can compute the evidence and the posterior

$$\begin{aligned} &\text{Evidence:} && \text{P}(\boldsymbol{y}) \;=\; \int \text{P}(\boldsymbol{y}|\boldsymbol{w})\text{P}(\boldsymbol{w})d\boldsymbol{w} \;=\; \text{Gauss}_n(\boldsymbol{y}|\boldsymbol{0},\beta^{-1}S^{-1}) && (14)\\ &\text{Posterior:} && \text{P}(\boldsymbol{w}|\boldsymbol{y}) \;=\; \text{P}(\boldsymbol{y}|\boldsymbol{w})\text{P}(\boldsymbol{w})/P(\boldsymbol{y}) \;=\; \text{Gauss}_d(\boldsymbol{w}|\hat{\boldsymbol{w}},A^{-1}) \end{aligned}$$

$$\begin{gathered} B:=\boldsymbol{\Phi}^\top\boldsymbol{\Phi}, \quad A:=\alpha C+\beta B, \quad M:=\beta\boldsymbol{\Phi}A^{-1}\boldsymbol{\Phi}^\top, \quad S:=1\!\!1-M, \\ \hat{\boldsymbol{w}}:=\beta A^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}, \quad \hat{\boldsymbol{y}}:=\boldsymbol{\Phi}\hat{\boldsymbol{w}}=M\boldsymbol{y} \end{gathered} \tag{15}$$

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) $\hat{\boldsymbol{w}}=\text{argmax}_{\boldsymbol{w}}\text{P}(\boldsymbol{w}|\boldsymbol{y})=\beta A^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}$. 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 $\text{P}_\mathcal{F}(\boldsymbol{y})$. Bayesian Model Selection (BMS) chooses the model class (here d) $\mathcal{F}$ of highest evidence:

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

Once the model class $\mathcal{F}^{\text{BMS}}$ is determined, the MAP (or other) regression function $f_{\boldsymbol{w}_{\mathcal{F}^{\text{BMS}}}}$ or $M_{\mathcal{F}^{\text{BMS}}}$ are chosen. The data variance β^{-1} may be known or estimated from the data, C is often chosen $1\!\!1$, 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 (13) into (14) and taking the logarithm we see that BMS minimizes

$$-\log \mathrm{P}_{\mathcal{F}}(\boldsymbol{y}) \;=\; \tfrac{\beta}{2}\boldsymbol{y}^\top S\boldsymbol{y} - \tfrac{1}{2}\log\det S - \tfrac{n}{2}\log\tfrac{\beta}{2\pi} \tag{16}$$

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 (16) we get

$$-\log \mathrm{P}_{\mathcal{F}}(\boldsymbol{y}) \;=\; \tfrac{n}{2}\log \boldsymbol{y}^\top S\boldsymbol{y} - \tfrac{1}{2}\log\det S - \tfrac{n}{2}\log\tfrac{n}{2\pi\mathrm{e}} \tag{17}$$

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 (17) 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, $M = \gamma\boldsymbol{\Phi}B^{-1}\boldsymbol{\Phi}^\top$, with $\gamma = \frac{\beta}{\alpha+\beta} < 1$ for $\alpha > 0$, is a simple multiplicative regularization of projection $\boldsymbol{\Phi}B^{-1}\boldsymbol{\Phi}^\top$, and (17) coincides with (11) for suitable α, apart from an irrelevant additive constant, hence minimizing (17) over α also leads to (12).

5 Comparison to other Model Selection Schemes

In this section we give a brief introduction to Penalized Maximum Likelihood (PML) for (semi)parametric regression, and its major instantiations, the Akaike and the Bayesian Information Criterion (AIC and BIC), and the Minimum Description Length (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 [Mac92] trace formula for Gaussian Bayesian LBFR and Hastie's et al. [HTF01] trace formula 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 4. Let $\mathrm{P}_d(\boldsymbol{y}|\boldsymbol{w})$ be the data likelihood under d-dimensional model $\boldsymbol{w} \in I\!R^d$. The maximum likelihood (ML) estimator for fixed d is

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

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

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

The penalty introduces a tradeoff between the first and second term with a minimum at $\hat{d}<\infty$. Various penalties have been suggested: The Akaike Information Criterion (AIC) [Aka73] uses d, the Bayesian Information Criterion (BIC) [Sch78] and the (crude) Minimum Description Length (MDL) principle use $\frac{d}{2}\log n$ [Ris78, Grü04] for 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 (AIC,BIC,MDL) is mostly limited to (semi)parametric models (with d "true" parameters).

We discuss two approaches to the last item in the remainder of this section: 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. For instance, ridge regression can work even for d larger than n, because a penalty pulls most parameters towards (but not exactly to) zero. MacKay [Mac92] suggests an expression for the effective number of parameters d_{eff} as a substitute for d in case of (b), and Hastie et al. [HTF01] more generally also for (a).

The trace penalty for parametric Gaussian LBFR. We continue with the Gaussian Bayesian linear regression example (see Section 4 for details and notation). Performing the integration in (14), MacKay [Mac92, Eq.(21)] derives the following expression for the Bayesian evidence for $C=\mathbb{1}$

$$\begin{aligned} -\log\mathrm{P}(\boldsymbol{y}) \;&=\; (\alpha\hat{E}_W+\beta\hat{E}_D)+(\tfrac{1}{2}\log\det A-\tfrac{d}{2}\log\alpha)-\tfrac{n}{2}\log\tfrac{\beta}{2\pi} \qquad (18)\\ \hat{E}_D \;&=\; \tfrac{1}{2}||\boldsymbol{\Phi}\hat{\boldsymbol{w}}-\boldsymbol{y}||_2^2, \quad \hat{E}_W=\tfrac{1}{2}||\hat{\boldsymbol{w}}||_2^2 \end{aligned}$$

(the first bracket in (18) equals $\frac{\beta}{2}\boldsymbol{y}^\top S\boldsymbol{y}$ and the second equals $-\frac{1}{2}\log\det S$, cf. (16)). Minimizing (18) 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

$$d_{\mathit{eff}}^{\mathrm{McK}} \;:=\; \alpha||\hat{\boldsymbol{w}}||_2^2 \;=\; 2\alpha\hat{E}_W \;=\; d-\alpha\mathrm{tr}A^{-1} \qquad (19)$$

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, Hastie et al. [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 $\beta_1,...,\beta_n$ are the shrinkages = eigenvalues of M, the effective number of parameters could be defined as [HTF01, Sec.7.6]

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

which generalizes the relation $d = \mathrm{tr}M$ beyond projections. For MacKay's M (15), $\mathrm{tr}M = d - \mathrm{tr}A^{-1}$, i.e. $d_{\mathit{eff}}^{\mathrm{HTF}}$ is consistent with and generalizes $d_{\mathit{eff}}^{\mathrm{McK}}$.

Problems. Though nicely motivated, the trace formula is not without problems. First, since for projections, $M = M^2$, one could equally well have argued for $d_{\mathit{eff}}^{\mathrm{HTF}} = \mathrm{tr}M^2$. 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' where 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 $d_{\mathit{eff}}^{\mathrm{HTF}} = 0$ 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 $S_0 = (1\!\!1 - M)^\top(1\!\!1 - M)$ (8), we can Taylor expand $-\frac{1}{2}\mathrm{logdet}S_0$ in M

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

For BMS (17) with $S = 1\!\!1 - M$ (15) 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'.

6 Outlook

So far we have only scratched at the surface of the Loss Rank Principle. 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, which may be elaborated on in the future: Experiments, Monte Carlo estimates for non-linear LoRP, numerical approximation of $\det S_\alpha$, LoRP for classification, self-consistent regression, explicit expressions for kNN on a grid, loss function selection, and others.

Experiments. Preliminary experiments on selecting k in kNN regression confirm that LoRP selects a "good" k. (Even on artificial data we cannot determine whether the "right" k is selected, since kNN is not a generative model). LoRP for LBFR seems to be consistent with rapid convergence.

Monte Carlo estimates for non-linear LoRP. For non-linear regression we did not present an efficient algorithm for the loss rank/volume $\text{LR}_r(\boldsymbol{y}|\boldsymbol{x})$. 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 $\text{LR}_r-\text{LR}_{r'}$, which is sufficient for LoRP. The usual tricks and problems with sampling apply here too.

Numerical approximation of $\det\boldsymbol{S_\alpha}$. Even for linear regression, a Monte Carlo algorithm may be faster than the naive $O(n^3)$ algorithm [BFG96]. 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 to approximate $\det S_\alpha$, sometimes in linear time [Reu02].

LoRP for classification. A classification problem is or can be regarded as a regression problem in which $\mathcal{Y}$ is finite. This implies that we need to compute (count) LR_r for non-linear r somehow, e.g. approximately by Monte Carlo.

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 $x\not\in\{x_1,...,x_n\}$. One can construct canonical regressors for off-data x from regressors given only on-data in the following way: We add a virtual data point (x,y) to D, where x is the off-data point of interest. If we knew y we could estimate $\hat{y}=r(x|\{(x,y)\}\cup D)$, but we don't know y. But if we require consistency, namely that $\hat{y}=y$, we get a canonical estimate for $\hat{y}$. First, this bootstrap may ease the specification of the regression models, second, it is a canonical way for interpolation (LoRP can't distinguish between r that are identical on D), and third, many standard regressors (kNN, Kernel, LBFR) are self-consistent in the sense that they are canonical.

Explicit expressions for kNN on a grid. In order to get more insight into LoRP, a case that allows an analytic solution is useful. For k nearest neighbors classification with $\boldsymbol{x}$ lying on a hypercube of the regular grid $\mathcal{X}=\mathbb{Z}^d$ one can derive explicit expressions for the loss rank as a function of k, n, and d. For $n\gg k\gg 3^d$, the penalty $-\frac{1}{2}\log\det S$ is proportional to $\text{tr}M$ with proportionality constant decreasing from about 3.2 for $d=1$ to 1.5 for $d\to\infty$.

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).

General additive loss. Linear LoRP $\hat{\boldsymbol{y}}=M(\boldsymbol{x})\boldsymbol{y}$ of Section 3 can easily be generalized from quadratic to ρ-norm $\text{Loss}_M(\boldsymbol{y}|\boldsymbol{x})=||\boldsymbol{y}-\hat{\boldsymbol{y}}||_\rho^p$ (any p). For $\alpha=0$, $\boldsymbol{y}^\top S_0\boldsymbol{y}$ in (9) becomes $||\boldsymbol{y}-\hat{\boldsymbol{y}}||_\rho^2$ and v_ρ the volume of the unit d-dimensional ρ-norm "ball". Useful expressions for general additive $\text{Loss}_N=\sum_i h(y_i-\hat{y}_i)$ can also be derived. 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. In reality, though, having to specify the loss function can be a nuisance. We could interpret the regularized loss (5) as a class of loss functions parameterized by α, and $\text{argmin}_\alpha\text{LR}_r^a$ as a loss function optimization or selection. This suggests to choose in general the loss function that has minimal loss rank. This leads to sensible results if the considered class of loss functions is not too large (e.g. all ρ-norm losses in the previous paragraph). So LoRP can be used not only for model selection, but also for loss function selection.

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.

[BFG96] Z. Bai, M. Fahey, and G. Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1–2):71–89, 1996.

[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.

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

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

[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.