Neural Network Interest Group

Título/*Title*:

PDF Estimation Methods

Autor(es)/*Author(s)*:

Luís M. Silva

Relatório Técnico/*Technical Report* No. 7 /2006

FEUP/INEB, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

Título/*Title*:

PDF Estimation Methods

Autor(es)/*Author(s)*:

Luís M. Silva

Relatório Técnico/*Technical Report* No. 7 /2006

Publicado por/*Published by*: NNIG. http://paginas.fe.up.pt/~nnig/

Contents

1 Introduction **5**

2 Projection pursuit density estimation **7**

2.1 The optimization procedure 9

2.1.1 How to determine the updating vector α_m^j 10

2.2 The log likelihood ratio statistic 10

2.2.1 Feature removal 12

3 Quasi-Newton method **15**

3.1 The Algorithm 17

3.2 Support Vector Method for Multivariate density estimation . 18

3.3 Questions and future work? 20

Chapter 1

Introduction

Pdf estimation is a very important task in many areas. For example, in classification, we might estimate class densities to further obtain the *a posteriori* probabilities of each class. Usually, the data density is not known and all we have is and i.i.d. sample $x_1, \ldots, x_l$ that came from a certain density $p(x)$. Parametric methods assume an *apriori* distribution for the data and the parameters are estimated, for example, with maximum likelihood or Bayesian inference methods. These are higly restrictive methods and solutions are not always obtainable [1]. Non-parametric methods are more flexible, because they do not assume any distribution and the density is mainly obtained from the data itself. Histograms, KNN, and kernel (Parzen's windows) based methods are well known non parametric density estimates. The latter, being the most used one, suggests an estimate of the form

$$\hat{p}(x) = \frac{1}{l} \sum_{i=1}^{l} K_h(x - x_i) \tag{1.1}$$

where $K_h(x - x_i)$ is a smooth function designated *kernel* with property $\int K_h(x - x_i)dx = 1$; h is a parameter that controls the width of the window and consequently the smoothness of the solution. Parzen's method is assured to converge to $p(x)$ at a fast asymptotic rate given some conditions on h and K_h [1]. As we can see from (1.1), the Parzen estimate has a model with high complexity: one kernel function for each data point and the free parameter h (the width of the kernel) has to be set properly with, in principle, no *apriori* knowledge. Also these methods suffer from curse of dimensionality when the problem to be solved has a large input space. This is mostly because data in higher dimensions is very sparse and largest neighborhoods are needed (large h in Parzen's estimate) to get sufficient counts or else the estimate will be biased.

Here we study two different approaches of density estimation intended to overcome the performance of usual methods in high dimensional problems. The first one was proposed by Friedman *et al.* [3] and is based in the projection pursuit methodology [4]. The second one, proposed by Vapnik *et al.* [7, 8], uses the SVM methodology.

Chapter 2

Projection pursuit density estimation

In 1974, Friedman *et al.* [4], proposed the method of projection pursuit (PP). It uses the data to find low dimensional projections that provide the most revealing views of the full-dimensional data. PP intends to discover the nonlinear effects like clustering (existence of different classes) that are not captured by the covariance structure. Ten years later, the method of projection pursuit was used to extend the classical methods of density estimation to the multidimensional cases with the only need of univariate estimates [3]. In 1987, Friedman [2] proposed new algorithms and a new statistical index to find the projection directions in order to improve the original method. He designated this new methodology by *Exploratory projection pursuit* (EPP) and a corresponding density estimation procedure was developed and incorporated in the same paper [2]. The two density estimation procedures were thought to be equivalent until the corrections made by Zhu [9]. In his paper, Zhu [9] shows the non-equivalence of the two procedures is proved and it is also shown that the density estimate model of EPP is cumbersome. In this sense, it is here adopted the framework proposed by the original projection pursuit density estimation procedure.

The projection pursuit density estimate is given by[1] [3]

$$p_M(\mathbf{x}) = p_0(\mathbf{x}) \prod_{m=1}^{M} f_m(\theta_m \cdot \mathbf{x}) \qquad \mathbf{x} \in \mathbb{R}^p$$

where

- p_0 is an initial estimate for the multivariate density
- θ_m is a direction in $\mathbb{R}^p$ (unit vector)
- f_m is an univariate function

Here, $\theta_m \cdot \mathbf{x} = \sum_{i=1}^{p} \theta_{im} x_i$ is a linear combination of the original variables, thus obtaining a projection of each data point in the space spanned by θ_m.

The choice of the initial density p_0 is left to the user and his *a priori* knowledge of the data, but a common choice is the Gaussian density with parameters (μ and Σ) estimated from the sample. The procedure has to estimate directions θ_m and build $f_m(\theta_m \cdot \mathbf{x})$ (the latter designated by the authors as *augmenting functions*). This is done in a recursive manner

$$p_m(\mathbf{x}) = p_{m-1}(\mathbf{x}) f_m(\theta_m \cdot \mathbf{x}) \qquad m = 1, \ldots, M \tag{2.1}$$

At each step m of the procedure a direction θ_m and corresponding $f_m(\theta_m \cdot \mathbf{x})$ are estimated in order to maximize the goodness of fit of $p_m(x)$.The measure of goodness of fit used in [3] is the cross-entropy term of the Kulback-Leibler distance

$$CE_m = \int \log p_m(\mathbf{x}) p(\mathbf{x}) dx$$

It is easy to see from (2.1) that maximizing CE_m is equivalent to maximize

$$W_m = \int \log f_m(\theta_m \cdot \mathbf{x}) p(\mathbf{x}) dx$$

It can be shown [3] that for every fixed θ_m, W_m is maximized when

$$f_m(\theta_m \cdot \mathbf{x}) = \frac{p^{\,\theta_m}(\theta_m \cdot \mathbf{x})}{p_{m-1}^{\theta_m}(\theta_m \cdot \mathbf{x})} \tag{2.2}$$

[1] Where · means dot product

where p^{θ_m} and $p_{m-1}^{\theta_m}$ are estimates along the one-dimensional subspace spanned by θ_m of the data marginal density and the current model marginal density, respectively. Basically, we are dividing out the marginal of the current model and replacing it with the one estimated from the data. If this is done in all directions we probably get the true density [9]. Hence, we have to solve at each step m the optimization procedure

$$\begin{aligned} &\max_{\theta_m} \; W_m = \int \log f_m(\theta_m \cdot \mathbf{x}) p(\mathbf{x}) dx \\ &\text{s.t.} \int p_m(\mathbf{x}) d\mathbf{x} = 1 \end{aligned}$$

In practice we only have and i.i.d. sample $\mathbf{x}_1, \ldots, \mathbf{x}_l$ and W_m has to be estimated by the natural way

$$\hat{W}_m = \frac{1}{l} \sum_{i=1}^{l} \log f_m(\theta_m \cdot \mathbf{x}_i)$$

The above procedure is taken only for a few steps to obtain the directions that achieve the best fit.

In order to find each optimal direction θ_m, we have to estimate the marginal densities in (2.2). Recall that these are one dimensional densities. Friedman *et al.* [3] proposal is to estimate p^{θ_m} by

$$\hat{p}^{\theta_m}(\theta_m \cdot \mathbf{x}) = \frac{1}{2hN} \sum_{i=1}^{N} I(\theta_m \cdot \mathbf{x} - h \leq \theta_m \cdot \mathbf{x}_i \leq \theta_m \cdot \mathbf{x} + h)$$

where $I(t)$ is the indicator function. Now, for the current model marginal density $p_{m-1}^{\theta_m}$ the estimate is given by

$$\hat{p}_{m-1}^{\theta_m}(\theta_m \cdot \mathbf{x}) = \frac{1}{2hN_m} \sum_{j=1}^{N_m} I(\theta_m \cdot \mathbf{x} - h \leq \theta_m \cdot \mathbf{y}_j \leq \theta_m \cdot \mathbf{x} + h)$$

where $\mathbf{y}_1, \ldots, \mathbf{y}_{N_m}$ is a Monte Carlo sample generated with density $p_{m-1}(x)$. Hence, we will get an estimate

$$\hat{f}_m(\theta_m \cdot \mathbf{x}) = \frac{N_m \sum_{i=1}^{N} I(\theta_m \cdot \mathbf{x} - h \leq \theta_m \cdot \mathbf{x}_i \leq \theta_M \cdot \mathbf{x} + h)}{N \sum_{j=1}^{N_s} I(\theta_m \cdot \mathbf{x} - h \leq \theta_m \cdot \mathbf{y}_j \leq \theta_m \cdot \mathbf{x} + h)}$$

To stabilize the denominator, h is chosen to always include exactly αN_m Monte Carlo observations [3], getting

$$\hat{f}_m(\theta_m \cdot \mathbf{x}) = \frac{1}{\alpha N} \sum_{i=1}^{N} I(\theta_m \cdot \mathbf{x} - h \leq \theta_m \cdot \mathbf{x}_i \leq \theta_m \cdot \mathbf{x} + h)$$

Note that this is the proposal from Friedman *et al.* [3] and we can get some modifications here. Instead of these histogram estimates we can use traditional Parzen estimates (that we know of it's good performance in one dimensional cases) to get better one dimensional estimates

$$\begin{aligned} \hat{p}^{\,\theta_m}(\theta_m \cdot \mathbf{x}) &= \frac{1}{lh} \sum_{i=1}^{l} K\left(\frac{\theta_m \cdot \mathbf{x} - \theta_m \cdot \mathbf{x}_i}{h}\right) \\ \hat{p}^{\theta_m}_{m-1}(\theta_m \cdot \mathbf{x}) &= \frac{1}{N_m h} \sum_{j=1}^{N_m} K\left(\frac{\theta_m \cdot \mathbf{x} - \theta_m \cdot \mathbf{y}_j}{h}\right) \end{aligned}$$

where K is a kernel function with bandwidth h.

In practice all we need is to get the set of values $\{\theta_m \cdot \mathbf{x}_i, f_m(\theta_m \cdot \mathbf{x}_i)\}_{i=1,\ldots,l}$ and use an approximation of $f_m(\theta_m \cdot \mathbf{x})$ given by a cubic spline function [3].

2.1 The optimization procedure

Leaving, for now, possible restrictions to the optimization problem, we have to

$$\max\ \hat{W}_m(\theta_m) = \frac{1}{l} \sum_{i=1}^{l} \log \frac{\hat{p}^{\,\theta_m}(\theta_m \cdot \mathbf{x}_i)}{\hat{p}^{\theta_m}_{m-1}(\theta_m \cdot \mathbf{x}_i)}$$

to obtain the direction θ_m. My proposal is to do it iteratively like

1. set step $j = 1$
2. having the estimate θ^j_m of θ_m
3. calculate the estimates $\hat{p}^{\,\theta^j_m}$ and $\hat{p}^{\theta^j_m}_{m-1}$ (the latter using Monte Carlo sampling or other equivalent sampling procedure)
4. calculate $\hat{W}_m(\theta^j_m)$

5. update the projection direction in order to maximize $\hat{W}_m$

$$\theta_m^j + \alpha_m^j = \alpha_m^{j+1} \tag{2.3}$$

6. set step $j = j + 1$ and get back to 2 if a stopping criterion is not reached.

2.1.1 How to determine the updating vector α_m^j

At each step i we have to determine an updating vector α_m^j (2.3). For example, we could use steepest ascent in the surface defined by $\hat{W}_m$. In this case we have

$$\alpha_m^j = \eta \nabla \hat{W}_m^j$$

where η controls the rate (speed) of convergence. We can write

$$\hat{W}_m^j = \frac{1}{l} \sum_{i=1}^{l} \left[\log \hat{p}^{\,\theta_m^j}(\theta_m^j \cdot \mathbf{x}_i) - \log \hat{p}_{m-1}^{\theta_m^j}(\theta_m^j \cdot \mathbf{x}_i) \right]$$

and, thus, we have

$$\begin{aligned} \nabla \hat{W}_m^j &= \left[\frac{\partial \hat{W}_m^j}{\partial \theta_{m1}^j}, \ldots, \frac{\partial \hat{W}_m^j}{\partial \theta_{md}^j} \right] \\ &= \frac{1}{l} \sum_{i=1}^{l} \left[\frac{\left(\hat{p}^{\,\theta_m^j}(\theta_m^j \cdot \mathbf{x}_i) \right)'}{\hat{p}^{\,\theta_m^j}(\theta_m^j \cdot \mathbf{x}_i)} - \frac{\left(\hat{p}_{m-1}^{\theta_m^j}(\theta_m^j \cdot \mathbf{x}_i) \right)'}{\hat{p}_{m-1}^{\theta_m^j}(\theta_m^j \cdot \mathbf{x}_i)} \right] \mathbf{x}_i \end{aligned}$$

From these equations we realize the need of estimating the derivatives of the marginal densities. After having the densities themselves their derivatives can be estimated using standard numerical tools like finite diferences.

2.2 The log likelihood ratio statistic

The density estimation procedure above doesn't take account to the particular problem of estimating class densities from a set of samples. Of course, if the patterns come with class information (with a target vector), we could estimate separately each class density using PPDE. However, Zhu and Hastie [10] proposed an explicit method to estimate class densities based on the log likelihood ratio statistic. The paper proposes a new index to find discriminant directions (those that allow the distinction between classes). It is shown that the method is capable to deal with more general problems like **xor**, when the class centroids coincide, and perform better than Fisher's LDA; it is also shown that LDA is a special case of their method.

The generalized log-likelihood ratio statistic is given by

$$LR(\theta) = \log \frac{\max_{p_k} \prod_{k=1}^{K} \prod_{\mathbf{x}_j \in C_k} p_k^{\theta}(\theta \cdot \mathbf{x}_j)}{\max_{p_k = p} \prod_{k=1}^{K} \prod_{\mathbf{x}_j \in C_k} p^{\,\theta}(\theta \cdot \mathbf{x}_j)} \tag{2.4}$$

It can be shown that the criterion used in the LDA framework is a special case of $LR(\theta)$, when $p_k(\mathbf{x})$ is taken or estimated by $N(\mu_k, \Sigma)$ for all k. This means that $LR(\theta)$ can handle with more general problems, where the information is not simply contained in the class centroids. Equation (2.4) is associated with the resolution of an hypotheses test of the form

$$\begin{aligned} H_0 &: \; p_1 = p_2 = \ldots = p_K \\ H_1 &: \; \text{at least one } p_k \text{ differs} \end{aligned}$$

For arbitrary class densities, the aim is to find directions θ that maximize

$$LR(\theta) = \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in C_k} \log \hat{p}_k^{\theta}(\theta \cdot \mathbf{x}_j) - \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in C_k} \log \hat{p}^{\,\theta}(\theta \cdot \mathbf{x}_j) \tag{2.5}$$

where $\hat{p}_k$ is the MLE of p_k and $\hat{p}$ is the MLE of p. It is important to note that we only need to restrict our search to unit vectors, $||\theta|| = 1$ [9].

This index is incorporated and adapted in the PPDE methodology (substituting the cross-entropy term) to rule the search of projection directions, giving explicit ways of estimating the class densities. In this case $LR(\theta)$ is given by

$$\theta_m = argmax_{\theta} \log \frac{\prod_{k=1}^{K} \prod_{\mathbf{x}_j \in C_k} p_{m-1,k}(\mathbf{x}_j) f_{mk}(\theta \cdot \mathbf{x}_j)}{\prod_{k=1}^{K} \prod_{\mathbf{x}_j \in C_k} p_{m-1}(\mathbf{x}_j) f_m(\theta \cdot \mathbf{x}_j)} \tag{2.6}$$

and the class models by

$$p_{M,k}(\mathbf{x}) = p_0(\mathbf{x}) \prod_{m=1}^{M} f_{mk}(\theta \cdot \mathbf{x})$$

As this method doesn't assume a parametric form for p_k, $LR(\theta)$ has to be maximized numerically. The proposal is to use Newton's method (or quasi-Newton) because we can obtain explicit expressions for the gradient and Hessian matrix of $LR(\theta)$. All that is needed is to have estimates for marginal densities and corresponding derivatives. The author's have used parametric models and Locfit. We can use more flexible models by estimating the univariate densities with kernel methods.

With straightforward calculations we can see that equation (2.6) simply amounts to

$$\theta_m = argmax_\theta \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in C_k} [\log f_{mk}(\theta \cdot \mathbf{x}_j) - \log f_m(\theta \cdot \mathbf{x}_j)] \tag{2.7}$$

where

$$\begin{aligned} f_{mk}(\theta \cdot \mathbf{x}_j) &= \frac{p_k^{(\theta)}(\theta \cdot \mathbf{x}_j)}{p_{m-1,k}^{(\theta)}(\theta \cdot \mathbf{x}_j)} \\ f_m(\theta \cdot \mathbf{x}_j) &= \frac{p^{(\theta)}(\theta \cdot \mathbf{x}_j)}{p_{m-1}^{(\theta)}(\theta \cdot \mathbf{x}_j)} \end{aligned}$$

We can also see that

$$\sum_{k=1}^{K} \sum_{\mathbf{x}_j \in C_k} \log f_m(\theta \cdot \mathbf{x}_j) = \sum_{j=1}^{l} \log f_m(\theta \cdot \mathbf{x}_j) \tag{2.8}$$

Hence, we can see that the procedure is very similar to PPDE, but here, information about classes (that we have from the targets associated to each pattern) is incorporated in (2.7). Note the cross entropy term in (2.7) and (2.8).

To obtain more discriminat projections we need a procedure that avoids finding already found directions. This can be achieved in two ways. First using the usual orthogonalization procedure, i.e, ensuring that the actual direction found θ_m is othogonal to the previous ones, θ_k $k < m$. However this poses the problem of which metric to choose [10]. A preferred procedure is the feature removal strategy of exploratory projection pursuit. This transforms the data in a way that there is no class distinction in direction θ_m, i.e. $p_k^\theta = q$ $\forall k$, while keeping unchanged all other directions. Hence, the procedure can continue without the risk of finding any of the previous directions.

2.2.1 Feature removal

The transformation applied to the data is defined as

$$\mathbf{x}^{'} = h(\mathbf{x}) = \mathbf{A}^{-1}t(\mathbf{A}\mathbf{x})$$

where $\mathbf{A}$ is an orthogonal rotation matrix such that

$$\mathbf{z} = \mathbf{A}\mathbf{x} = \begin{pmatrix} \theta \cdot \mathbf{x} \\ \mathbf{A}^{*}\mathbf{x} \end{pmatrix}$$

and t is given by

$$t(z_j) = \begin{cases} \gamma(z_j) & j = 1 \\ z_j & j > 1 \end{cases}$$

$\gamma(.)$ is a monotonic transformation that removes the distinction between classes in direction θ. It is defined by

$$\gamma(z_1) = Q^{-1}(F_k(\theta \cdot \mathbf{x}))$$

for each class k, and Q^{-1} is the cdf corresponding to the common density function q and F_k is the marginal cdf of $\theta \cdot \mathbf{x}$ for class k. Hence

$$\mathbf{x}^{'} = h(\mathbf{x}) = \mathbf{A}^{-1} \begin{pmatrix} Q^{-1}(F_k(\theta \cdot \mathbf{x})) \\ \mathbf{A}^{*}\mathbf{x} \end{pmatrix}$$

Chapter 3

Quasi-Newton method

To optimize a function F, Newton's method uses the information of the function's curvature that is provided by the Hessian matrix H. Quasi-Newton methods obtain an aproximation to H, without explicit calculations, aliviating some computational effort.

We have

$$\begin{aligned} F(\theta_k + s_k) &= F(\theta_k) + \underbrace{g(\theta_k)}_{g_k} s_k + \frac{1}{2} s_k^T H_k s_k + \ldots \\ \underbrace{g(\theta_k + s_k)}_{g_{k+1}} &= g_k + H_k s_k + \ldots \\ s_k^T (g_{k+1} - g_k) &\approx s_k^T H_k s_k \qquad (3.1) \end{aligned}$$

where, $g_k = \nabla F(\theta_k)$ and $H_k = \nabla^2 F(\theta_k)$, the Hessian matrix at θ_k. In quasi-newton methods we have, at each iteration k, an approximation to H_k given by B_k. The search direction p_k is then obtained from the linear system [5]

$$B_k p_k = -g_k \qquad (3.2)$$

Usually, $B_0 = I$ (the identity matrix) and we need an updating formula for B like

$$B_{k+1} = B_k + U_k$$

where U_k is a low rank update matrix. Define

$$\begin{aligned} s_k &= \theta_{k+1} - \theta_k = \alpha_k p_k \\ y_k &= g_{k+1} - g_k \end{aligned}$$

where α_k is the step length (in direction p_k) obtained from a line search method. Based on (3.1), the required condition for B_{k+1} to approximate the curvature of F along s_k is [5, 6]

$$B_{k+1} s_k = y_k \tag{3.3}$$

Equation (3.3) is known as *quasi-newton condition.* It is desirable that each approximation B_k has some additional properties encountered on Hessian matrices, like symmetry and, in the case of strong minima, positive definitess. The Broyden one-parameter family of updates gives two-rank update formulas for the approximate Hessian

$$B_{k+1}^{\phi} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi_k \left(s_k^T B_k s_k \right) w_k w_k^T$$

where $\phi_k \in [0, 1]$. From practical and some theoretical research, it is believed that the most effective update is given when $\phi_k = 0$, leading to the update formula of Broyden-Fletcher-Goldfarb-Shanno (BFGS)

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

If p_k is obtained from (3.2) we have $B_k s_k = -\alpha_k g_k$ and the BFGS formula becomes

$$B_{k+1} = B_k + \frac{g_k g_k^T}{g_k^T p_k} + \frac{y_k y_k^T}{\alpha_k y_k^T p_k} \tag{3.4}$$

These Broyden family formulas guarantee that B_{k+1} satisfies the quasi-Newton condition (3.3) and is symmetric. To guarantee the hereditary positive definitess property in the BFGS update, that is, if B_k is positive definite then B_{k+1} is positive definite, the following condition has to be satisfied [5, 6]

$$y_k^T s_k > 0$$

It is important to note that things can be simplified in the estimation of p_k from (3.2). As B_k is positive definite, we can perform a Cholesky factorization

$$B_k = L_k L_k^T$$

Here, L_k is a lower triangular matrix and the search direction p_k is simply computed by forward and backward substitution.

$$\begin{aligned} L_k z &= -g_k \\ L_k^T p_k &= z \end{aligned}$$

It is also possible to derive update formulas for the Cholesky factors, i.e, the Cholesky factors for B_{k+1} can be obtained by an update to the ones of B_k. This eliminates the need of determining a factorization at each step k.

Instead of computing the approximate Hessian matrix at each iteration, it is possible to compute the approximate inverse Hessian, and avoid the problem of solving (3.2) (see [5]). However, due to rounding errors this procedure can loose the positive definitess in the Hessian approximations and the above method is, thus, preferable.

3.1 The Algorithm

1. Input θ_0, B_0 and termination criteria

2. For any k, obtain the solution of the system $B_k p_k = -g_k$. This is done taking the Cholesky factorization of B_k.

3. Compute a step size α_k (e.g. by line search on $F(\theta_k + \alpha_k p_k)$) and set $\theta_{k+1} = \theta_k + \alpha_k p_k$

4. Compute the updated matrix B_{k+1} using BFGS

$$B_{k+1} = B_k + \frac{g_k g_k^T}{g_k^T p_k} + \frac{y_k y_k^T}{\alpha_k y_k^T p_k}$$

5. If a termination criteria is not met, set $k = k + 1$ and get back to 2. Note that it is possible to avoid calculating the Cholesky factorization of the updated approximate Hessian B_{k+1}. Just make an update of the factors.

3.2 Support Vector Method for Multivariate density estimation

In [7, 8], Vapnik *et al.* propose a pdf estimation method based in the SVM solution to inverse ill-posed problems. The SVM method has no free parameters and finds a consistent and sparse solution.

The idea is to search a solution of the integral equation

$$\int_{-\infty}^{x} p(t)dt = F(x) \tag{3.5}$$

where, $F(x)$ is the probability distribution function. The procedure starts from and i.i.d. sample $x_1, \ldots, x_l$ and a solution to (3.5) is found in a set $p(t, \alpha)$ with $\alpha \in \Lambda$. In fact, we do not have $F(x)$ but we can use an estimate given by the empirical distribution function $F_l(x)$. We thus obtain a problem of the form

$$Ap = F \tag{3.6}$$

where A is a linear operator and F will be substituted by F_l. This is an ill-posed problem. Note that it is known that $F_l \rightarrow F$ and the distribution of the supremum error between $F(x)$ and $F_l(x)$ is given by the Kolmogorov-Smirnov distribution [7, 8].

Several methods to solve these kind of problems were proposed in the 1960s and use theory of regularization. A functional $\Omega(p)$ is introduced (conditions on $\Omega(p)$ can be found in [7] page 235) in the problem to control the smoothness of the final solution. This final solution, p_l, will be a tradeoff between $\Omega(p)$ e $||Ap - F||$. The SVM procedure, uses the method proposed by Phillips

$$\min_p \Omega(p) \qquad s.t. \qquad ||Ap - F_l|| < \epsilon_l \qquad \epsilon_l > 0,\ \epsilon_l \rightarrow 0 \tag{3.7}$$

The first problem encountered is in the choice of ϵ_l. Morozov proposes the *residual method*, where ϵ_l should be chosen to define a solution p_l for which the equality holds

$$||Ap - F_l|| = ||F(x) - F_l(x)|| = \sigma_l \tag{3.8}$$

where σ_l is the known accuracy of approximation of $F(x)$ by $F_l(x)$. For pdf estimation it is possible to get a good estimate of σ_l (......)

The integral equation (3.5) is solved in a set of functions belonging to a Reproducing Kernel Hilbert Space (RKHS) and using the functional

$$\Omega(p) = ||p||_H^2 \tag{3.9}$$

To define the RKHS we need

1. An Hilbert space H
2. A positive definite kernel $K(x, y)$
3. An inner product $(f, g)_H$ with the *reproducing property*

$$(p(x), K(x, y))_H = p(y) \; \forall p \in H \tag{3.10}$$

Recall that an Hilbert space is a complete inner product space.

In this case we define the Hilbert space of functions

$$f(x, c) = \sum_{i=1}^{\infty} c_i \phi_i(x) \tag{3.11}$$

and the kernel

$$K(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) \tag{3.12}$$

where λ_i and $\phi_i(x)$ are the eigenvalues and eigenfunctions of $K(x, y)$. Finally, with (3.11), (3.12) and the inner product given by

$$(f(x, c), f(x, d)) = \sum_{i=1}^{\infty} \frac{c_i d_i}{\lambda_i} \tag{3.13}$$

we have defined the RKHS. The property that $K(x, y)$ is positive definite come from its expansion in eigenvalues and eigenfunctions. It is also easy to show that the reproducing property (3.10) is present.

The aim is to solve the integral equation in a RKHS with a solution that satisfies (3.8)

$$\min \Omega(p) = (p, p)_H \qquad s.t. \qquad \left| F_l(x) - \int_{-\infty}^{x} p(t)dt \right|_{x=x_i} = \sigma_l \qquad 1 \leq i \leq l \tag{3.14}$$

The method searches for a solution of the form

$$p(t) = \sum_{i=1}^{l} \beta_i K_{\gamma_l}(x_i, t) \tag{3.15}$$

Hence, the problem can be written in the form [7]

$$\begin{aligned} \min \Omega(p,p) &= \sum_{i=1}^{l} \beta_i K_{\gamma_l}(x_i, t) && (3.16) \\ s.t. & \left| F_l(x) - \sum_{j=1}^{l} \beta_j \int_{-\infty}^{x} K_{\gamma_l}(x_j, t)dt \right|_{x=x_i} = \sigma_l \qquad 1 \leq i \leq l \end{aligned}$$

By Vapnik [7] we have in (3.16) an optimization problem closely related to the SV regression problem with an ϵ_l-insensitive zone, which can be solved with the standard SVM technique. Mostly of the β_i will be zero and the x_i values corresponding to nonzero β_i are called support vectors. To obtain a solution as a mixture of densities, we choose a kernel that is a density and establish the constraints $\beta_i \geq 0$ and $\sum_{i=1}^{l} \beta_i = 1$.

Vapnik call γ_l and admissible value if for this value there exists a solution of the optimization problem (3.16), i.e, the solution satisfies (3.8). There exists a non-empty admissible set $\gamma_{min} \leq \gamma_l \leq \gamma_{max}$ [7, 8].

(...)

To improve the performance of the estimation procedure in high dimensional problems, the authors introduce a new type of kernel functions that change their form upon the distance to their nearest neighbours. As every distribution function that has a density is continuous, they first start to introduce a continuous empirical distribution function (recall that the usual one is discontinuous) given by

$$F_l^1(x) = \begin{cases} \frac{k}{l} + \frac{1}{l}\frac{x - x_k - \tau_k/2}{\tau_k} & x \in [x_k - \tau_k/2, x_k + \tau_k/2) \\ \frac{k}{l} & x \in [x_k, x_{k+1}) \text{ and } x \notin [x_k - \tau_k/2, x_k + \tau_k/2) \end{cases} \tag{3.17}$$

where x_k is the k-th data point and τ_k the distance between x_k and it's nearest neighbour.

The final problem to be solved by the SVM approach is given by

$$\begin{aligned}
\min \Omega(p,p) &= \sum_{i,j=1}^{l} \beta_i \beta_j M(x_i, x_j) & (3.18)\\
s.t. & \left| F_l^1(x) - \sum_{j=1}^{l} \beta_j \int_{-\infty}^{x} L(x_j, t) dt \right|_{x=x_i} = \sigma_l \qquad 1 \le i \le l\\
& \beta_i \ge 0 \qquad i = 1, \ldots, l\\
& \sum_{i=1}^{l} \beta_i = 1
\end{aligned}$$

with final solution given by

$$p(t) = \sum_{i=1}^{l} \beta_i L(x_i, t) \tag{3.19}$$

Exact expressions for $M(x_i, x_j)$ and $L(x_j, t)$ depending on K_{γ_l} can be seen in [8]; in the appendix of the same reference we have analytic expressions for different choices of K_{γ_l}.

3.3 Questions and future work?

- In projection pursuit, how does the initial choice $p_0(x)$ influences the final result.
- If we separate the data in classes, we probably loose the clustering structure. Will PPDE be a worthless method? Answer can be found in [?].
- comparison of PPDE and SVM in simulated data
- comparison of PPDE and SVM for classification tasks, using simulated and real data
- use variable kernel's proposed by Vapnik in Parzen's method

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, UK, 1995.

[2] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266, 1987.

[3] J. H. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

[4] J. H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C23:881–890, 1974.

[5] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* Academic Press, 1981.

[6] Stephen G. Nash and Ariela Sofer. *Linear and Nonlinear Programming.* McGraw-Hill, 1996.

[7] V. N. Vapnik. *The Nature of Statistical Learning Theory (Stat's for Eng. and Inf. Sci.).* Springer Verlag, 1999.

[8] V. N. Vapnik and S. Mukherjee. Support vector method for multivariate density estimation.

[9] M. Zhu. On the forward and backward algorithms of projection pursuit. *The Annals od Statistics*, 32(1):233–244, 2004.

[10] M. Zhu and T. Hastie. Feature extraction for nonparametric discriminant analysis. *Journal of Computational and Graphical Statistics*, 12(1):101–120, 2003.