ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 147-152

Pruning kernel density estimators

Olivier Fambon and Christian Jutten

TIRF-INPG, 46 Av. Felix Viallet, F-38000 Grenoble
E-mail: fambon@tirf.inpg.fr chris@tirf.inpg.fr

Abstract: We present a variation of Hassibi & Stork's Optimal Brain Surgeon [3] in the case of Radial Basis Function Networks that perform density estimation. The theoretical framework of "constrained minimisation of a second order expansion of the error" is adapted and presented in details. Practical issues are raised, and a few experiments are carried out in the 1-D case.

1 Background and motivation

The classical technique of kernel density estimation, applied on a set of n points drawn from an unknown true density, consists in putting a kernel centered at every data point and then tune the width factor in some optimal way so as to obtain the best estimator [2]. As this may become computationally prohibiting with large databases, some pre-processing can be performed such as vector quantisation (VQ) in order to reduce the size of the model (number of kernels). From now on, what we call points or centers will refer to the relatively small set of centroids that has been obtained by VQ or clustering.

Our goal will be, from a given fixed kernel estimator (same width for each kernel) based on N points, to derive a smaller estimator, based on n points. Thus we will need to provide a method for calculating the new centers from the previous ones, as well as the width factor. We will have to 'move' the centers and tune the width in order to compensate for the deletion of some kernels. We propose to automatically select the centers that will be pruned according to a density criterion.

A typical application of this kind of pruning would be dynamic resource reallocation on a constrained architecture. For example, when a new class appears in a Bayesian classification problem, this method would automatically pick up some kernels in previous estimators, slightly modify them, and make the kernels available for an estimator of the new density. A multi-resolution scheme could also be derived using this approach.

For a fixed kernel estimator, the output function can be written

$$\hat{f}_n(x) = \frac{1}{nh_n^d} \sum_{i=1}^{n} K(\frac{x - C_i}{h_n})$$

where d is the dimension of the input space. h_n is the 'optimal' width for n kernels and can be written

$$h_n = D_{f,K,d} \cdot n^{-1/(d+4)} \quad (1)$$

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 147-152

$D_{f,K,d}$ is a constant depending only on f, the (unknown) density, K, the shape of the kernel, and d, the dimension of the space (see [1]).

We wish to prune the net from N kernels to n kernels, but maintaining the structure of the estimator, that is, keeping the widths the same for every kernel, and keeping a unity integral of the estimator (it is supposed to be a density). This could be achieved if we were able to provide a new set of n independent examples issued from f. One solution would be to calculate the inverse of the repartition function associated with $\hat{f}_N$ which is not an easy task. An other would be to re-run a VQ with n centroids instead of N, either on the original database —this would be computationally expensive— or on the previous result —but would a VQ perform well on a small set of points.

We intend to use an other approach, derived from Optimal Brain Surgeon (OBS) [3], which is a function approximation/optimization scheme.

2 OBS and the kernel density estimator

The idea is to use $\hat{f}_N$, the estimate of f with N kernels as a target, and then move slightly the n selected centers in order to compensate for the loss of $N-n$ kernels. Thus we need to define an error and a way of varying the parameters of the model that will lead to pruning.

2.1 The error

The error is defined as the square of the L_2 norm, so as to allow formal calculations —instead of averaging on a database— but any additive measure will do. Suppose we have a target function f and a parameterised model of f denoted by $\hat{f}_P$ (P is the parameter vector), we define

$$E(P) = \frac{1}{2} \parallel f - \hat{f}_P \parallel^2 = \frac{1}{2} \int_\Omega (f(x) - \hat{f}_P(x))^2 dx \ ,$$

the Mean Integrated Squared Error (MISE, over the whole space Ω) associated with estimator $\hat{f}_P$, thus function of P. We wish to minimise this error by varying the parameters. This variation of P is constrained for some parameters p_i indicating that we wish to prune some kernels and keep the structure of the estimator. Other parameters are free to evolve allowing a constrained minimisation scheme. This is OBS.

2.2 Parameterisation

We now need to define a parameterised expression of our estimators that will allow for pruning via parameter variation. We write

$$\underbrace{\phi(C_i, h, p)}_{P} = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - C_i}{h}) + p \cdot \frac{1}{Nh_N^d} \sum_{i=n+1}^{N} K(\frac{x - C_i^N}{h_N})$$

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 147-152

where C_i, h and p are variable parameters, respectively the new centers, the new width and p is a 'pruning' parameter that is used to simulate the elimination of $N-n$ kernels (last term of ϕ). They form together the parameter vector P. The C_i^N are the old centers from estimator $\hat{f}_N$ and are constant for our purpose. n, used to set the bounds of the summations, is the desired number of kernels and is actually fixed by the user; it is not considered as a varying parameter. Forcing h to take its optimal value h_n, the term $1/nh^d$ in the first part of ϕ can be written $1/D^{d+4} \cdot h^4$ through eq. (1). ϕ can finally be written:

$$\phi(C_i, h, p) = \alpha \cdot h^4 \sum_{i=1}^{n} K(\frac{x - C_i}{h}) + \beta \cdot p \qquad \alpha, \beta \text{ constants}$$

Then, if we set the C_i's to their old values C_i^N, h to h_N and p to 1, we recover $\hat{f}_N$ (call P_N the corresponding parameter vector). And if we set p to 0, and h to h_n, we have a n-kernel net (pruned net) that is obtained after a variation of the parameters of ϕ, i.e the centers C_i of the remaining kernels (first term in ϕ).

2.3 A constrained minimisation: OBS

As in OBS, we will resort to a Taylor expansion of the error. This is written

$$\delta E = \nabla E \cdot \delta P + \frac{1}{2} \delta P^t \cdot H \cdot \delta P + o(\| \delta P \|^2) ,$$

where H is the Hessian matrix of E. This expansion is performed around the minimum of E that is reached for $\hat{f}_N$, that is for $P = P_N$. Derivatives and second derivatives are then taken in P_N, minimum of E. The gradient term of the Taylor expansion is thus exactly zero.

We wish to minimise δE with respect to the parameters, thus causing a variation δP of P. But we require that some pruning is achieved, and this is written as a constraint over δP:

$$\begin{cases} \delta p &= 0 - 1 \\ \delta h &= h_n - h_N . \end{cases}$$

The first constraint indicates that we prune $N-n$ kernels. We have chosen the simplest way to write this, using a 'cancelling' parameter, but this may lead to a non homogeneous parameter vector P. An other way to write this pruning constraint would be to force the width parameter to zero in the pruned kernels.

In OBS, the selection of the parameters that should be pruned is achieved via eq. (2), which would requires in our case a full evaluation of H for every center. Moreover, we wish to prune several kernels ($N-n$) at the same time. We propose to use a density criterion, because it seems natural that compensating for the loss of a kernel in a region where there are many others will be easy (see fig. 2).

The second constraint states that we wish to use the 'optimal' width on our resulting estimator.

Usually, we have the centers, and we define an optimal width parameter. Here, we are have the 'optimal' width factor for n kernels by eq. (1) and we

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 147-152

define the centers. Actually, we have at hand the old value of h, h_N, by the old estimator $\hat{f}_N$. So we can estimate D_f, which provides via eq. (1) an estimate for h_n. This second constraint may also have a desirable side effect: it restricts the space of solutions so that noisy solutions are eliminated. Furthermore, it forces the width factor to be positive, which is essential for the model.

Vector δP has the following structure:

$$\delta P = [\delta h, \delta p, \delta C_1, \ldots, \delta C_n]^t$$

Then, the matrix of constraints can be written

$$\begin{bmatrix} \delta P^t \cdot e_1 & - & (h_n - h_N) \\ \delta P^t \cdot e_2 & - & 1 \end{bmatrix} = 0$$

where e_1 and e_2 denote the canonical column vectors associated with the first and second coordinates. This can be written $M\delta P + B$ with vector $B = [h_N - h_n, -1]$ and matrix $M = [e_1, e_2]^t$. Then, we form a Lagrangian, in the classical way, to handle our constrained minimisation:

$$L(\delta P) = \frac{1}{2}\delta P^t \cdot H \cdot \delta P + \Lambda \cdot (M\delta P + B)$$

where Λ is the unknown Lagrange multiplier $\Lambda = [\lambda_1, \lambda_2]$. Differentiating L with respect to δP and using constraints, we find the constrained minimum and the associated δP

$$Min = \frac{1}{2}B^t A^{-1} B \quad (2)$$

$$\delta P^t = -B^t A^{-1} M H^{-1} \quad (3)$$

where matrix A is a 2×2 matrix defined by $A = MH^{-1}M^t$

There is no proof that matrix A has an inverse nor than H itself has one. This is a frequent problem when using second order methods. A way to ensure the invertibility of H is to apply a regularisation technique by adding a small perturbation εI_n on H, and calculate the inverse of $H + \varepsilon I$. This is a classical way of going around the problem. As long as ε is small enough (we take 10^{-4}), it corresponds to a a good approximation of H in the space of invertible matrices.

3 Calculating the Hessian

It is worth noting that the Hessian has a particular form in our case. As we use the MISE, the Hessian can be written

$$\nabla\nabla E = \int_\Omega \left[(y_P - y_{P_N}) \frac{\partial^2 y_P}{\partial p_i \partial p_j} + \frac{\partial y_P}{\partial p_i} \frac{\partial y_P}{\partial p_j} \right] dx$$

Here, y_P is a shorthand for $\hat{f}_P(x)$ and y_{P_N} stands for $\hat{f}_N(x)$, the original net output. As derivatives are taken in $P = P_N$, it is straightforward that the first

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 147-152

term vanishes. The Hessian is then exactly

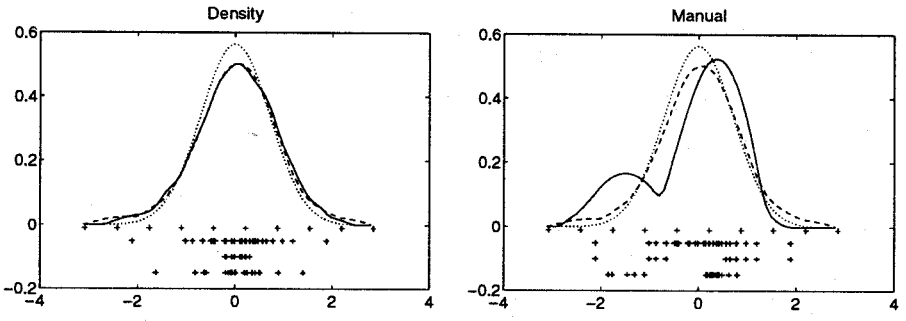
$$\int_{\Omega} \nabla y_P \cdot \nabla y_P^t dx \ .$$

Depending on the choice of the kernel function, it may be possible to have an exact expression of H. It is simple to get an analytic expression of the integrals when using quadric kernels. In the case of Gaussian or hyper-Gaussian kernels, no proper expression can be derived as some integrals cannot be evaluated analytically. Of course, numerical methods could be used to calculate those integrals. For example, a grid can be designed to evaluate the integral by a discrete sum. Note that the number of points needed will become huge as the dimension of the input space increases.

If the grid approach is used, a special inversion scheme can be derived as matrix H is nothing but a sum of matrices of the form $\nabla y \nabla y^t$ (see [3] for the method). This allows for an iterative calculation of H^{-1} that needs only one pass over the database. It is a classical recursive calculation of an inverse that is often used in signal processing.

4 Illustration

We illustrate the method on a toy problem in dimension 1. Note that as the dimension increases, the number of required centers increases also in order to guaranty the quality of the estimator. Then the Hessian rapidly becomes huge. This may be the weak point of the method, as long as no fast/economical Hessian inversion scheme is available.



(a) (b)

Figure 1: true density (dotted), original estimator (dashed), resulting pruned estimator (solid), grid used to calculate the Hessian (first line of crosses), original centers C_i^N (second line), pruned centers/kernels (third line) resulting — moved— centers (last line).

We generate a (good) set of $N = 30$ points drawn from a normal 1-D distribution ($d = 1$, $f = N(0,1)$), and consider that they constitute the centers

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 147-152

C_i^N of our kernel estimator of the normal distribution. We take $D_{f,K,d} = 1.9$, so $h_N = 1.9 \cdot N^{-1/(d+4)} = 0.98$ for we know that the underlying density is Gaussian. In a real situation, we should use the value of h_N provided by the old estimator $\hat{f}_N$, or a proper estimate of $D_{f,K,d}$.

We show on fig. 1 the results obtained after pruning 10 kernels (out of 30) and using a grid of 10 equally spaced points to evaluate the error.The performance is given in terms of relative MISE ($\rho = \| \hat{f}_N - \hat{f}_n \|^2 / \| \hat{f}_N \|^2$): (a) density driven pruning $\rho = 1.710^{-3}$ and (b) manual selection $\rho = 0.1$.

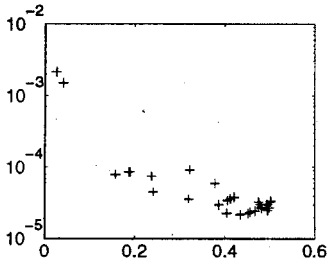


Figure 2: Relative MISE ρ against density $\hat{f}_N$ at kernel center. ρ is obtained after pruning one kernel at a time on the previous example. Each cross represents one pruned kernel.

On fig. 2 we show that the density criterion we used makes sense, at least for smooth and well behaved estimators. Clearly, pruned kernels centered at low density locations degrade more the estimator. But a kernel located at the highest density area will not necessarily be the best one to prune.

5 Conclusion

We presented a theoretical adaptation of Hassibi & Stork's Optimal Brain Surgeon to the case of kernel density estimators. Pruning is driven by a density criterion, and the method aims at producing an 'optimal' estimator, provided the previous one was optimal. Current work deals with a theoretically justified stop /acceptance criterion for pruning. Comparison with direct VQ approach will be carried out soon.

This work has been partly funded by ELENA Nerves 2 Esprit Basic Research Project 6891.

References

[1] P. Comon and C. Jutten et al. ELENA Deliverable R1-A-P, Axis A: Theory. Esprit Basic Research Project 6891, CEC, June 1993.

[2] W. Härdle. *Smoothing Techniques with implementation in S*, chapter 2, Kernel Density Estimators, pages 43–83. Springer-Verlag, 1990.

[3] B. Hassibi and D. Stork. Second order derivatives for network pruning: Optimal Brain Surgeon. In *NIPS 5*, 1993.