1 Support Vector Density Estimation

J. Weston, A. Gammerman, M. O. Stitson, V. Vapnik, V. Vovk, C. Watkins
Royal Holloway, University of London
J.Weston@dcs.rhbnc.ac.uk

We describe how the SV technique of solving linear operator equations can be applied to the problem of density estimation and how this method makes use of a special type of problem-specific regularization. We present a new optimization procedure and set of kernels that guarantee the estimate to be a density (be non-negative everywhere and have an integral of 1). We introduce a dictionary of kernel functions to find approximations using kernels of different widths adaptively. A method of SV regression using square loss is introduced and it is shown how this technique is useful for density estimation. Finally, a way of compressing density estimates from classical kernel based methods is described, and all these algorithms are compared to classical kernel density estimates (Parzen's windows).

1.1 The density estimation problem

We wish to approximate the density function $p(x)$ from data where the corresponding distribution function is

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(t)dt.$$

(If not specified otherwise our densities are with respect to the usual Lebesgue measure.) Finding the required density means solving the linear operator equation[1]

$$\int_{-\infty}^{\infty} \theta(x-t)p(t)dt = F(x), \tag{1.1}$$

1.

$$\theta(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

where instead of knowing the distribution function $F(x)$ we are given the iid (independently and identically distributed) data

$$x_1, \ldots, x_\ell \tag{1.2}$$

generated by F.

The problem of density estimation is known to be ill-posed. "Ill-posed" means that when finding p that satisfies the equality $Ap = F$, where A is a linear operator, we can have large deviations in solution p corresponding to small deviations in F. In our terms, a small change in the distribution function of the continuous random variable X can cause large changes in the derivative, the density function. To solve ill-posed problems, regularization techniques can be used.

1.2 SV method of estimating densities

Using the data (1.2) we construct the empirical distribution function

$$F_\ell(x) = \frac{1}{\ell}\sum_{i=1}^{\ell}\theta(x - x_i)$$

instead of the right hand side of (1.1), which is unknown. We use the SV method to solve the regression problem of approximating the right hand side, using the data

$$(x_1, F_\ell(x_1)), \ldots, (x_\ell, F_\ell(x_\ell)).$$

Applying the SV method of solving linear operator equations Vapnik et al. (1997) (using $y_i = F_\ell(x_i)$), the parameters of the regression function can then be used to express the corresponding density. Regularization is controlled with the parameters ε and C.

This approach can be refined by further control of the regularization Vapnik (1998). For any fixed point x the random value $F_l(x)$ is an unbiased estimate of $F(x)$ and has the standard deviation

$$\sigma = \sqrt{\frac{1}{\ell}F(x)(1 - F(x))}$$

so we can characterize the accuracy of our approximation at the data points with

$$\varepsilon_i = \lambda\sigma_i = \lambda\sqrt{\frac{1}{\ell}F_\ell(x_i)(1 - F_\ell(x_i))}, \tag{1.3}$$

where λ is usually chosen to be 1.

Therefore we consider triples

$$(x_1, F_\ell(x_1), \varepsilon_1), \ldots, (x_l, F_l(x_\ell), \varepsilon_\ell). \tag{1.4}$$

We will use a generalization of the usual support vector regression technique (SVR) to allow the value ε_i to define the loss at the training point x_i , $i = 1, \ldots, \ell$; in the usual SV technique $\varepsilon_1 = \ldots = \varepsilon_\ell$.

In the next section we will review how the SV method is used to solve linear operator equations and then use this technique to construct kernels specifically for density estimation.

1.3 SV density estimation by solving the linear operator equation

To solve the density estimation problem we use the SV method for solving linear operator equations

$$Ap(t) = F(x),$$

where operator A is a linear mapping from a Hilbert space of functions $p(t)$ to a Hilbert space of functions $F(x)$. We solve a regression problem in the image space $(F(x, \mathbf{w}))$ and this solution, which is an expansion on the support vectors, can be used to describe the solution in the pre-image space $(p(t, \mathbf{w}))$. The method is as follows. Choose a set of density functions $p(t, \mathbf{w})$ to solve the problem in the pre-image space that are linear in the flattening space:

$$p(t, \mathbf{w}) = \sum_{r=0}^{\infty} w_r \phi_r(t) = (\mathbf{w} \cdot \Phi(t));$$

that is, $p(t, \mathbf{w})$ are linear combinations of the functions

$$\Phi(t) = (\phi_0(t), \ldots, \phi_n(t), \ldots).$$

Each $p(t, \mathbf{w})$ can be thought of as a hyperplane in this flattening space, where $w = (w_0, \ldots, w_n, \ldots)$ are the coefficients to the hyperplane. The result of the mapping from the pre-image to the image space by the operator A can then be expressed as a linear combination of functions in the image Hilbert space defined thus:

$$F(x, \mathbf{w}) = Ap(t, \mathbf{w}) = \sum_{r=0}^{\infty} w_r \psi_r(x) = (\mathbf{w} \cdot \Psi(x)),$$

where $\Psi(x) = (\psi_0(x), \ldots, \psi_n(x), \ldots)$ and ψ_r is the r^{th} function from our set of functions after the linear operator A has been applied, i.e $\psi_r(x) = A\phi_r(t)$.

The problem of finding the required density (finding the vector $\mathbf{w}$ in the pre-image space) is equivalent to finding the vector of coefficients $\mathbf{w}$ in the image space, where $\mathbf{w}$ is an expansion on the support vectors

$$\mathbf{w} = \sum_{i=1}^{\ell} \beta_i \Psi(x_i),$$

giving the approximation to the desired density

$$p(t, \mathbf{w}) = \sum_{i=1}^{\ell} \beta_i \Psi(x_i) \Phi(t).$$

To find the required density we solve a linear regression problem in the image space by minimizing the same functional we used to solve standard regression problems Vapnik (1995); Vapnik et al. (1997). Instead of directly finding the infinite dimensional vector $\mathbf{w}$ which is equivalent to finding the parameters which describe the density function, we use kernel functions to describe the mapping from input space to the image and pre-image Hilbert spaces.

We use the kernel

$$k(x_i, x_j) = \sum_{r=0}^{\infty} \psi_r(x_i)\psi_r(x_j)$$

to represent the inner product in the image space defined by the set of functions Ψ. We solve the corresponding regression problem in the image space, using the coefficients to define the density function in the pre-image space:

$$p(t, \boldsymbol{\beta}) = \sum_{i=1}^{\ell} \beta_i \mathcal{K}(x_i, t)$$

where $\mathcal{K}$ is the so-called cross kernel:

$$\mathcal{K}(x_i, t) = \sum_{r=0}^{\infty} \psi_r(x_i)\phi_r(t). \tag{1.5}$$

1.4 Spline approximation of a density

We can look for the solution to equation (1.1) in any set of functions where we can construct a corresponding kernel and cross kernel. For example, consider the set of constant splines with infinite number of nodes, similar to Vapnik et al. (1997). That is we approximate the unknown density by the function:

$$p(t) = \int_0^1 g(\tau)\theta(t-\tau)d\tau + a_0$$

(which is assumed to be concentrated on [0,1]) where function $g(\tau)$ and parameter a_0 are to be estimated. We thus define the regression problem in image space

$$\begin{aligned} F(x) &= \int_0^1 g(\tau)[\int_0^x \theta(t-\tau)dt]d\tau + \int_0^x a_0 dt \\ &= \int_0^1 g(\tau)[(x-\tau)_+]d\tau + a_0 x. \end{aligned} \tag{1.6}$$

So the corresponding kernel is

$$\begin{aligned} k(x_i, x_j) &= \int_0^1 (x_i-\tau)_+(x_j-\tau)_+ d\tau + x_i x_j \\ &= (x_i \wedge x_j)^2(x_i \vee x_j) - \frac{1}{2}(x_i \wedge x_j)^3 - \frac{1}{2}(x_i \wedge x_j)^2(x_i \vee x_j) + \frac{1}{3}(x_i \wedge x_j)^3 + x_i x_j \end{aligned} \tag{1.7}$$

where we denoted by $(x_i \wedge x_j)$ the minimum and $(x_i \vee x_j)$ the maximum of two values x_i and x_j . The corresponding cross kernel is (notice that $\frac{d}{dt}(t-\tau)_+ = \theta(t-\tau)$)

$$\mathcal{K}(x,t) = \int_0^1 \theta(t-\tau)(x-\tau)_+ d\tau + x$$

$$= x(x \wedge t) - \frac{(x \wedge t)^2}{2} + x.$$

Using kernel (1.7) and triples (1.4) we can obtain the support vector coefficients $\beta_i = \alpha_i^* - \alpha_i$, only some of which are non-zero. This is achieved by using the standard SV regression approximation Vapnik et al. (1997) with generalized ε-insensitive loss function, by maximizing the quadratic form

SV regression with generalized ε-insensitive loss function

$$W(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}) = -\sum_{i=1}^{\ell} \varepsilon_i(\alpha_i^* + \alpha_i) + \sum_{i=1}^{\ell} y_i(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j)$$

subject to the constraints

$$\sum_{i=1}^{\ell} \alpha_i^* = \sum_{i=1}^{\ell} \alpha_i \tag{1.8}$$

$$0 \le \alpha_i^* \le C, \quad i = 1, \ldots, \ell,$$

$$0 \le \alpha_i \le C, \quad i = 1, \ldots, \ell. \tag{1.9}$$

For density estimation constraint (1.8) can be removed as the threshold b is not used. These coefficients define the approximation to the density

$$p(t) = \sum_{i=1}^{\ell^0} \beta_i^0 \mathcal{K}(x_i^0, t)$$

where x_i^0 are the $\ell^0 \le \ell$ support vectors with corresponding non zero coefficients β_i^0.

1.5 Considering a monotonic set of functions

Unfortunately, the described technique does not guarantee the chosen density will always be positive (recall that a probability is always nonnegative, and the distribution function monotonically increases). This is because the set of functions $F(x, \mathbf{w})$ from which we choose our regression in the image space can contain non-monotonic functions.

We can choose a set of monotonic regression functions and require that the coefficients β_i, $i = 1, \ldots, \ell$, are positive. However, many sets of monotonic functions expressed with Mercer Kernels are too weak in their expressive power to find the desired regression — for example if we choose from the set of polynomials with only positive coefficients.

Using kernels from classical density estimation theory (for example, see **?**)) in the SV method means solving a regression problem using a non-Mercer Kernel. In the next section we consider a slightly different method of SV regression estimation that allows us to use kernels of this form.

1.6 Linear programming (LP) approach to SV regression estimation

In the SV approach, regression estimation problems are solved as a quadratic optimization problem (1.9), giving the approximation

$$F(x) = \sum_{i=1}^{\ell^0} \beta_i^0 k(x_i^0, x).$$

If we choose as our regularizing term the L_1 norm of $\mathbf{w}$ (in the usual approach we choose the L_2 norm) we are only required to solve a linear program Vapnik (1998). In this alternative approach our regularizing term is the sum of the support vector weights. This approach is justified by bounds obtained in the problem of Pattern Recognition that the probability of test error is less than the minimum of three terms, one of which is a function of the number of support vectors.

This gives us the linear program:

$$\min \left(\sum_{i=1}^{\ell} \alpha_i + \sum_{i=1}^{\ell} \alpha_i^* + C \sum_{i=1}^{\ell} \xi_i + C \sum_{i=1}^{\ell} \xi_i^* \right)$$

SV regression with L_1 norm regularization

with constraints

$$y_i - \varepsilon - \xi_i \le \left(\sum_{j=1}^{\ell} (\alpha_j^* - \alpha_j) k(x_i, x_j) \right) + b \le y_i + \varepsilon + \xi_i^*, \quad i = 1, \ldots, \ell,$$

$$\alpha_i \ge 0, \quad \xi_i \ge 0, \quad \alpha_i^* \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, \ell.$$

Minimizing the sum of coefficients is a (convex) approximation to minimizing the number of support vectors. This regularizing term can be seen as a measure of smoothness in input space; a small number of support vectors will mean a less complex decision function. Note that in this approach $k(x, x')$ does not have to satisfy Mercer's condition. We shall use this freedom to construct kernels to estimate densities from a mixture of Gaussian-like shapes. In general we will consider kernels of the following form: $\mathcal{K}(x, x_0)$ is a density function, and $k(x, x_0)$ is its integral, for any fixed x_0.

1.7 Gaussian-like approximation of a density

A common approximation to an unknown density is a mixture of bumps (Gaussian-like shapes). Using the SV method this means approximating the regression in the image space (approximating the unknown distribution function) with a mixture of sigmoidal functions. Considering sigmoids of the form

Sigmoid kernel

$$k(x, x') = \frac{1}{1 + e^{-\gamma(x - x')}} \tag{1.10}$$

the approximation of the density becomes

$$p(x) = \sum_{i=1}^{\ell} \beta_i \mathcal{K}(x_i, x)$$

Gaussian-like cross kernel

where

$$\mathcal{K}(x, x') = -\frac{\gamma}{2 + e^{\gamma(x - x')} + e^{-\gamma(x - x')}}. \tag{1.11}$$

The chosen centres for the bumps are defined by the support vectors, and their heights by the size of their corresponding weights. Note that the kernel (1.10) is non-symmetrical and can only be used with the approach to SV regression described in *section* 1.6.

In fact, we can consider any kernel function from classical density estimation theory **?**) (Uniform, Cosine arch,...) which has a known integral in order to construct both kernel and cross kernel functions.

Typically, these kernels have a width parameter which is used to choose the smoothness of the density estimate. We would like to remove this free parameter by considering a set of functions which contains functions of different widths. This can be achieved by considering a dictionary of kernel functions.

1.8 SV density estimation using a dictionary of kernels

Density estimate with dictionary of kernels

We would like to estimate the density with kernels of varying widths, allowing the technique to choose the best widths at different training points. This can be achieved by considering a dictionary of κ kernel functions giving the decision function

$$p(x) = \sum_{i=1}^{\ell} (\alpha_i^1 \mathcal{K}_1(x_i, x) + \alpha_i^2 \mathcal{K}_2(x_i, x) + \ldots + \alpha_i^\kappa \mathcal{K}_\kappa(x_i, x))$$

where each vector x_i , $i = 1, \ldots, \ell$, has coefficients $\alpha_i^j \geq 0$, $j = 1, \ldots, \kappa$, which are positive to guarantee our density estimate is non-negative everywhere . We then have a corresponding dictionary of κ cross kernels, where $\mathcal{K}_i$ has the width γ_i, for example

$$\gamma_1 = \frac{1}{2}, \quad \gamma_2 = \frac{1}{3}, \ldots$$

As usual we want many of these coefficients to be zero. As the sum of coefficients is 1 (to make a density) the regularization described in *section* 1.6 is not suitable for density estimation. Instead, we choose the regularizer

$$\sum_{i=1}^{\ell} \sum_{n=1}^{\kappa} \frac{1}{\gamma_n} \alpha_i^n$$

to penalize kernels with small width. This results in a linear program. It is possible to choose other regularizers, and it is not yet clear whether there exist more appropriate measures of smoothness.

Generalized LP SV regression with dictionary of kernels

We can thus transform the LP SV regression technique (*section* 1.6) to the following optimization problem:

$$\min\left(\sum_{i=1}^{\ell}\sum_{n=1}^{\kappa}\frac{1}{\gamma_n}\alpha_i^n + C\sum_{i=1}^{\ell}\xi_i + C\sum_{i=1}^{\ell}\xi_i^*\right) \tag{1.12}$$

with constraints

$$y_i - \varepsilon_i - \xi_i \le \sum_{j=1}^{\ell}\sum_{n=1}^{\kappa}\alpha_j^n k_n(x_i, x_j) \le y_i + \varepsilon_i + \xi_i^*, \qquad i = 1, \ldots, \ell,$$

$$\sum_{i=1}^{\ell}\sum_{n=1}^{\kappa}\alpha_i^n = 1, \tag{1.13}$$

$$\alpha_i \ge 0, \quad \xi_i \ge 0, \quad \xi_i^* \ge 0, \qquad i = 1, \ldots, \ell.$$

1.9 One more method of SV density estimation

Square loss SVR with L_1 norm regularization

Densities can also be estimated using square loss instead of absolute loss using a linear regularizer (as introduced in *section* 1.6). This method gives a quadratic optimization problem:

$$\min\left(\sum_{i=1}^{\ell}\left(y_i - \left[\sum_{j=1}^{\ell}(\alpha_j^* - \alpha_j)k(\mathbf{x}_i, \mathbf{x}_j) + b\right]\right)^2 + \lambda\sum_{i=1}^{\ell}(\alpha_i^* + \alpha_i)\right)$$

with constraints

$$\alpha_i \ge 0, \quad \alpha_i^* \ge 0, \quad i = 1, \ldots, \ell.$$

Generalized square loss SVR with dictionary of kernels

The sparsity again comes from the regularizing term. Using this method of regression for density estimation with kernels of different width we obtain:

$$\min\left(\sum_{i=1}^{\ell}(y_i - \sum_{j=1}^{\ell}\sum_{n=1}^{\kappa}\alpha_j^n k_n(\mathbf{x}_i, \mathbf{x}_j))^2 + \lambda\sum_{i=1}^{\ell}\sum_{n=1}^{\kappa}\frac{1}{\gamma_n}\alpha_i^n\right)$$

with constraint (1.13) and the constraints

$$\alpha_i \ge 0, \quad i = 1, \ldots, \ell.$$

This solution cannot take advantage of the special regularization that we gained from the ε-insensitive loss function. However, the optimization problem we obtain is more suitable to training with a large number of data points as we can employ decomposition methods (as in Osuna et al. (1996, 1997)). This becomes feasible due to the simplicity of the constraints.

1.10 Parzen's windows

Classical kernel based density estimates use the decision function

$$p_{est}(\mathbf{x}) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathcal{K}(\mathbf{x},\mathbf{x}_i;\gamma).$$

We choose the kernel

$$\mathcal{K}(\mathbf{x},\mathbf{x}_i;\gamma) = \frac{1}{\gamma^N}\mathcal{K}\Big(\frac{\mathbf{x}-\mathbf{x}_i}{\gamma}\Big), \mathbf{x}\in\mathbf{R}^N$$

where $\mathcal{K}(u)$ is a symmetric unimodal density function. The decision function is an expansion on all ℓ training vectors, rather than just the support vectors. In our experiments we compare density approximations from this classical technique with our techniques.

1.11 Approximating density estimates using SV regression techniques

The support vector approach can also be used to compress the description of density estimates that are generated by some other method, for example the Parzen's windows estimator. Parzen's windows estimation is an expansion on all of the training data. An approximation to this estimate that has a sparse representation (that uses only some vectors in the training set) can be found using a special kind of regression estimation.

Approximating density estimates using SVR

This can be done in the following way: construct the pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)$ where $y_i = p_{est}(\mathbf{x}_i)$, and $\mathbf{x}_i$, $i = 1, \ldots, \ell$, are the original training data. We then approximate this data with a regression function. If we restrict our set of functions to be densities (that are non-negative everywhere and have an integral of 1) we try to find a sparse approximation to the density estimate. Accuracy vs complexity of the approximation is controlled by the regularization. This can be achieved by using the regression techniques described in *section* 1.8 or *section* 1.9, replacing the kernel $k(x, y)$ with the set of functions you wish to approximate with, for example the radial basis function (RBF) kernel (1.11).

1.12 Multi-dimensional density estimation

To estimate multi-dimensional densities the generalization of the SV method is straightforward. We estimate the density $p(\mathbf{x})$ which has the corresponding distribution function

$$F(\mathbf{x}) = P(X \leq \mathbf{x}) = \int_{-\infty}^{x^1}\cdots\int_{-\infty}^{x^N} p(t)dt\ldots dt$$

where $\mathbf{x} = (x^1, \ldots, x^N) \in \mathbf{R}^N$ using the multi-dimensional empirical distribution function

$$F_\ell(\mathbf{x}) = \frac{1}{\ell}\sum_{i=1}^{\ell}\theta(x^1 - x_i^1)\ldots\theta(x^N - x_i^N).$$

Multi-dimensional kernels can be chosen to be tensor products of one dimensional kernels, or other kernels can be chosen directly.

1.13 Experiments

We considered a density generated from a mixture of two normal distributions

$$p(x,\mu,\sigma) = \frac{1}{2\sigma\sqrt{(2\pi)}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} + \frac{1}{2\sqrt{(2\pi)}}\exp\left\{-\frac{x^2}{2}\right\} \tag{1.14}$$

where $\mu = -4$, $\sigma = \frac{1}{2}$. We drew 50 , 75, 100 and 200 examples generated by this distribution, and estimated the density by choosing the best value of parameters for each of the following four techniques:

- Linear Programming SVM with ε-insensitive loss (EL-SVM) (*section* 1.8). Here $\gamma = 1$ (which fixes ε) and C is a free parameter (although it is typically close to ∞).
- Square loss SVM (SL-SVM) (*section* 1.9). This has the free parameter λ.
- Approximating Parzen's windows estimates using SV regression (*section* 1.11). We fixed $C = \infty$ and adjusted the free parameter ε (we do not use different values of ε at different training points in this case).
- Parzen's windows (*section* 1.10), controlling the kernel width γ.

In all techniques we used the RBF kernel (1.11). In all three SV techniques a dictionary of four kernel widths was used: $\gamma_n = \frac{2\pi}{n}$, $n = 1, \ldots, 4$.

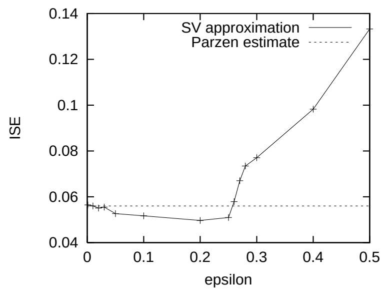
For each method we chose the values of the free parameter(s) which gave the lowest value of ISE (integrated squared error estimated using Simpson's method) given knowledge of the true density. In practice, of course, the true density is unknown and the best parameter(s) cannot be selected; we only use this information to find how close the best case prediction of an estimator can possibly get to the true density.

	EL-SVM		SL-SVM		PE-SVM		Parzen	
pts	SVs	ISE	SVs	ISE	SVs	ISE	SVs	ISE
50	5	0.045	6	0.031	10	0.050	50	0.056
75	7	0.095	6	0.036	13	0.086	75	0.087
100	7	0.105	7	0.064	10	0.079	100	0.091
200	5	0.072	7	0.056	20	0.053	200	0.053

Table 1.1 A comparison of the SV density estimator with ε-insensitive loss (EL-SVM), square loss (SL-SVM), a SV approximation of the Parzen's estimator (PE-SVM) and the Parzen's windows estimator (Parzen). Each estimator was picked with the best possible value of parameters given knowledge of the true density.

The results shown in Table 1.1 indicate that all three SV techniques are competitive with the Parzen's estimator, whilst possessing less complex (sparse) decision functions. EL-SVM performed worst but has the advantage of less parameter selection (only C was chosen, and typically $C = \infty$). PE-SVM approximations of Parzen's estimates obtained significantly reduced numbers of support vectors (as Parzen's windows is an expansion on all the training examples) whilst slightly reducing ISE. This reduction is probably due to the decision function being marginally smoother. Trade off between accuracy and complexity when controlling ε in this method can be seen in Figure 1.1.

SL-SVM performed best of all, providing functions very close to the true density with small numbers of support vectors. It is worth pointing out this is not because the loss function is square loss instead of absolute loss; setting $\varepsilon = 0$ and controlling the free parameter C in EL-SVM gives as good results (or better) than SL-SVM (results not reported here). However, SL-SVM is better at dealing with a large number of data points (one can employ more efficient decomposition techniques.)



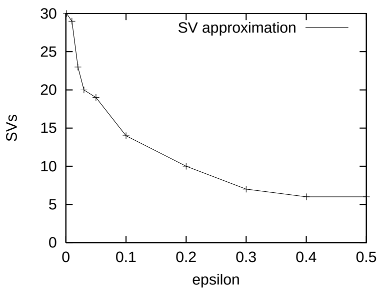


Figure 1.1 Approximating Parzen estimates with the SV method (PE-SVM) gives the same or lower ISE using only a small number of support vectors. Here ε is plotted against ISE (left) and the number of support vectors (right).

1.14 Conclusions and further research

We have described a new SV technique for estimating multi-dimensional densities. Although this results in different optimization problems to normal SVMs, the algorithms have the common feature of possessing sparse decision functions.

We have shown how to solve the density estimation problem as a linear operator equation with a special type of problem-specific regularization using the ε-insensitive loss function.

The integrals of powerful density estimation kernels are not typically symmetrical kernel functions. We show two methods of using non-Mercer kernels with SVM: one with an ε-insensitive loss function and one with square loss. We show how both of these methods can use a dictionary of kernel functions to choose the density estimate from a wide class of functions (for example a mixture of Gaussian-like shapes of different widths).

The results suggest that these methods could obtain good results in real applications. Multi-dimensional problems remain untested; however results in other domains suggest the SV techniques will work well in the multi-dimensional case (the SV kernel regression method is well suited to multi-dimensional problems). Particularly, the square loss method of SV regression is expected to work well as decomposition techniques can be used for dealing with large numbers of training points while using non-Mercer kernels.

Further research lies in the following areas: assessing the performance of the square loss SVM, using dictionaries of kernels in ordinary regression problems and obtaining results in real density estimation applications, in particular in the multi-dimensional case.

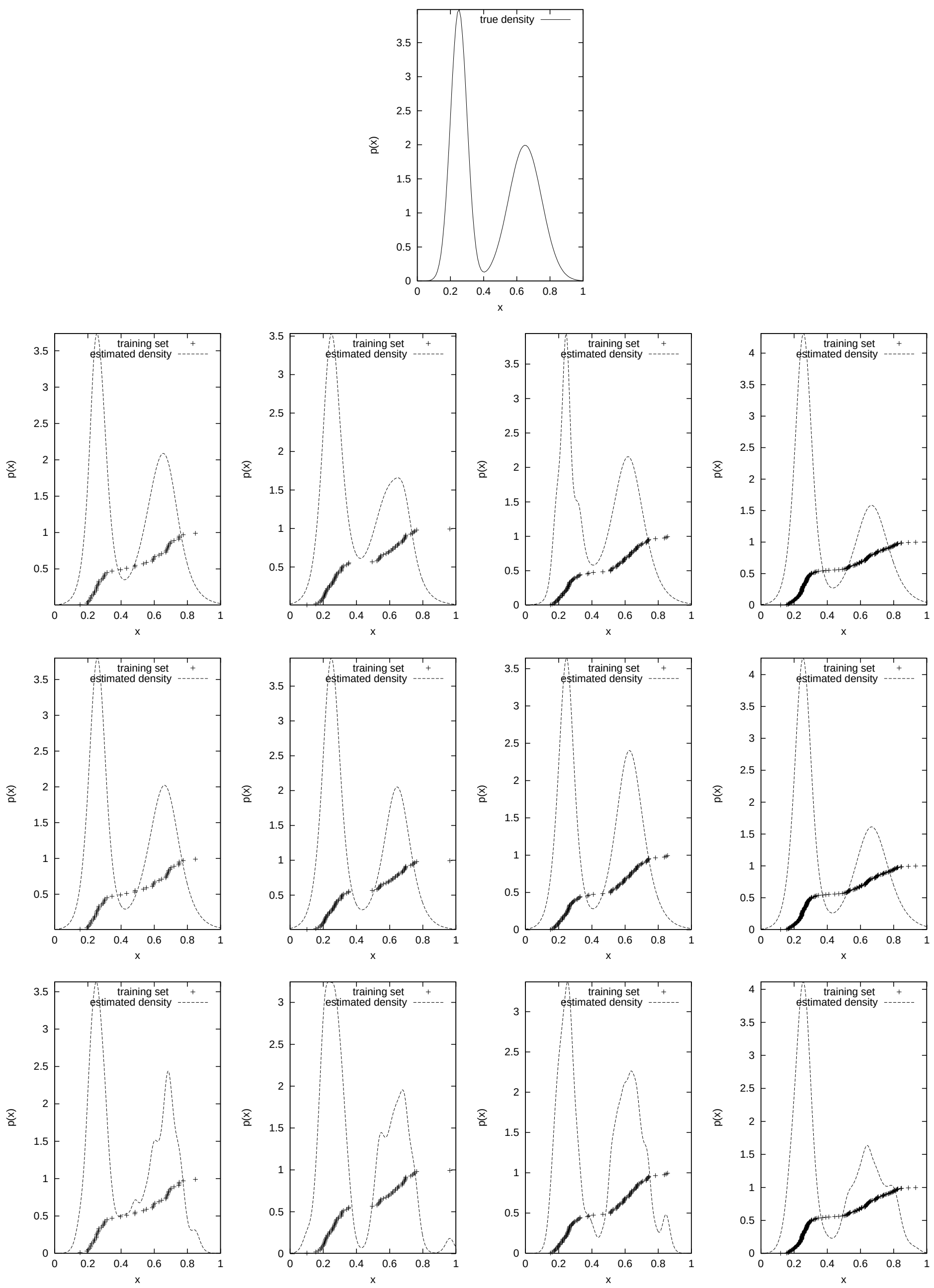


Figure 1.2 The true density (top row) and density estimates by the methods ε-insensitive loss SVM (EL-SVM) (row two), square loss SVM (SL-SVM) (row three), and Parzen's windows estimate (bottom row) generated from (from left to right) 50 , 75 , 100 and 200 points.

References

E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. A.I. Memo (in press), MIT A. I. Lab., 1996.

E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276 – 285, New York, 1997. IEEE.

V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.

V. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998. forthcoming.

V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 281–287, Cambridge, MA, 1997. MIT Press.