Mean Field Theory for Density Estimation Using Support Vector Machines

Refaat M Mohamed and Aly A Farag
Computer Vision and Image Processing Laboratory
University of Louisville, Louisville, KY, 40292
{refaat, farag}@cvip.uofl.edu
www.cvip.uofl.edu

Abstract – *This paper presents a novel algorithm for density estimation. This algorithm is based on the support vector machines (SVM) approach and the Mean Field (MF) theory. The SVM decomposes the parameters of the density estimation problem into a quadratic optimization form. This form is suitable for optimization using Mean Field theory. The new algorithm selects the weights of the mixture of kernels used in the SVM estimate more accurately and faster than traditional quadratic programming algorithms. The performance of the proposed algorithms is illustrated using a number of simulated densities. The evaluation shows that the method provides satisfactory results while keeping a reasonable convergence speed.*

Keywords: Density estimation, learning methods, kernel learning, SVM.

1 Introduction

Density estimation is a problem of fundamental importance to all aspects of machine learning and pattern recognition. The probability density function (PDF) of a continuous distribution is estimated from a representative sample drawn from the underlying density. The estimation can be carried out either in a parametric or non-parametric way. When it is reasonable to assume, a priori, a particular functional form for the PDF then the problem reduces to the estimation of the required functional parameters; parametric approach. For estimating arbitrary density functions, finite mixture models [1, 2] are very powerful approaches and they are routinely employed in many practical applications. One can consider a finite mixture model as providing a condensed representation of the data sample in terms of the sufficient statistics of each of the mixture components and their respective mixing weights.

Support Vector Machines; SVM, is one of the non-parametric methods for density estimation. The PDF is estimated as a mixture of functions that represent the training sample. The training sample is projected into a higher dimensional space using a symmetric, semi-definite mapping function; called the kernel function. The PDF corresponding to a specific point is then calculated as a weighted sum of the kernels. The task of finding the mixing parameters is reduced to a quadratic programming problem which can be solved using optimization routines [3].

The size of the quadratic programming problem, raised from the SVM density is the same as the size of the training sample. Hence, despite a number of practical successes, SVM methods have not yet proved themselves as standard tools in machine learning. The reason for this is the difficulty of implementing such systems since solution of a complex quadratic programming problem is required. Despite the fact that the perceptron was invented in the sixties, interest in feed-forward neural networks only took off in the eighties, due largely to a new training algorithm. That is the same for research into SVM which has been hampered by the fact that training requires solving a quadratic programming problem of a large size, [4].

Mean field (MF) methods provide efficient approximations which are able to cope with the complexity of probabilistic data models. They replace the intractable task of computing high dimensional sums and integrals by the tractable problem of solving a system of linear equations [5].

In this paper, we addressed the density estimation problem using the SVM. The MF method is used to approximate the quadratic programming problem raised from the SVM approach. The performance of the proposed method is evaluated using several examples in one and two dimensional spaces.

2 Density estimation: Problem statement

Given a random vector, $\mathbf{x}$, the relation:

$$F(\mathbf{x}) = P(\ \mathbf{x} < x) \tag{1}$$

defines the cumulative probability distribution function (CDF) of the random vector $\mathbf{x}$. The probability density function (PDF), $p(\mathbf{x})$, of the random vector $\mathbf{x}$ is a nonnegative quantity and it is related to the CDF by the relation:

$$F(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{x}')\, d\mathbf{x}' \tag{2}$$

Hence, in order to estimate the probability density function it is needed to obtain a solution for the integral equation:

$$\int_{-\infty}^{\mathbf{x}} p(\mathbf{x}', \alpha)\, d\mathbf{x}' = F(\mathbf{x}) \tag{3}$$

on a given set of densities $p(\mathbf{x},\alpha)$ where, the integration is a vector integration, and α is the set of parameters to be determined.

The estimation problem in (4) can be regarded as solving the linear operator equation:

$$A\,p(\mathbf{x}) = F(\mathbf{x}) \tag{4}$$

where the operator A is a one-to-one mapping for the elements $p(\mathbf{x})$ of the Hilbert space E_1 into elements $F(\mathbf{x})$ of the Hilbert space E_2. Thus, the density estimation problem is reduced to a *regression problem* which is solved in the image space (right hand side of (5)) and this solution can be used to describe the solution in the pre-image space (before the operator A is applied). In this paper, the SVM is used to find a solution for this regression problem.

Unfortunately, the distribution function $F(\mathbf{x})$ is usually unknown. But, if we have a random sample from a distribution, Đ = $\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$, a reasonable approximation for $F(\mathbf{x})$ can be obtained by

$$F_N(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N} \mathrm{I}(\mathbf{x}-\mathbf{x}_k) \tag{5}$$

where, N is the size of the training sample and $I(u)$ is the indicator function. Therefore, the pairs:

$$(\mathbf{x}_1, F_N(\mathbf{x}_1)),...,(\mathbf{x}_N, F_N(\mathbf{x}_N)) \tag{6}$$

are constructed from the training sample and used by the SVM to solve the regression problem (4).

3 SVM Regression

As stated above, the density estimation problem is reduced to a regression problem (with some constraints as will be discussed later). Thus, the SVM regression is explained below.

In the following discussion, the SVM is considered as the maximum a posteriori (MAP) prediction with a Gaussian prior, under the Bayesian framework (Bayes' theorem is used to relate the prior and posterior distributions). The idea is that, instead of defining prior distributions over parameters of learning machine, a Gaussian prior distribution is assumed over the function space on which the machine computes.

The supervised regression learning problem can be stated as follows. Given a training set Đ = $\{(\mathbf{x}_i,t_i)\,|\,i=1,2,...,N\}$, of input vectors $\mathbf{x}_i$ and associated targets t_i, the goal is to infer the output t for a new input $\mathbf{x}$. To characterize the regression problem, a loss function which relates the estimated $y(\mathbf{x})$ and the true target t is defined. The Vapnik's ε-loss function is used in this paper which is defined as:

$$L(t,y(\mathbf{x})) = \begin{cases} 0 & |t-y(\mathbf{x})| \le \varepsilon \\ |t-y(\mathbf{x})| - \varepsilon & |t-y(\mathbf{x})| > \varepsilon \end{cases} \tag{7}$$

where $\varepsilon \ge 0$ is a prespecified constant controlling the noise tolerance.

To construct a Bayesian framework under the assumed loss function (7), an exponential model is employed. In this model, the likelihood $P(t\,|\,y(\mathbf{x}))$ for the probability of the output t at a given point $\mathbf{x}$, providing that the machine output is $y(\mathbf{x})$, is assumed by the following relationship:

$$P(t\,|\,y(\mathbf{x})) = \frac{C}{2(\varepsilon C+1)}\exp\{-C\,L(t,y(\mathbf{x}))\} \tag{8}$$

Since the elements of the training sample are assumed to be statistically independent random vectors, the probabilistic interpretation of SVM regression can be regarded as the following likelihood:

$$P(\text{Đ}|\,y(\mathbf{x})) = \left(\frac{C}{2(\varepsilon C+1)}\right)^N \exp\left\{-C\sum_{i=1}^{N} L(t_i, y(\mathbf{x}_i))\right\} \tag{9}$$

where $\mathbf{y}(\mathbf{x}) = [y(\mathbf{x}_1), y(\mathbf{x}_2),..., y(\mathbf{x}_N)]$.

Since, the SVM is considered as a MAP with a Gaussian prior, the prior probability distribution of the prediction $y(\mathbf{x})$ is assumed as a Gaussian Process, GP. A GP is a stochastic process completely specified by the mean vector and covariance matrix. Thus, for a sample $\mathbf{x} = [\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N]$ the prior probability can be specified as a GP with zero mean (for simplicity) and a covariance function $K(\mathbf{x},\mathbf{x}')$,

$$P(\mathbf{y}(\mathbf{x})) = \frac{1}{\sqrt{2\pi\,\det(K)}}\exp(-\frac{1}{2}\mathbf{y}(\mathbf{x})K_N^{-1}\,\mathbf{y}(\mathbf{x})^T) \tag{10}$$

where $K_N = [K(\mathbf{x}_i,\mathbf{x}_j)]$ is the covariance matrix at the points of $\mathbf{x}$.

From Bayes' theorem:

$$P(\mathbf{y}(\mathbf{x})\,|\text{Đ}) = \frac{P(\text{Đ}|\,\mathbf{y}(\mathbf{x}))P(\mathbf{y}(\mathbf{x}))}{P(\text{Đ})} = M\,\frac{\exp\left\{-C\sum_{i=1}^{N} L(t_i,y(\mathbf{x}_i)) - \frac{1}{2}\mathbf{y}(\mathbf{x})K_N^{-1}\,\mathbf{y}(\mathbf{x})^T\right\}}{\sqrt{2\pi\,\det(K_N)}\quad P(\text{Đ})} \tag{11}$$

where $M = \left(\frac{C}{2(\varepsilon C+1)}\right)^N$ and the normalization constant $P(\text{Đ})$ is given by:

$$P(\text{Ð}) = \frac{M}{\sqrt{2\pi \det(K_N)}} \cdot$$
$$\int \exp\left\{ -C \sum_{i=1}^{N} L(t_i, y(\mathbf{x}_i)) - \frac{1}{2} \mathbf{y}(\mathbf{x}) K_N^{-1} \mathbf{y}(\mathbf{x})^T \right\} d\mathbf{y}(\mathbf{x}) \qquad \dots \quad (12)$$

It is obvious from the above discussion that the MAP estimate of the posterior prediction distribution $P(\mathbf{y}(\mathbf{x}) \mid D)$ is the maximizer of the numerator of (10). Equivalently, the MAP estimate is the minimizer of :

$$\min_{\mathbf{y}(\mathbf{x})} C \sum_{i=1}^{N} L(t_i, y(\mathbf{x}_i)) - \frac{1}{2} \mathbf{y}(\mathbf{x}) K_N^{-1} \mathbf{y}(\mathbf{x})^T \qquad (13)$$

The traditional SVM setting [6], uses quadratic programming optimization by introducing Lagrange variables to solve (13). The size of the optimization problem is the same as the size of the training sample. As the training sample becomes high, the optimization problem becomes invisible (in time considerations) and some time fails.

4 Mean field theory for SVM Regression

An approximation for the optimization problem (13) is needed to facilitate a visible implementation of the SVM algorithm. Recently [5] have introduced an advanced mean field theory approach based on ideas to cope with the Gaussian classification problem. In this paper, we extend this approach to be used in density estimation through SVM regression. From (11), the prediction on a new test input x is given by:

$$\begin{aligned} \langle y(x) \rangle &= \int y(x)\, P(y(x) \mid \text{Ð})\, dy(x) \\ &= \int y(x)\, P(y(x), \mathbf{y}(\mathbf{x}) \mid \text{Ð})\, dy(x) d\mathbf{y}(\mathbf{x}) \qquad (14) \\ &= \frac{M}{\sqrt{2\pi \det(K_N)}} \cdot \int y(x)\, \Lambda\, dy(x) d\mathbf{y}(\mathbf{x}) \end{aligned}$$

where:

$$\Lambda = \frac{\exp\left\{ -C \sum_{i=1}^{N} L(t_i, y(\mathbf{x}_i)) - \frac{1}{2} \mathbf{y}(\mathbf{x}, x) K_{N+1}^{-1} \mathbf{y}(\mathbf{x}, x)^T \right\}}{P(\text{Ð})},$$

$\mathbf{y}(\mathbf{x}, x) = [y(\mathbf{x}_1), y(\mathbf{x}_2), \dots, y(\mathbf{x}_N), y(x)]$, and

$$K_{N+1} = \begin{pmatrix} K_N & K_N(x)^T \\ K_N(x) & K(x,x) \end{pmatrix}.$$

with $K_N(\mathbf{x}) = [\, K(\mathbf{x}_1, x), K(\mathbf{x}_2, x), \dots, K(\mathbf{x}_N, x)]$.

But:

$$y(x) \exp\left\{ \frac{1}{2} \mathbf{y}(\mathbf{x}, x) K_{N+1}^{-1} \mathbf{y}(\mathbf{x}, x)^T \right\} =$$
$$\sum_{i=1}^{N+1} K(x, \mathbf{x}_i) \frac{\partial}{\partial y(\mathbf{x}_i)} \exp\left\{ \frac{1}{2} \mathbf{y}(\mathbf{x}, x) K_{N+1}^{-1} \mathbf{y}(\mathbf{x}, x)^T \right\} \qquad \dots(15)$$

By substituting (15) into (14), then:

$$\begin{aligned} \langle y(x) \rangle &= \frac{M}{P(\text{Ð})} \sum_{i=1}^{N} K(x, \mathbf{x}_i) \int N(\mathbf{y}(\mathbf{x}) \mid \mathbf{0}, K_N) y(x) \cdot \\ &\quad \frac{\partial}{\partial y(\mathbf{x}_i)} \exp\left\{ -C \sum_{j=1}^{N} L(t_j - y(\mathbf{x}_j)) \right\} d\mathbf{y}(\mathbf{x}) \qquad (16) \\ &= \sum_{i=1}^{N} w_i\, K(\mathbf{x}, \mathbf{x}_i) \end{aligned}$$

where w_i is a constant defined as:

$$w_i = \frac{M}{P(\text{Ð})} \int N(\mathbf{y}(\mathbf{x}) \mid \mathbf{0}, K_N) y(x) \cdot$$
$$\frac{\partial}{\partial y(x_i)} \exp\left\{ -C \sum_{j=1}^{N} L(t_j - y(x_j)) \right\} d\mathbf{y}(\mathbf{x}) \qquad \dots(17)$$

To facilitate the calculation of w_i from the training sample, a new distribution for each sample i is defined as follows:

$$P(y(x_i) \mid \overline{\text{Ð}}_{\mathrm{i}}) =$$
$$\frac{\int N(\mathbf{y}(\mathbf{x}) \mid \mathbf{0}, K_N) \exp\left\{ -C \sum_{j \neq i} L(t_j, y(x_j)) \right\} d\mathbf{y}(\overline{\mathbf{x}}_i)}{\int N(\mathbf{y}(\mathbf{x}) \mid \mathbf{0}, K_N) \exp\left\{ -C \sum_{j \neq i} L(t_j, y(x_j)) \right\} d\mathbf{y}(\mathbf{x}_i)} \qquad \dots(18)$$

where $\overline{\text{Ð}}_{\mathrm{i}}$ and $\overline{\mathbf{x}}_i$ are obtained by removing the data pattern $(\mathbf{x}_i, t_i)$ from Ð. It can be noted that $P(y(\mathbf{x}_i) \mid \overline{\text{Ð}}_{\mathrm{i}})$ is the predictive distribution at the “test” point $\mathbf{x}_i$ given the data set $\overline{\text{Ð}}_{\mathrm{i}}$.

Let an average with respect to this predictive distribution is defined by:

$$\langle v \rangle_i = \int v\, P(y(\mathbf{x}_i) \mid \overline{\text{Ð}})\, dy(\mathbf{x}_i) \qquad (19)$$

Then, the coefficient w_i in (14) can be rewritten as:

$$w_i = \frac{\left\langle M \frac{\partial}{\partial y(\mathbf{x}_i)} \exp\{ -C\, L(t_i - y(\mathbf{x}_i)) \} \right\rangle_i}{\left\langle M \exp\{ -C\, L(t_i - y(\mathbf{x}_i)) \} \right\rangle_i} \qquad (20)$$

Thus, the weight coefficients (16) can be obtained by the likelihood variant rates with respect to the local predictive distribution $P(y(\mathbf{x}_i) \mid \overline{\text{Ð}}_{\mathrm{i}})$. Again, a Gaussian approximation is used for the local predictive distribution:

$$P(y(x_i) \mid \overline{\text{Ð}}_{\mathrm{i}}) \approx \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{ -\frac{(y(x_i) - \langle y(x_i) \rangle_i)^2}{2\sigma_i^2} \right\} \qquad (21)$$

with the variance defined as $\sigma_i^2 = \left\langle y(x_i)^2 \right\rangle_i - \left\langle y(x_i) \right\rangle_i^2$.

Inserting (21) into (19) and substituting into (20), the following closed form is obtained for the coefficient:

$$w_i \approx \frac{F(\langle y(x_i)\rangle_i, \sigma_i^2)}{G(\langle y(x_i)\rangle_i, \sigma_i^2)} \tag{22}$$

where:

$$\begin{aligned} F(\langle y(x_i)\rangle_i, \sigma_i^2) = {} & \frac{C}{2}\exp\left\{\frac{C}{2}\left(2\langle y(x_i)\rangle_i - 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ & \left[1 - erf\left[\frac{\langle y(x_i)\rangle_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ & -\frac{C}{2}\exp\left\{\frac{C}{2}\left(-2\langle y(x_i)\rangle_i + 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ & \left[1 - erf\left[\frac{-\langle y(x_i)\rangle_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \end{aligned}$$

…(23)

and

$$\begin{aligned} G(\langle y(x_i)\rangle_i, \sigma_i^2) = {} & \frac{1}{2}erf\left[\frac{t_i - \langle y(x_i)\rangle_i + \varepsilon}{\sqrt{2\sigma_i^2}}\right] \\ & -\frac{1}{2}erf\left[\frac{t_i - \langle y(x_i)\rangle_i - \varepsilon}{\sqrt{2\sigma_i^2}}\right] \\ & +\frac{1}{2}\exp\left\{\frac{C}{2}\left(2\langle y(x_i)\rangle_i - 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ & \left[1 - erf\left[\frac{\langle y(x_i)\rangle_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ & +\frac{1}{2}\exp\left\{\frac{C}{2}\left(-2\langle y(x_i)\rangle_i + 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ & \left[1 - erf\left[\frac{-\langle y(x_i)\rangle_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \end{aligned}$$

…(24)

Equations (22), (23) and (24) are called the mean field equations corresponding to the weight parameters w_i. The local predictive average $\langle y(x_i)\rangle_i$ and variance σ_i^2 in the approximated Gaussian function (21) are needed to obtain the weight coefficients. In [5], the detailed derivation for both $\langle y(x_i)\rangle_i$ and σ_i^2, but the final results are summarized here. The posterior average at x_i is:

$$\langle y(x_i)\rangle = \sum_{j=1}^{N} K(x_i, x_j)\, w_j \tag{25}$$

From [5], the following results are obtained:

$$\langle y(x_i)\rangle_i \approx \langle y(x_i)\rangle - \sigma_i^2 w_i \tag{26}$$

$$\sigma_i^2 \approx \frac{1}{[(\Sigma + K)^{-1}]_{ii}} - \Sigma_i \tag{27}$$

where $\Sigma = diag(\Sigma_1, \Sigma_2, ..., \Sigma_N)$ and

$$\Sigma_i = -\sigma_i^2 - \left(\frac{\partial w_i}{\partial\langle y(x_i)\rangle_i}\right)^{-1}.$$

The expression for $\dfrac{\partial w_i}{\partial\langle y(x_i)\rangle_i}$ can be obtained from (22), (23) and (24) as:

$$\begin{aligned} \frac{\partial w_i}{\partial\langle y(x_i)\rangle_i} & \approx C^2 - w_i^2 \\ & - \frac{w_i\langle y(x_i)\rangle_i + \sigma_i^2\, C^2 \int_{t_i-\varepsilon}^{t_i+\varepsilon} P\left(y(x_i) \,|\, \overline{\text{Đ}}_{\mathrm{i}}\right) dy(x_i)}{\sigma_i^2\, G(\langle y(x_i)\rangle_i, \sigma_i^2)} \\ & \approx C^2 - w_i^2 - \frac{w_i\langle y(x_i)\rangle_i + \sigma_i^2\, C^2 + IG_i}{\sigma_i^2\, G(\langle y(x_i)\rangle_i, \sigma_i^2)} \end{aligned}$$

…(28)

where:

$$IG_i = \frac{1}{2}erf\left[\frac{t_i - \langle y(x_i)\rangle_i + \varepsilon}{\sqrt{2\sigma_i^2}}\right] - \frac{1}{2}erf\left[\frac{t_i - \langle y(x_i)\rangle_i - \varepsilon}{\sqrt{2\sigma_i^2}}\right]$$

5 The kernel function

The covariance matrix K_N assumed in (10) is called the kernel function in SVM terminology. To obtain a solution in the form of a mixture of kernels, we choose a nonnegative kernel which satisfies the following conditions:

1) $K_\gamma(\mathbf{x}, \mathbf{x}_i) = a(\gamma) K\left(\dfrac{\mathbf{x} - \mathbf{x}_i}{\gamma}\right)$, (29)

2) $a(\gamma)\displaystyle\int K\left(\dfrac{\mathbf{x} - \mathbf{x}_i}{\gamma}\right) d\mathbf{x} = 1$, and (30)

3) $K(0) = 1$. (31)

In this paper a Gaussian kernel is used with:

$$K(\mathbf{x}, \mathbf{x}_i) = \frac{1}{\sqrt{2\pi \det(\Lambda)}} \exp(-0.5(\mathbf{x} - \mathbf{x}_i)\Lambda^{-1}(\mathbf{x} - \mathbf{x}_i)^T)$$

…(32)

where Λ is a parameter that needs to be chosen.

6 The algorithm

In this section, the steps for density estimation with the Mean field theory principle applied for SVM optimization will be summarized:

Step 1: Set the training pairs (6) from the training sample Đ.

Step 2: Set the learning rate η and randomly set w_i.

Step 3: Calculate the covariance matrix K and let $\sigma_i^2 = K_{ii}$.

Step 4: Iterate steps 5-6 until getting a convergence in w_i.

Step 5: **“*inner loop*”:** For $i = 1, \ldots, N$ do

5.1) Calculate $\langle y(x_i) \rangle$ from (16)

5.2) Calculate $\langle y(x_i) \rangle_i$ from (26)

5.3) Calculate F_i and G_i from (23) and (24)

5.4) update w_i by:

$$w_i = w_i + \eta\left(\frac{F_i}{G_i} - w_i\right)$$

Step 6: **“*outer loop*”:** For every M iterations of w_i, update σ_i^2 from (27)

The most computationally expensive step in the above algorithm is the inversion of the matrix $\mathrm{K} + \Sigma$ in step 6. So, it is recommended that step 6 “*outer loop*” will iterate less frequently than step 5 “inner loop”. For example, after $M = 10$ iterations of updating w_i, Σ_i and σ_i^2 will be updated.

7 Experimental Simulations

In this section, the performance of the mean field method for the SVM density estimation is studied. The simulation is carried out using different illustrative examples.

7.1 1-D Gaussian example

In this example, a data set of size 100 points from a 1-D zero-mean and unit variance density function is generated. The above algorithm is used to estimate the underlying density function. Fig (1) shows the results for this example.

The results show that the algorithm approximates the function very well compared to traditional methods presented in [6]. Fig (2) shows the effect of increasing the value of the controlling parameter C which distorts the results.

For weights convergence with an absolute difference of 0.0005, this simulation takes 10 iterations from the outer loop for 25 iterations of the inner loop. While, when the inner loop takes 10 iterations, the outer loop takes 15 iterations. This concludes that the algorithm is reasonably fast in both cases.

7.2 1-D Mixture of Gaussians example

In this example, a more challenging example is provided. A data set of size 100 points is generated from a 1-D distribution consists of a mixture of two Gaussians. The parameters for this mixture are shown in table 1.

The above algorithm is used to estimate the underlying density function. Fig (3) shows the results for this example. The figure illustrates that the algorithm performs well even with this difficult example. The control constant

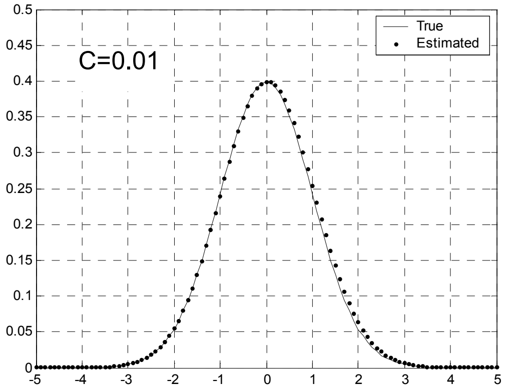

Fig (1) 1-D Gaussian example

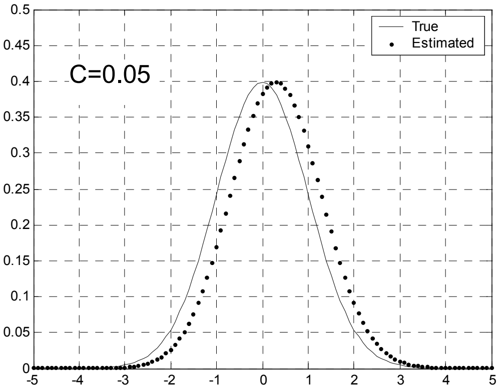

Fig (2) 1-D Gaussian example
Note: Increasing C distorts the estimate

Table 1. Mixture parameters for 1-D mixture density.

Parameters	Reference
μ_1	-1
μ_2	7
σ_1^2	9
σ_2^2	4
α_1	0.6
α_2	0.4

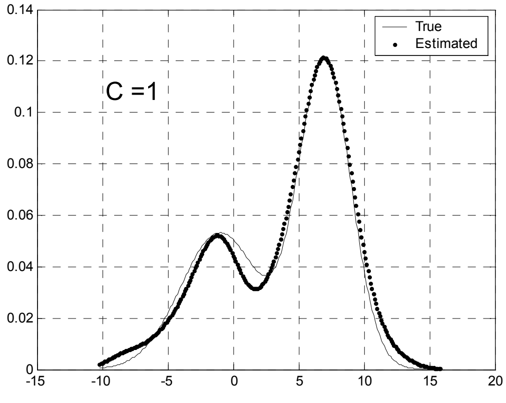

Fig (3) Mixture of Gaussians Example

C is set to 1 in this example and the algorithm takes 1 iteration only from the outer loop for 10 iterations of the inner loop for convergence.

7.3 2-D Gaussian example

In this example, the performance of the proposed algorithm in higher dimensional spaces is demonstrated. A data set of size 100 points from a 2-D Gaussian zero-mean and unit variance density function is generated. The proposed algorithm is applied to estimate the density and Fig (4) illustrates the results. It is clear that the algorithm outperforms traditional methods for the same example presented in [6]. This demonstrates the robustness of the algorithm. It takes 1 iteration of the outer loop only for 10 iterations of the inner loop to converge and the control constant C is set to 0.1 in this case.

8 Conclusions and future work

In this paper we presented a new approach for density estimation. The method uses the Mean Field theory for the implementation of Support Vector Machines density estimation algorithm. The Mean Field theory reduces the quadratic programming problem raised from the SVM formulation to an iterated procedure. This reduction facilitates visible implementation of the SVM method.

The proposed approach is tested using different simulated densities. The results show that the approach is both accurate and fast. However, the simulations also show that the approach is sensitive to the choice of the parameters, e.g. the control constant C, which are chosen empirically.

For the future work, a statistical automatic method for the parameters selection will be developed. Also, more practical examples will be done.

References

[1] Ayman El-Baz and Aly A. Farag. Pararmeter Estimation In Gibbs-Markov Image Models. Proc. 6th International Conference on Information Fusion, Queensland, Australia, pp. 934-942, Jul. 8-11, 2003.

[2] Refaat M Mohamed and Aly A Farag. A New Unsupervised Approach For The Classification Of Multispectral Data. Proc. 6th International Conference on Information Fusion, Queensland, Australia, pp. 951-958, Jul. 8-11, 2003.

[3] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 2nd Edition, 2000.

[4] B. Schoelkop, C. Burges and A. Smola, Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge, Mass., 1999.

[5] Manfred Opper and Ole Winther. Gaussian processes for classification: Mean field algorithms. N. Comp., 12(11):2655--2684, 2000.

[6] Refaat M Mohamed and Aly A. Farag, “Two Sequential Stages Classifier for Multispectral Data,” Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) 2003 workshop on Intelligent Learning, Madison, WS, June 16-22, 2003.

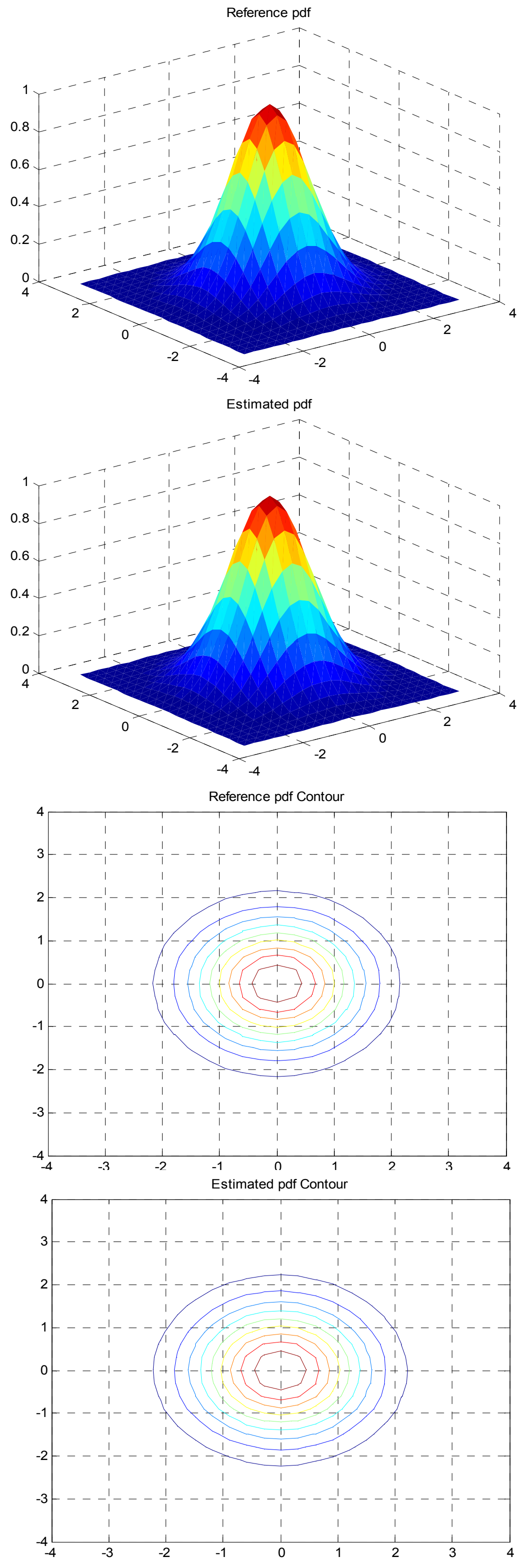

Fig (4) 2-D Gaussian

The contours of the reference and estimated pdf’s are almost the same except for a little difference. This emphasizes that the estimated pdf is so close to the reference pdf.