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 which is based on the support vector machines (SVM) approach and it uses the Mean Field (MF) theory for developing an easy and efficient learning procedure for the SVM. The traditional formulation of the SVM density estimation decomposes the parameters of the problem into a quadratic optimization which can be solved using standard optimization techniques. The proposed algorithm approximates the distribution of the SVM parameters as a Gaussian Process and uses the Mean Field theory to easily estimate these parameters. The new algorithm selects the weights of the mixture of kernels used in the SVM estimate more accurately and faster than traditional quadratic programming-based algorithms. The performance of the proposed algorithm 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, Mean Field Theory.

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 is reduced to the estimation of the required functional parameters; parametric approach. For estimating arbitrary density functions, finite mixture models [1, 2] are gaining much attention as 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 weighting parameters is reduced to a quadratic programming problem in the traditional formulation of the SVM density estimation problem [3].

The size of the raised quadratic programming problem 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 [4]. The reason for this is the difficulty of implementing such systems since the solution of a complex quadratic programming problem is not easy, if it is at all possible. 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.

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]. The MF methods have been introduced for Support Vector Machines regression problems [6]. In this paper, that approach is extended to be used in density estimation problems.

This paper addresses the density estimation problem using the SVM with the assumption that *the parameters* of the SVM are distributed according to a Gaussian Process with a specific parameter set (mean vector and covariance matrix) which can be estimated from the training data set . The MF theory is used to approximate the quadratic programming problem which arises from the SVM approach. This approximation makes the learning process much efficient in both time and accuracy considerations. 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:

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

This work is being supported by Air Force Office for Scientific Research, AFSOR Grant f49620-01-1-0367.

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 $\mathbf{X}$ at the point $\mathbf{x}$ 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 required to obtain a solution for the inverse of 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 parameter set which characterizes the density function $p(\mathbf{x})$.

From another point of view, the estimation problem in Eq.(3) 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 of the Hilbert space E_1 where $p(\mathbf{x})$ is defined into elements of the Hilbert space E_2 where $F(\mathbf{x})$ is defined. But, neither $p(\mathbf{x})$ nor $F(\mathbf{x})$ in Eq.(4) is known. However, from the principles of the probability theory [7], given a random sample $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ from an unknown distribution, a practical estimation for $F(\mathbf{x})$ can be obtained by:

$$F_n(\mathbf{x}) = \frac{1}{n} \sum_{k=1}^{n} \mathrm{I}_{(-\infty, \mathbf{x}]}(\mathbf{x}_k) \tag{5}$$

where, n is the size of the sample and $I_{(-\infty,\, x]}(u)$ is the indicator function which is defined as:

$$I_{(-\infty,\, x]}(u) = \begin{cases} 1 & \text{if } u <= x \\ 0 & \text{else} \end{cases} \tag{6}$$

if both x and u are just scalars (1D data). If both $\mathbf{x}$ and $\mathbf{u}$ are vectors of length d then:

$$I_{(-\infty,\, \mathbf{x}]}(\mathbf{u}) = \prod_{i=1}^{d} I_{(-\infty,\, x_i]}(u_i) \tag{7}$$

This estimator, $F_n(\mathbf{x})$ which is called the empirical distribution function, converges in probability to the original distribution function $F(\mathbf{x})$ [8].Therefore, the training data set $\mathfrak{D}$:

$$\mathfrak{D} = \{(\mathbf{x}_1, F_n(\mathbf{x}_1)), (\mathbf{x}_2, F_n(\mathbf{x}_2)), ..., (\mathbf{x}_n, F_n(\mathbf{x}_n))\} \tag{8}$$

can be constructed from the sample $\mathcal{D}$. Using this training data set, the density estimation problem can be reduced to a *regression problem* which is solved in the image space (right hand side of Eq. (4)) to get an approximation for $F(\mathbf{x})$. This approximation can be then reflected in describing the solution in the pre-image space, before the operator A is applied because the operator A is known. In this paper, the SVM is used to find a solution for the regression problem which represents an approximation for the distribution function $F(\mathbf{x})$ and then the density function $p(\mathbf{x})$ can be easily obtained as will be described later.

3 SVM Regression

The above discussion shows how the supervised density estimation problem can be reduced to a regression problem. In this section, the SVM is presented as a supervised regression tool. 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. In general, the supervised regression learning problem can be stated as follows:

Given a training set $\mathfrak{D} = \{(\mathbf{x}_i, t_i);\ i = 1,2,...,n\}$, of input vectors $\mathbf{x}_i's$ and associated targets $t_i's$, the goal is to infer the target t for a new input vector $\mathbf{x}$. To characterize the regression problem, a loss function which relates the estimated machine output $g(\mathbf{x})$ and the true target t is usually defined. The Vapnik's ε-loss function is used in this paper which is defined as:

$$L(t, g(\mathbf{x})) = \begin{cases} 0 & |t - g(\mathbf{x})| \le \varepsilon \\ |t - g(\mathbf{x})| - \varepsilon & \text{otherwise} \end{cases} \tag{9}$$

where $\varepsilon \ge 0$ is a predefined constant which controls the noise tolerance.

To construct a Bayesian framework under the assumed loss function in Eq.(9), an exponential model is employed. In this model, the likelihood $p(t \mid g(\mathbf{x}))$ for the probability of the output t at a given point $\mathbf{x}$, providing that the machine output is $g(\mathbf{x})$, is assumed by the following relationship:

$$p(t \mid g(\mathbf{x})) = \frac{C}{2(\varepsilon C + 1)} \exp\{-C\, L(t, g(\mathbf{x}))\} \tag{10}$$

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:

$$p(\mathbf{t} \mid \mathbf{g}(\mathcal{D})) = \left(\frac{C}{2(\varepsilon C + 1)}\right)^n \exp\left\{-C \sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i))\right\} \tag{11}$$

where $\mathbf{t} = [t_1, t_2, ..., t_n]$ and $\mathbf{g}(\mathcal{D}) = [g(\mathbf{x}_1), g(\mathbf{x}_2), ..., g(\mathbf{x}_n)]$.

Since, the SVM is considered as a MAP predictor with a Gaussian prior, the prior probability distribution of the prediction $g(\mathbf{x})$ is assumed to be a Gaussian Process, GP. Generally, a GP is a stochastic process which is completely specified by the mean vector and the covariance matrix. Thus, for a sample $\mathcal{D}$, the prior probability can be specified as a GP with a zero mean (for simplicity) and a covariance function $K(\mathbf{x}, \mathbf{x}')$ as:

$$p(\mathbf{g}(\mathcal{D})) = \frac{1}{\sqrt{2\pi \det(K_n)}} \exp(-0.5\mathbf{g}(\mathcal{D})K_n^{-1}\mathbf{g}(\mathcal{D})^T) \quad (12)$$

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

From Bayes' theorem:

$$p(\mathbf{g}(\mathcal{D}) \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{g}(\mathcal{D}))\, p(\mathbf{g}(\mathcal{D}))}{P(\mathcal{D})}$$
$$= M \frac{\exp\left\{ -C\sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i)) - \frac{1}{2}\mathbf{g}(\mathcal{D})K_n^{-1}\mathbf{g}(\mathcal{D})^T \right\}}{\sqrt{2\pi \det(K_n)} \quad P(\mathcal{D})} \quad \ldots (13)$$

where $M = \left(\frac{C}{2(\varepsilon C + 1)} \right)^n$. Let I be defined as:

$$I = \frac{\int \exp\left\{ -C\sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i)) - \frac{1}{2}\mathbf{g}(\mathcal{D})K_n^{-1}\mathbf{g}(\mathcal{D})^T \right\} d\mathbf{g}(\mathcal{D})}{\sqrt{2\pi \det(K_n)}}$$
$$= \int N(\mathbf{g}(\mathcal{D}) \mid \mathbf{0}, K_n) \exp\left\{ -C\sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i)) \right\} d\mathbf{g}(\mathcal{D}) \quad \ldots (14)$$

where $N(\mathbf{g}(\mathcal{D}) \mid \mathbf{0}, K_n)$ is a normal distribution with a zero mean and a covariance matrix K_n. Then, the normalization constant $P(\mathcal{D})$ is given by:

$$P(\mathcal{D}) = MI$$
$$= M \int N(\mathbf{g}(\mathcal{D}) \mid \mathbf{0}, K_n) \exp\left\{ -C\sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i)) \right\} d\mathbf{g}(\mathcal{D}) \quad (15)$$

After some mathematical reduction, it can be shown from the above discussion that the MAP estimate of the posterior prediction distribution $p(\mathbf{g}(\mathcal{D}) \mid \mathcal{D})$ is the one which maximizes the numerator of Eq. (13). Equivalently, the MAP estimate is the one which minimizes:

$$\min_{\mathbf{g}(\mathcal{D})} C\sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i)) + \frac{1}{2}\mathbf{g}(\mathcal{D})K_n^{-1}\mathbf{g}(\mathcal{D})^T \quad (16)$$

The traditional SVM setting [9], uses quadratic programming optimization by introducing Lagrange variables to solve Eq.(16). The size of the optimization problem is the same as the size of the training sample. Thus, if the size of the training sample increases, the optimization problem becomes invisible (in time and accuracy considerations), if it is at all possible [4]. It is this reason which calls for learning algorithms which avoid such quadratic optimization problem. A learning algorithm which satisfies such requirement is presented in the following section.

4 Mean Field Theory for SVM Regression

An approximation for the optimization problem in Eq.(16) is needed to facilitate a visible implementation of the SVM algorithm. Recently, the authors of [5] have introduced an advanced mean field theory approach based on ideas which cope with the Gaussian classification problem. In this paper, we extend this approach to be used in density estimation through SVM regression. From (13), the prediction on a new test input $\mathbf{x}$ is given by:

$$< g(\mathbf{x}) > = \int g(\mathbf{x})\, p(g(\mathbf{x}) \mid \mathcal{D})\, dg(\mathbf{x})$$
$$= \int g(\mathbf{x})\, p(g(\mathbf{x}), \mathbf{g}(\mathcal{D}) \mid \mathcal{D})\, dg(\mathbf{x}) d\mathbf{g}(\mathcal{D}) \quad (17)$$

Substituting from Eq.(13) into Eq.(17) then:

$$< g(\mathbf{x}) > = \frac{M}{\sqrt{2\pi \det(K_n)}} \cdot \int g(\mathbf{x}) \Lambda\, dg(\mathbf{x}) d\mathbf{g}(\mathcal{D}) \quad (18)$$

where:

$$\Lambda = \frac{\exp\left\{ -C\sum_{i=1}^{n} L(t_i, g(\mathbf{x}_i)) - \frac{1}{2}\mathbf{g}(\mathcal{D}, \mathbf{x})K_{n+1}^{-1}\mathbf{g}(\mathcal{D}, \mathbf{x})^T \right\}}{P(\mathcal{D})},$$

$\mathbf{g}(\mathcal{D}, \mathbf{x}) = [g(\mathbf{x}_1), g(\mathbf{x}_2), ..., g(\mathbf{x}_n), g(\mathbf{x})]$, and

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

with $K_n(\mathbf{x}) = [K(\mathbf{x}_1, \mathbf{x}), K(\mathbf{x}_2, \mathbf{x}), ..., K(\mathbf{x}_n, \mathbf{x})]$.

But:

$$g(\mathbf{x}) \exp\left\{ \frac{1}{2}\mathbf{g}(\mathcal{D}, \mathbf{x})K_{n+1}^{-1}\mathbf{g}(\mathcal{D}, \mathbf{x})^T \right\} =$$
$$\sum_{i=1}^{n+1} K(\mathbf{x}, \mathbf{x}_i) \frac{\partial}{\partial g(\mathbf{x}_i)} \exp\left\{ \frac{1}{2}\mathbf{g}(\mathcal{D}, \mathbf{x})K_{n+1}^{-1}\mathbf{g}(\mathcal{D}, \mathbf{x})^T \right\} \quad (19)$$

By substituting (19) into (18), then:

$$< g(\mathbf{x}) > = \frac{M}{P(\mathcal{D})} \sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i) \int N(\mathbf{g}(\mathcal{D}) \mid \mathbf{0}, K_n) g(\mathbf{x}) \cdot$$
$$\frac{\partial}{\partial g(\mathbf{x}_i)} \exp\left\{ -C\sum_{j=1}^{n} L(t_j, g(\mathbf{x}_j)) \right\} d\mathbf{g}(\mathcal{D})$$
$$= \sum_{i=1}^{n} w_i K(\mathbf{x}, \mathbf{x}_i) \quad (20)$$

where w_i is a constant defined as:

$$w_i = \frac{M}{P(\mathcal{D})}\int N(\mathbf{g}(\mathcal{D})\,|\,\mathbf{0}, K_n)\,g(\mathbf{x})\cdot \frac{\partial}{\partial\, g(\mathbf{x}_i)}\exp\left\{-C\sum_{j=1}^{n} L(t_j\,, g(\mathbf{x}_j))\right\} d\mathbf{g}(\mathcal{D}) \quad (21)$$

The learning process suggests that the weights $w_i's$ should be estimated using the training sample. One way to facilitate this estimation is to define a distribution for the expected output corresponding to an instant which is lifted out from the training data set; this idea is known as the "Leave-One-Out" principle. In this principle, one instant $\mathbf{x}_i$ is taken away from the training sample and its corresponding weight w_i is estimated using the new assumed distribution which is defined as:

$$p(g(\mathbf{x}_i)\,|\,\overline{\mathfrak{D}}) = \frac{\int N(\mathbf{g}(\mathcal{D})\,|\,\mathbf{0}, K_n)\exp\left(-C\sum_{j\neq i} L(t_j, g(\mathbf{x}_j))\right) d\mathbf{g}(\overline{\mathcal{D}})}{\int N(\mathbf{g}(\mathcal{D})\,|\,\mathbf{0}, K_n)\exp\left(-C\sum_{j\neq i} L(t_j, g(\mathbf{x}_j))\right) d\mathbf{g}(\mathcal{D})} \quad (22)$$

where $\overline{\mathfrak{D}}$ is obtained by removing the training pattern $(\mathbf{x}_i, t_i)$ from $\mathfrak{D}$, and $\overline{\mathcal{D}}$ is obtained by removing the instant $\mathbf{x}_i$ from $\mathcal{D}$. It can be noted that $p(g(\mathbf{x}_i)\,|\,\overline{\mathfrak{D}})$ is the predictive distribution at the "test" point $\mathbf{x}_i$ given the data set $\overline{\mathfrak{D}}$.

With the above defined predictive distribution $p(g(\mathbf{x}_i)\,|\,\overline{\mathfrak{D}})$, an average (expected) value can be defined as:

$$< \nu >_i = \int \nu\; p(g(\mathbf{x}_i)\,|\,\overline{\mathfrak{D}})\, dg(\mathbf{x}_i) \quad (23)$$

Substituting from Eq.(15) into Eq.(21) for the normalizing constant $P(\mathcal{D})$, then using Eq.(22) and Eq.(23), the coefficient w_i in Eq.(21) can be rewritten as:

$$w_i = \frac{\left\langle M\,\frac{\partial}{\partial\, g(\mathbf{x}_i)}\exp\{-C\,L(t_i, g(\mathbf{x}_i))\}\right\rangle_i}{\left\langle M\,\exp\{-C\,L(t_i\,, g(\mathbf{x}_i))\}\right\rangle_i} \quad (24)$$

Thus, the weight coefficients in Eq.(20) can be obtained by the likelihood variant rates with respect to the local predictive distribution $p(g(\mathbf{x}_i)\,|\,\overline{\mathfrak{D}})$. Again, a Gaussian approximation is used for the local predictive distribution:

$$p(g(\mathbf{x}_i)|\overline{\mathfrak{D}}) \approx \frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left\{-\frac{(g(\mathbf{x}_i) - < g(\mathbf{x}_i) >_i)^2}{2\sigma_i^2}\right\} \quad (25)$$

with the variance defined as:

$$\sigma_i^2 = < g(\mathbf{x}_i)^2 >_i - < g(\mathbf{x}_i) >_i^2 .$$

Inserting (25) into (23) and substituting into (24), the following closed form is obtained for the coefficient:

$$w_i \approx \frac{F_i}{G_i} = \frac{F(< g(\mathbf{x}_i) >_i, \sigma_i^2)}{G(< g(\mathbf{x}_i) >_i, \sigma_i^2)} \quad (26)$$

where:

$$\begin{aligned} F_i = F(< g(\mathbf{x}_i) >_i, \sigma_i^2) = \\ &\frac{C}{2}\exp\left\{\frac{C}{2}\left(2 < g(\mathbf{x}_i) >_i - 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ &\left[1 - erf\left[\frac{< g(\mathbf{x}_i) >_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ &-\frac{C}{2}\exp\left\{\frac{C}{2}\left(-2 < g(\mathbf{x}_i) >_i + 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ &\left[1 - erf\left[\frac{- < g(\mathbf{x}_i) >_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \end{aligned} \quad \ldots(27)$$

and

$$\begin{aligned} G_i = G(< g(\mathbf{x}_i) >_i, \sigma_i^2) = \\ &\frac{1}{2}erf\left[\frac{t_i - < g(\mathbf{x}_i) >_i + \varepsilon}{\sqrt{2\sigma_i^2}}\right] \\ &-\frac{1}{2}erf\left[\frac{t_i - < g(\mathbf{x}_i) >_i - \varepsilon}{\sqrt{2\sigma_i^2}}\right] \\ &+\frac{1}{2}\exp\left\{\frac{C}{2}\left(2 < g(\mathbf{x}_i) >_i - 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ &\left[1 - erf\left[\frac{< g(\mathbf{x}_i) >_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ &+\frac{1}{2}\exp\left\{\frac{C}{2}\left(-2 < g(\mathbf{x}_i) >_i + 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\}\cdot \\ &\left[1 - erf\left[\frac{- < g(\mathbf{x}_i) >_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \end{aligned} \quad \ldots(28)$$

Equations (26), (27) and (28) are called the Mean Field equations corresponding to the weight coefficient w_i. To evaluate the weight coefficient in Eq.(26), it is required to get both the mean (average) $< g(\mathbf{x}_i) >_i$ and variance σ_i^2 of the assumed Gaussian model for the local predictive distribution $p(g(\mathbf{x}_i)\,|\,\overline{\mathfrak{D}})$. The detailed derivation for both $< g(\mathbf{x}_i) >_i$ and σ_i^2 depending on the Mean Field theory can be found in [5]. But here, only the final results are summarized. The posterior average at $\mathbf{x}_i$ is (Eq.(20)):

$$< g(\mathbf{x}_i) > = \sum_{j=1}^{n} w_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (29)$$

From [5], the following results are obtained:

$$< g(\mathbf{x}_i) >_i \approx < g(\mathbf{x}_i) > - \sigma_i^2 w_i \qquad (30)$$

and:

$$\sigma_i^2 \approx \frac{1}{[(\Sigma + K_n)^{-1}]_{ii}} - \Sigma_i \qquad (31)$$

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

$$\Sigma_i = -\sigma_i^2 - \left(\frac{\partial w_i}{\partial < g(\mathbf{x}_i) >_i} \right)^{-1}$$

The expression for $\frac{\partial w_i}{\partial < g(\mathbf{x}_i) >_i}$ can be obtained from Equations (26), (27) and (28) as:

$$\frac{\partial w_i}{\partial < g(\mathbf{x}_i) >_i} \approx C^2 - w_i^2 - \frac{w_i < g(\mathbf{x}_i) >_i + \sigma_i^2 C^2 \int_{t_i-\varepsilon}^{t_i+\varepsilon} P(g(\mathbf{x}_i)|\overline{\mathcal{D}}) dg(\mathbf{x}_i)}{\sigma_i^2 G(< g(\mathbf{x}_i) >_i, \sigma_i^2)}$$
$$\approx C^2 - w_i^2 - \frac{w_i < g(\mathbf{x}_i) >_i + \sigma_i^2 C^2 + IG_i}{\sigma_i^2 G(< g(\mathbf{x}_i) >_i, \sigma_i^2)} \qquad \ldots (32)$$

where:

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

5 Calculation of the Estimated Density Function and Chosen of the Kernel

The above discussion shows how the SVM can be used for approximating the distribution function, CDF, from the sample $\mathcal{D}$. The estimation will be in the form of a weighted sum of the kernel function:

$$F(\mathbf{x}) = \sum_{i=1}^{n} w_i \, K(\mathbf{x}, \mathbf{x}_i) \qquad (33)$$

Consequently, the estimate of the density function will simply be in the form:

$$p(\mathbf{x}) = \sum_{i=1}^{n} w_i \, K'(\mathbf{x}, \mathbf{x}_i) = \sum_{i=1}^{n} w_i \, \mathcal{K}(\mathbf{x}, \mathbf{x}_i) \qquad (34)$$

where $\mathcal{K}(\mathbf{x}, \mathbf{x}_i)$ is the derivative of $K(\mathbf{x}, \mathbf{x}_i)$.

The function $\mathcal{K}(\mathbf{x}, \mathbf{x}_i)$ which is used in the calculation of the density function and also in the covariance matrix K_n in Eq.(12) is called the kernel function in SVM terminology. There are some conditions on the kernel function to obtain a valid density function estimate, see [3]. These conditions are:

1) $\mathcal{K}_\gamma(\mathbf{x}, \mathbf{x}_i) = a(\gamma) \mathcal{K}\left(\frac{\mathbf{x} - \mathbf{x}_i}{\gamma} \right)$,

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

3) $\mathcal{K}(0) = 1$.

In this paper a Gaussian kernel is used with:

$$\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = \exp(-0.5(\mathbf{x} - \mathbf{x}_i)\Lambda^{-1}(\mathbf{x} - \mathbf{x}_i)^T) \qquad (35)$$

where Λ is a parameter which is used to be predefined in this paper.

6 Summary of the Proposed Algorithm

The steps of the proposed algorithm for the density function estimation using the SVM with the Mean field theory principle applied for the learning of the SVM will be summarized below.

Step 1: Generate the training data set $\mathcal{D}$ defined in Eq. (8).

Step 2: Set a learning rate η and randomly initiate w_i's.

Step 3: Calculate the covariance matrix K_n and let $\sigma_i^2 = [K_n]_{ii}$.

Step 4: Iterate steps 5 and 6 until getting a convergence in w_i's.

Step 5: **"*inner loop*":** For $i = 1, 2, ..., n$ do

5.1) Calculate $< g(\mathbf{x}_i) >$ from Eq.(29)

5.2) Calculate $< g(\mathbf{x}_i) >_i$ from Eq. (30)

5.3) Calculate F_i and G_i from Eq. (27) and Eq. (28)

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 Eq. (31)

Step 6: Calculate $p(\mathbf{x})$ from Eq. (34).

The most computationally expensive step in the above algorithm is the inversion of the matrix $\Sigma + \mathrm{K}_n$ 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 traditionally formulated SVM method presented in [9]. 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.

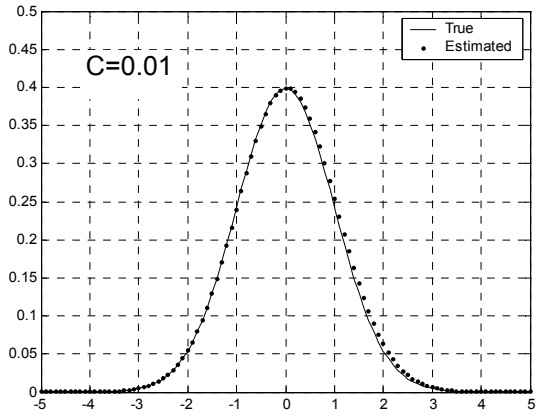

Fig (1) 1-D Gaussian example

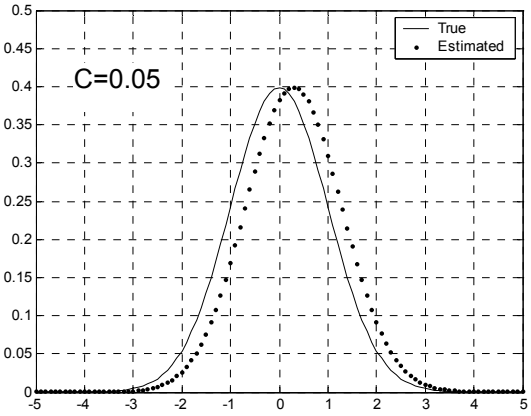

Fig (2) 1-D Gaussian example
Note: Increasing C distorts the estimate

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 estimation example. The figure illustrates that the algorithm performs well even with this little difficult example. The control constant 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.

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

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 variances 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 [9]. 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 which is 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.

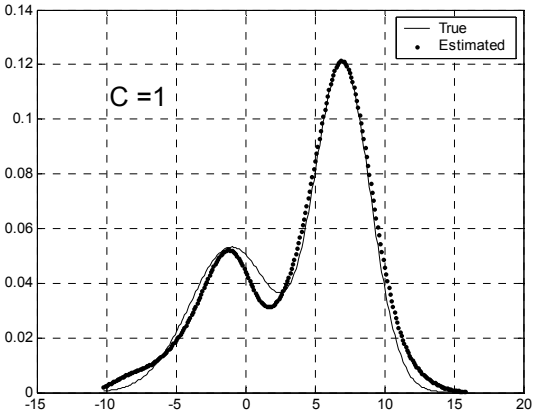

Fig (3) Mixture of Gaussians Example

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. Scholkopf, C. Burges and A. Smola, Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge, MA, 1999.

[5] Manfred Opper and Ole Winther, "Gaussian Processes for Classification: Mean Field Algorithms," Neural Computation, vol. 12, No.11, pp.2655-2684, 2000.

[6] J. B. Gao, S. R. Gunn and C. J. Harris, "Mean Field Method for the Support Vector Machine Regression," Neurocomputing, vol. 50, pp. 391-405, 2003.

[7] John W. Lamperti, Probability-A Survey of the Mathematical Theory. Wiley Series in Probability and Statistics, New York, 1996.

[8] Jun Shao, Mathematical Statistics. Springer-Verlag, New York, 1999.

[9] 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) workshop on Intelligent Learning, Madison, WS, June 16-22, 2003.

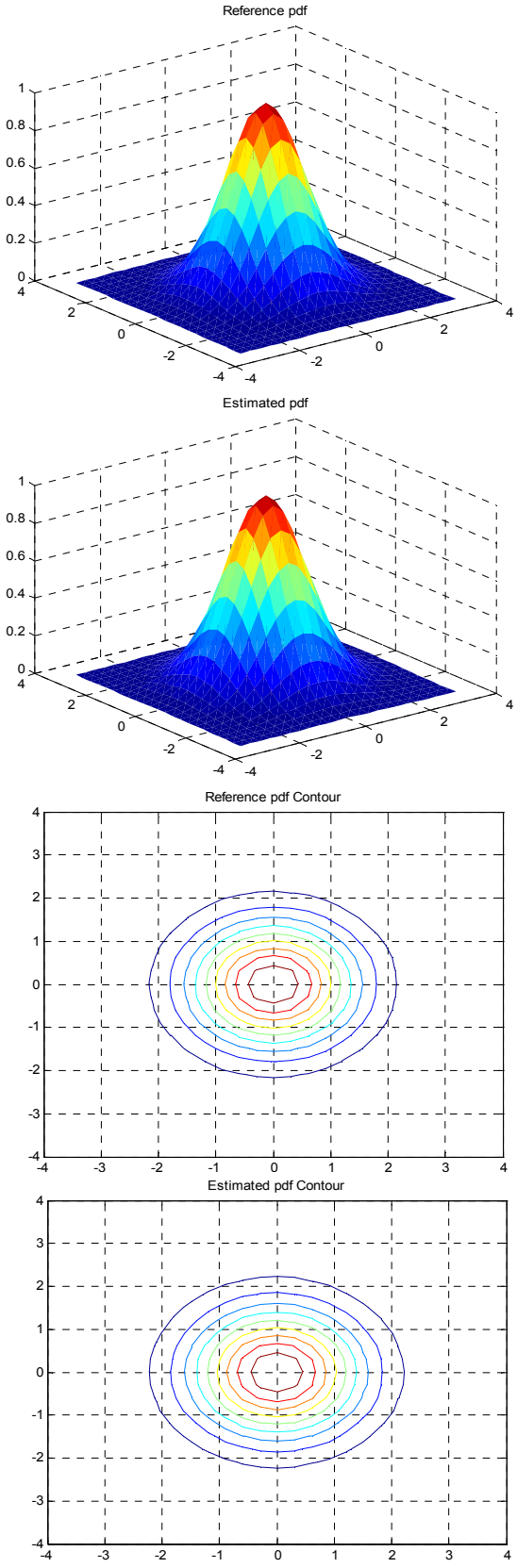

Fig (4) 2-D Gaussian

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.