

- [3] B. Hammer and P. Tino, "Recurrent neural networks with small weights implement definite memory machines," *Neural Computat.*, vol. 15, no. 8, pp. 1897–1929, 2003.
- [4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computat.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [5] J. F. Kolen, "Recurrent networks: State machines or iterated function systems?," in *Proc. Connectionist Models Summer School*, M. C. Mozer, P. Smolensky, D. S. Touretzky, J. L. Elman, and A. S. Weigend, Eds. Hillsdale, NJ, 1994, pp. 203–210.
- [6] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," *Proteins*, vol. 47, pp. 228–235, 2002.
- [7] J. Schmidhuber, F. Gers, and D. Eck, "Learning nonregular languages: A comparison of simple recurrent networks and LSTM," *Neural Computat.*, vol. 14, no. 9, pp. 2039–2041, 2002.
- [8] P. Tino, M. Cernansky, and L. Benuskova, "Markovian architectural bias of recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 6–15, Jan. 2004.
- [9] P. Tino and B. Hammer, "Architectural bias in recurrent neural networks: Fractal analysis," *Neural Computat.*, vol. 15, no. 8, pp. 1931–1957, 2003.

A Spatially Constrained Mixture Model for Image Segmentation

K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris

Abstract—Gaussian mixture models (GMMs) constitute a well-known type of probabilistic neural networks. One of their many successful applications is in image segmentation, where spatially constrained mixture models have been trained using the expectation–maximization (EM) framework. In this letter, we elaborate on this method and propose a new methodology for the M-step of the EM algorithm that is based on a novel constrained optimization formulation. Numerical experiments using simulated images illustrate the superior performance of our method in terms of the attained maximum value of the objective function and segmentation accuracy compared to previous implementations of this approach.

Index Terms—Convex quadratic programming (QP), expectation–maximization (EM), Gaussian mixture model (GMM), image segmentation, Markov random field (MRF).

I. INTRODUCTION

Image segmentation is the process that groups image pixels together based on attributes such as their intensity and spatial location. A variety of different methods have been proposed for image segmentation such as edge-based segmentation, region-based segmentation, pixel labeling, and hybrid techniques [1]–[3]. In this work, we elaborate on a pixel labeling (clustering) technique based on Gaussian mixture models (GMMs) which constitute a well known probabilistic neural-network model [4], [5]. The expectation–maximization (EM) framework constitutes an efficient method for GMM training based on likelihood maximization.

The application of clustering methods to image segmentation has the particular characteristic that spatial information should be taken

into account. That is, apart from the intensity values, the pixel location must also be used to determine the cluster to which each pixel is assigned. Intuitively speaking, in most cases it is desirable to assign the same cluster label to spatially adjacent pixels. The Bayesian framework provides a natural approach to implement these ideas. Following this formulation, a likelihood term which is based exclusively on the data captures the pixel intensity information, while a prior biasing term that uses a Markov random field (MRF) captures the spatial location information. Thus, it is no surprise that most recent image segmentation algorithms follow this paradigm (see, for example, [6] and [7]).

Nevertheless, an inherent difficulty with this formulation is that, due to the introduction of the prior, the M-step of the EM algorithm cannot be implemented using closed-form expressions. For this reason, in [6], a gradient projection (GP) algorithm was proposed to implement the M-step.

In this letter, we propose a novel method to implement the M-step based on a closed-form update equation followed by an efficient projection method. We demonstrate with numerical experiments using the synthetic image data in [7] that the proposed M-step provides a better maximum of the objective function than the GP approach proposed in [6]. In addition, it also yields better segmentation results.

The rest of this letter is organized as follows. In Section II, we describe the probabilistic model for image segmentation. In Section III, we present our improvements to this model. In Section IV, we provide comparative experimental results and finally in Section V, our conclusions and future work.

II. SPATIALLY VARIANT FINITE MIXTURE MODEL

Let x^i denote the observation at the i th pixel of an image ($i = 1, \dots, N$) modeled as i.i.d. The spatially variant finite mixture model (SVFMM) [6] provides a modification of the classical mixture model approach for pixel labeling. The SVFMM assumes a mixture model with K components, each one having its own vector of density parameters θ^j .

According to the SVFMM approach, the probabilities $\pi_j^i = P(j|x^i)$ of the i th pixel belonging to the j th cluster (class label) are considered as additional model parameters that should satisfy the following constraints: $0 \leq \pi_j^i \leq 1$ and $\sum_{j=1}^K \pi_j^i = 1$. Let π^i denote the probability vector for pixel i , $\Pi = \{\pi^1, \dots, \pi^N\}$ the set of probability vectors and $\Theta = \{\theta^1, \dots, \theta^K\}$ the set of component parameters. Then the SVFMM model assumes that the density function $f(x^i|\Pi, \Theta)$ at an observation x^i is given by

$$f(x^i|\Pi, \Theta) = \sum_{j=1}^K \pi_j^i \phi(x^i|\theta^j) \quad (1)$$

where $\phi(x^i|\theta^j)$ is a Gaussian distribution with parameters $\theta^j = \{\mu_j, \sigma_j\}$.

Based on the previous formulation, the parameters of the model can be estimated through likelihood maximization (ML) using the EM algorithm. Since the pixel observations are considered to be independent samples, a significant drawback of the ML approach is that the spatial pixel information is not taken into account [7], [8]. To overcome this difficulty, the SVFMM method considers a maximum *a posteriori* (MAP) approach by introducing a prior distribution for the parameter set Π that takes into account spatial information based on the following Gibbs function [6]–[8]:

$$p(\Pi) = \frac{1}{Z} \exp(-U(\Pi)), \text{ where } U(\Pi) = \beta \sum_{i=1}^N V_{\mathcal{N}_i}(\Pi). \quad (2)$$

Manuscript received May 4, 2004; revised October 18, 2004.
The authors are with the Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece (e-mail: kblekas@cs.uoi.gr; arlycs.uoi.gr; galatsanos@cs.uoi.gr; lagaris@cs.uoi.gr).

Digital Object Identifier 10.1109/TNN.2004.841773

The Z is a normalizing constant, while β is often called regularization parameter. The function $V_{\mathcal{N}_i}(\Pi)$ denotes the clique potential function of the pixel label vectors $\{\pi^m\}$ within the neighborhood \mathcal{N}_i of the i th-pixel and can be computed as follows:

$$V_{\mathcal{N}_i}(\Pi) = \sum_{m \in \mathcal{N}_i} g(u_{i,m}) \quad (3)$$

where the $u_{i,m}$ specifies the distance between the two label vectors π^i and π^m , i.e., $u_{i,m} = |\pi^i - \pi^m|^2 = \sum_{j=1}^K (\pi_j^i - \pi_j^m)^2$. The neighborhood \mathcal{N}_i is the set containing pixels that are horizontally or vertically adjacent to pixel i . Finally, the function $g(u)$ must be nonnegative and monotonically increasing [8]. We have selected $g(u) = (1 + u^{-1})^{-1}$ adopted from [8], while in [6] the identity function $h(u) = u$ was used. The function $g(u)$ penalizes less large values of u and, thus, is more robust to outliers.

Given the prior density, a *posteriori* log-density function can be as follows:

$$p(\Pi, \Theta | X) = \sum_{i=1}^N \log f(x^i | \Pi, \Theta) + \log p(\Pi). \quad (4)$$

The use of the EM algorithm for MAP estimation of the parameters $\{\pi_j^i\}$ and $\{\theta^j\}$ [6] requires that the conditional expectation values z_j^i of the hidden variables are computed at the E-step

$$z_j^i(t) = \frac{\pi_j^{i(t)} \phi(x^i | \theta_j^i(t))}{\sum_{l=1}^K \pi_l^{i(t)} \phi(x^i | \theta_l^i(t))} \quad (5)$$

while in the M-step the maximization of the following log-likelihood corresponding to the complete data set is performed

$$Q_{\text{MAP}}(\Pi, \Theta | \Pi^{(t)} \Theta^{(t)}) = \sum_{i=1}^N \sum_{j=1}^K z_j^i(t) \{\log(\pi_j^i) + \log(\phi(x^i | \theta_j^i))\} - \beta \sum_{i=1}^N \sum_{m \in \mathcal{N}_i} g(u_{i,m}) \quad (6)$$

where t indicates the iteration step. The function Q_{MAP} can be maximized independently for each parameter. This gives the following update equations for parameters of the component densities:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^N z_j^i(t) x^i}{\sum_{i=1}^N z_j^i(t)}, [\sigma_j^2]^{(t+1)} = \frac{\sum_{i=1}^N z_j^i(t) [x^i - \mu_j^{(t+1)}]^2}{\sum_{i=1}^N z_j^i(t)}. \quad (7)$$

However, the maximization of the function Q_{MAP} with respect to the label parameters $\{\pi_j^i\}$ does not provide closed-form update equations. In addition, the maximization procedure must also take into account the constraints $0 \leq \pi_j^i \leq 1$ and $\sum_{j=1}^K \pi_j^i = 1$. Due to this difficulty, a Generalized EM scheme for estimating the label parameters $\{\pi_j^i\}$ was adopted in [6] following the iterative GP method. According to this method the gradient of the MAP function is first projected onto the hyperplane of the constraints. Then a line search is performed along the direction of the projected gradient to find the label parameters $\{\pi_j^i\}$ that maximizes the Q_{MAP} function.

III. PROPOSED TECHNIQUE

In this section, we present the new M-step which we demonstrate experimentally in Section IV that improves the performance of the segmentation algorithm. In order to maximize Q_{MAP} (6) with respect to

π_j^i , we set its derivative equal to zero and obtain the following quadratic expression:

$$4\beta \left[\sum_{m \in \mathcal{N}_i} \dot{g}(u_{i,m}) \right] (\pi_j^{i(t+1)})^2 - 4\beta \left[\sum_{m \in \mathcal{N}_i} \dot{g}(u_{i,m}) \pi_j^m \right] (\pi_j^{i(t+1)}) - z_j^{i(t)} = 0 \quad (8)$$

where $\dot{g}(u)$ indicates the derivative of g . It must be noted that in the previous equation the neighborhood \mathcal{N}_i can include pixels with updated label parameter vectors ($t+1$ step), as well as pixels whose label vectors π^m have not yet been updated (t step).

The two roots of the previous equation are

$$\pi_j^{i(t+1)} = \frac{1}{2 \left[\sum_{m \in \mathcal{N}_i} \dot{g}(u_{i,m}) \right]} \left(\left[\sum_{m \in \mathcal{N}_i} \dot{g}(u_{i,m}) \pi_j^m \right] \pm \left(\left[\sum_{m \in \mathcal{N}_i} \dot{g}(u_{i,m}) \pi_j^m \right]^2 + \frac{1}{\beta} z_j^{i(t)} \left[\sum_{m \in \mathcal{N}_i} \dot{g}(u_{i,m}) \right] \right)^{1/2} \right) \quad (9)$$

and select only the root with the positive sign $+$ since it yields $\pi_j^i \geq 0$. The previous equation provides a straightforward update for the values of label parameters π_j^i of each pixel i at the M-step of every EM iteration. However, we also have to ensure that these values satisfy the constraints $0 \leq \pi_j^i \leq 1$ and $\sum_{j=1}^K \pi_j^i = 1$. In the following, we present an efficient novel projection algorithm to achieve this goal.

For convenience, let us now denote with a_j ($j = 1, \dots, K$) the label parameter values ($\pi_j^{i(t+1)} \geq 0$) computed from (9). The problem we address here is the following: "Given a vector $a \in \mathcal{R}^K$ with $a_j \geq 0$ and the hyperplane $\sum_{j=1}^K y_j = 1$, find the point y on the hyperplane with nonnegative components that is closest to a ." This can be formulated as a linear constrained convex quadratic programming (QP) problem

$$\min_y \sum_{j=1}^K (y_j - a_j)^2 \quad \text{subject to } \sum_{j=1}^K y_j = 1 \text{ and } y_j \geq 0 \forall j = 1, 2, \dots, K. \quad (10)$$

In order to solve the previous QP problem, several approaches may be employed [9], such as *active-set* methods that use Lagrange multipliers, as well as *penalty-barrier* methods that formulate an objective function with penalty terms for equality and barrier terms for inequality constraints. We use here an active-set type of method, where we exploit the fact that the Hessian is the identity matrix which in turn leads to the derivation of closed-form analytical expressions for the Lagrange multipliers. This is of great value for both the efficiency and the robustness of the method, since it avoids the burden of numerical instabilities that occur frequently in the solution of large linear systems when the associated matrices are nearly singular.

One may proceed using the following Lagrange function:

$$L(y, \lambda_0, \lambda) = \frac{1}{2} \sum_{j=1}^K (y_j - a_j)^2 - \lambda_0 \left(\sum_{j=1}^K y_j - 1 \right) - \sum_{j=1}^K \lambda_j y_j \quad (11)$$

where λ_0 is the multiplier for the equality and λ_j , $j = 1, \dots, K$ the multipliers for the inequality constraints. First-order necessary conditions imply

$$y_j = a_j + \lambda_0 + \lambda_j. \quad (12)$$

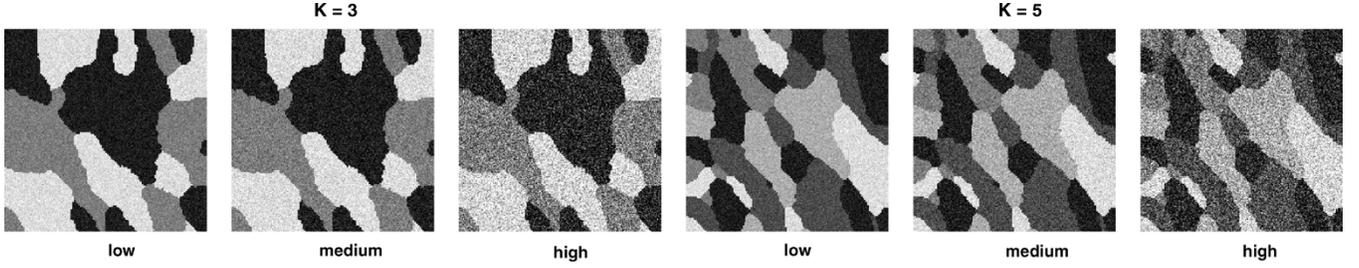


Fig. 1. Six noisy test images with $K = 3$ and $K = 5$ classes using three levels of noise.

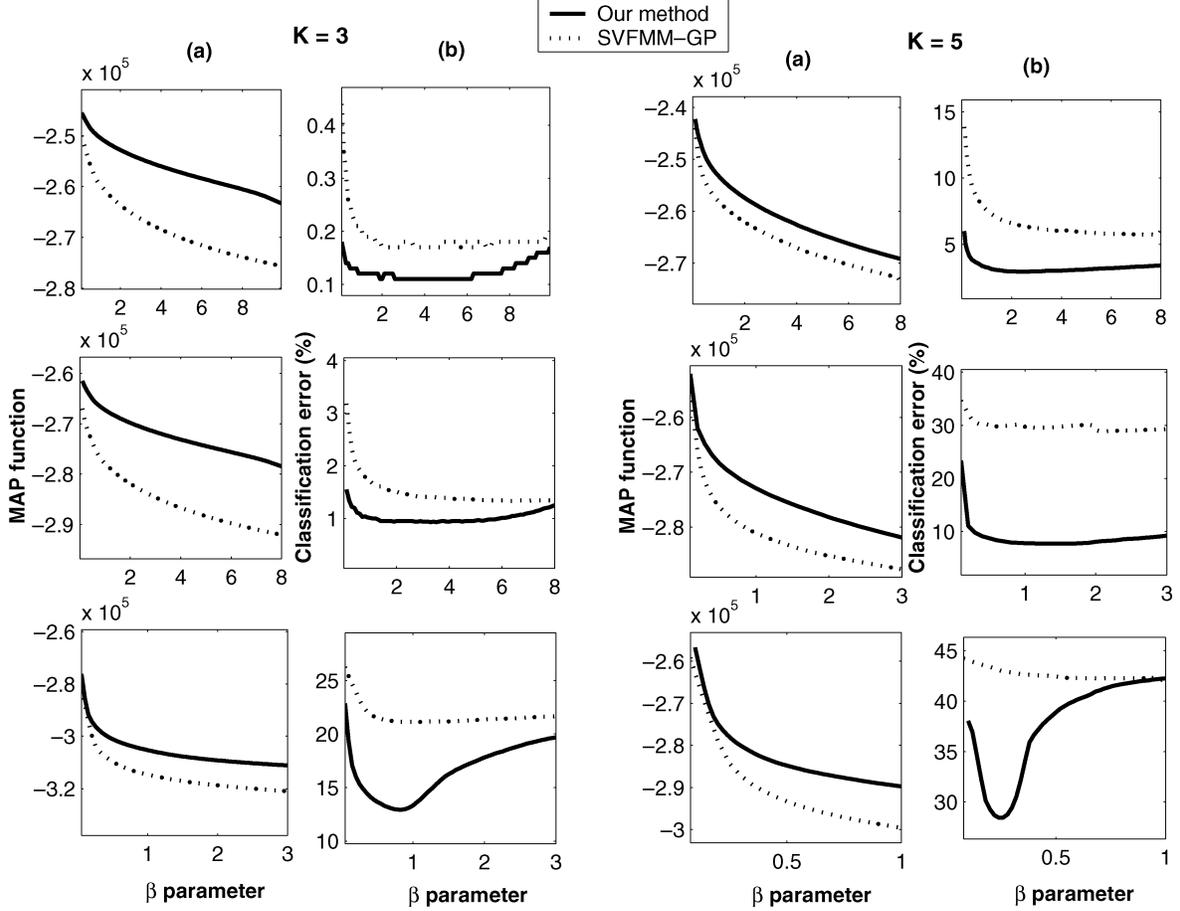


Fig. 2. Plot of (a) the MAP function and (b) the classification error for various β values in the case of the three noisy images with $K = 3$ and $K = 5$ classes.

Combining the previous with the equality constraint yields

$$\lambda_0 = \frac{1}{K} - \langle a \rangle - \langle \lambda \rangle \quad (13)$$

where $\langle v \rangle \equiv (1/K) \sum_{j=1}^K v_j$. Hence, substituting λ_0 in (12) we have that

$$y_j = \frac{1}{K} + a_j - \langle a \rangle + \lambda_j - \langle \lambda \rangle, \quad j = 1, \dots, K. \quad (14)$$

Note that the vector b with components $b_j = 1/K + a_j - \langle a \rangle$ is the projection of a on the hyperplane $\sum_{j=1}^K y_j = 1$. The λ 's must be chosen so as to satisfy the inequality constraints. Khun-Tucker conditions [9] state that at the minimizer y^*

$$\lambda_j \geq 0, \quad \lambda_j > 0 \text{ if } y_j^* = 0 \text{ (Active constraint), } \lambda_j y_j^* = 0. \quad (15)$$

We present a very efficient iterative strategy for calculating the λ 's for the previous problem.

Let y denote the vector at the current iteration. Initially, we set $y_j = b_j, \forall j = 1, \dots, K$. In the general case, there exist m negative components y_j . The corresponding set of indexes $S = \{j, \text{ with } y_j < 0\}$ constitutes the active set of constraints for the current vector y .

- For all $j \notin S$, we set $\lambda_j = 0$.
- For all $j \in S$, we set $y_j = y_j^* = 0$ and we compute the corresponding λ_j by solving an $m \times m$ linear system that force the inequalities to be satisfied as equalities, namely $y_j + \lambda_j - \langle \lambda \rangle = 0$, leading to

$$\lambda_j = \frac{1}{m - K} \sum_{k \in S} y_k - y_j. \quad (16)$$

- We compute the updated y_j values for $j \notin S$ using the new vector λ via (14).

The previous procedure is repeated until a feasible point is obtained, i.e., $y_j \geq 0, \forall j$. This is the desired minimizer ($y^* = y$).

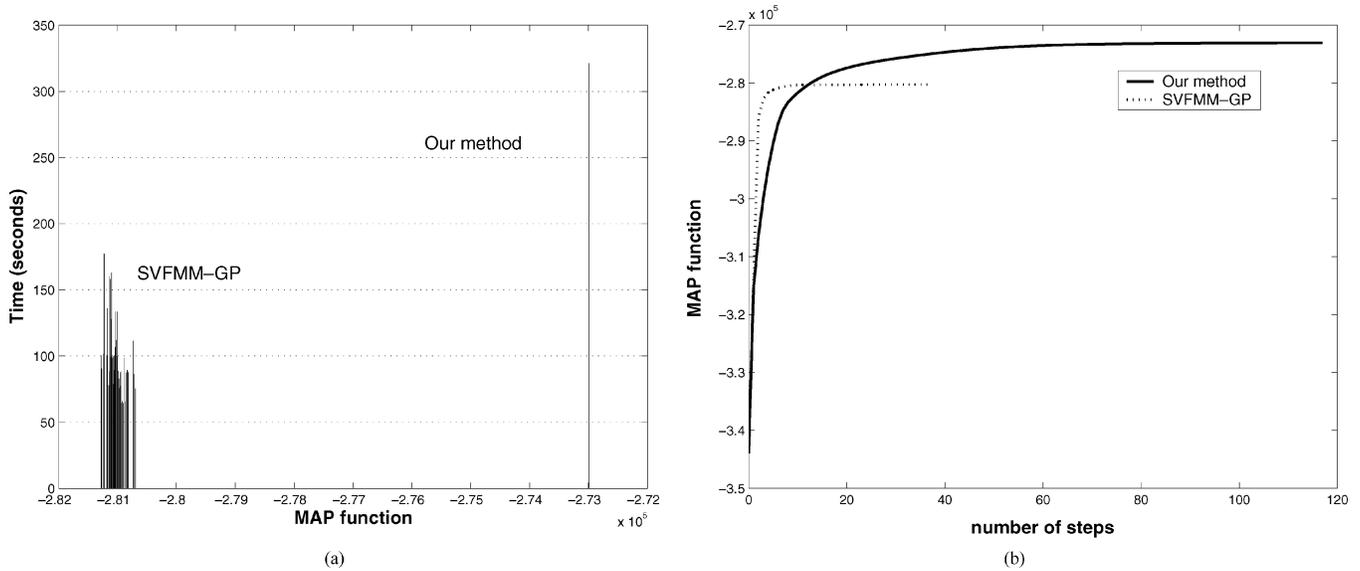


Fig. 3. (a) Execution time versus MAP function values for 50 runs of the proposed and the SVFMM-GP methods. (b) Typical run example of MAP function versus number of steps is also presented.

Note that (16) produces positive values for λ_j , hence, no constraint is to be dropped ever from the active set, i.e., if once some y_j becomes zero then it retains this value forever. This is a very important point as far as efficiency is concerned and, in addition, guarantees the finite termination property of the algorithm. When all constraints are satisfied, we have reached the sought solution.

Finally, it must be noted that apart from problems of low dimensionality as in the case of image segmentation (where K is equal to the number of pixel labels), we have also applied the previous projection technique to problems of large dimensions (e.g., $K = 65536$) arising in other image processing problems and we have verified its efficiency and robustness.

IV. EXPERIMENTAL RESULTS

A series of image segmentation experiments has been conducted to evaluate and compare the effectiveness of the proposed technique. Since the main contribution of our work is on improving the M-step of the SVFMM model that estimates the label parameters π_j^i , we compared our approach with the generalized EM scheme proposed in the original SVFMM model description that employs the GP technique (termed as SVFMM-GP) as described in [6].

In this letter, we present results using two simulated test images being sampled from the MRF model using a Gibbs sampler [7], with $K = 3$ and $K = 5$ classes, where we have added three levels of Gaussian noise with standard deviation of 18, 25, and 52, respectively (Fig. 1). Fig. 2 illustrates the comparative results from the application of the two methods to each noisy image. Two evaluation criteria have been used for the comparison study: a) the maximum attained value of the MAP objective function (4) and b) the classification (segmentation) error defined as the percentage of misclassified pixels. Therefore, for each image segmentation problem we provide two diagrams that illustrate the performance of the models according to the previous two criteria for several values of the β parameter. These results demonstrate that our approach provides a better maximum of the MAP function. Moreover, it provides significantly better segmentation accuracy, since the misclassification ratio is considerably lower in our approach, especially for high levels of noise.

In order to gain insight on the reason why our method yields consistently better results, the following additional experiments were conducted. For each method and test image (Fig. 1), we performed 50 runs

starting from different initial points π_j^i while keeping the same initial values for the mixture component parameters (μ_j, σ_j). The regularization parameter β was kept equal to $\beta = 1$ throughout. In Fig. 3(a), we plot the execution time for each of the 50 runs, versus the obtained MAP function value using both methods on the fifth test image. Note that the results for all images were similar.

In Fig. 3, we observe that our method returns in all cases the same result, whereas the SVFMM-GP technique depends on the initialization, hence, returning not one, but a number of different maxima. This behavior can be explained noting that our method locates the global maximum of the Q_{MAP} function (6) in every M-step, since it only projects the unique unconstrained maximum (9) onto the constraint boundary. On the contrary, SVFMM-GP is trapped to the first local maximum encountered. There is a small variation in the execution times (≈ 1 s) of our method that cannot be shown in Fig. 3(a) since the value of function is the same in all runs. This is expected since the projections onto the constraint boundary may require different times, depending on the initial conditions. However, in our experiments the dimension of the constraint polyhedron is small ($K = 3$ or 5), thus, these differences are negligible.

In Fig. 3(b), we plot the MAP function values versus the number of EM-steps. From that plot we see that our method requires more EM iterations to converge and it is slower as it can be deduced from Fig. 3(a). However, our method reaches higher maximum values which implies better segmentation performance.

V. CONCLUSION

We have presented a new method to maximize the label parameter values at the M-step of the EM algorithm for training GMMs with MRF priors for image segmentation. Experimental results on simulated images demonstrate that the proposed modification improves, in some cases significantly, the segmentation performance of this method. It must be noted that we have also tested the proposed algorithm with real images, where in all tested cases the proposed M-step provides a better maximum of the objective function. However, due to space constraints, we do not present these results here. Future work will focus on applying the method to real world segmentation problems arising in medical imaging and bioinformatics and also to consider segmentation of color and texture images. We also plan to design more sophisticated prior functions that will take into account not only pixel adjacency, but also image information, such as for example the existence of edges.

REFERENCES

- [1] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 901–914, Apr. 1992.
- [2] N. K. Pal and S. K. Pal, "A review of image segmentation techniques," *Pattern Recognit.*, vol. 26, pp. 1277–1294, 1993.
- [3] K. Chen, D. Wang, and X. Liu, "Weight adaptation and oscillatory correlation for image segmentation," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1106–1123, Sep. 2000.
- [4] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- [5] D. Ormoneit and V. Tresp, "Averaging, maximum penalized likelihood and Bayesian estimation for improving Gaussian mixture probability density estimates," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 639–650, Jul. 1998.
- [6] S. Sanjay-Gopal and T. J. Hebert, "Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1014–1028, Jul. 1998.
- [7] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation–maximization algorithm," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, Jan. 2001.
- [8] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified EM algorithm," *IEEE Trans. Med. Imag.*, vol. 9, no. 1, pp. 84–93, Jan. 1990.
- [9] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer-Verlag, 1999.

An Improved Conjugate Gradient Scheme to the Solution of Least Squares SVM

Wei Chu, Chong Jin Ong, and S. Sathiya Keerthi

Abstract—The least square support vector machines (LS-SVM) formulation corresponds to the solution of a linear system of equations. Several approaches to its numerical solutions have been proposed in the literature. In this letter, we propose an improved method to the numerical solution of LS-SVM and show that the problem can be solved using one reduced system of linear equations. Compared with the existing algorithm for LS-SVM, the approach used in this letter is about twice as efficient. Numerical results using the proposed method are provided for comparisons with other existing algorithms.

Index Terms—Conjugate gradient (CG), least square support vector machines (LS-SVM), sequential minimal optimization (SMO).

I. INTRODUCTION

As an interesting variant of the standard support vector machines (SVMs) [2], least squares support vector machines (LS-SVM) have been proposed by Suykens and Vandewalle [3] for solving pattern recognition and nonlinear function estimation problems. The links

Manuscript received September 21, 2003; revised March 7, 2004. This work was supported in part by the National Institutes of Health (NIH) and its National Institute of General Medical Sciences (NIGMS) Division under Grant 1 P01 GM63208 (NIH/NIGMS Tools and Data Resources in Support of Structural Genomics). The work of W. Chu was supported by a National University of Singapore Research Scholarship.

W. Chu is with Gatsby Computational Neuroscience Unit, University College London, London WC1E 6BT, U.K.

C. J. Ong is with Department of Mechanical Engineering, National University of Singapore, Singapore 119260 (e-mail: mpeongcj@nus.edu.sg).

S. S. Keerthi is with Yahoo! Research Labs, Pasadena, CA 91105 USA.

Digital Object Identifier 10.1109/TNN.2004.841785

between LS-SVM classifiers and kernel Fisher discriminant analysis have also been established by Van Gestel *et al.* [4]. The LS-SVM formulation has been further extended to kernel principal component analysis, recurrent networks and optimal control [5]. As for the training of the LS-SVM, Suykens *et al.* [1] proposed an iterative algorithm based on the conjugate gradient (CG) algorithm. Keerthi and Shevade [6] adapted the sequential minimal optimization (SMO) algorithm for SVM [7] for the solution of LS-SVM.

In this letter, we propose an improved algorithm with CG methods for LS-SVM. We first show the optimality conditions of LS-SVM, and establish its equivalence to a reduced linear system. CG methods can then be employed for its solution. Compared with the algorithm proposed by Suykens *et al.* [1], our algorithm is equally robust and is at least twice as efficient.

We adopt the following notations. $x \in R^d$, $D \in R^{n \times m}$ are d -dimensional column vector and $n \times m$ matrix of real entries, respectively; x^T is the transpose of x ; $\mathbf{1}_n$ and $\mathbf{0}_n$ are n -column vectors of entries 1 and 0, respectively. This letter is organized as follows. In Section II, we review the optimization formulation of LS-SVM, and then show the simplification of the optimality conditions to a reduced linear system. In Section III, we present the results of numerical experiments using our proposed algorithm on some benchmark data sets of different sizes, and compare with the results obtained using the conjugate method by Suykens *et al.* [1] and the SMO algorithm by Keerthi and Shevade [6]. We conclude in Section IV.

II. LS-SVM AND ITS SOLUTION

Suppose that we are given a training data set of n data points $\{x_i, y_i\}_{i=1}^n$, where $x_i \in R^d$ is the i th input vector and y_i is the corresponding i th target. For binary classification problems y_i takes only two possible values $\{-1, +1\}$, whereas $y_i \in R$ for regression problems. We employ the idea to transform the input patterns into the reproducing kernel Hilbert space (RKHS) by a set of mapping functions $\phi(x)$ [5]. The reproducing kernel $K(x, x')$ in the RKHS is the dot product of the mapping functions at x and x' , i.e.

$$K(x, x') = \langle \phi(x) \cdot \phi(x') \rangle. \quad (1)$$

In the RKHS, a linear classification/regression is performed. The discriminant function takes the form $f(x) = \sum_{i=1}^n \langle \mathbf{w} \cdot \phi(x) \rangle + b$, where \mathbf{w} is the weight vector in the RKHS, and $b \in R$ is called the bias term. The discriminant function of LS-SVM classifier [3] is constructed by solving the following minimization problem:

$$\min_{\mathbf{w}, b, \xi} P(\mathbf{w}, b, \xi) = \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle + \frac{C}{2} \sum_{i=1}^n \xi_i^2 \quad (2)$$

$$\text{s.t. } y_i - (\langle \mathbf{w} \cdot \phi(x_i) \rangle + b) = \xi_i \quad i = 1, \dots, n \quad (3)$$

where $C > 0$ is the regularization factor and ξ_i is the difference between the output y_i and $f(x_i)$. Using standard techniques [8], the Lagrangian for (2)–(3) is

$$L(\mathbf{w}, b, \xi; \alpha) = \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle + \frac{C}{2} \sum_{i=1}^n \xi_i^2 + \sum_{i=1}^n \alpha_i (y_i - (\langle \mathbf{w} \cdot \phi(x_i) \rangle + b) - \xi_i) \quad (4)$$