Learning to Segment Using Machine-Learned Penalized Logistic Models

Yong Yue
Department of Diagnostic Radiology, School of Medicine,
Yale University, New Haven, CT 06511 USA
yong.yue@yale.edu

Hemant D. Tagare
Department of Diagnostic Radiology, School of Medicine,
Department of Biomedical Engineering, Yale University, New Haven, CT 06520
hemant.tagare@yale.edu

Abstract

Classical maximum-a-posteriori (MAP) segmentation uses generative models for images. However, creating tractable generative models can be difficult for complex images. Moreover, generative models require auxiliary parameters to be included in the maximization, which makes the maximization more complicated.

This paper proposes an alternative to the MAP approach: using a penalized logistic model to directly model the segmentation posterior. This approach has two advantages: (1) It requires fewer auxiliary parameters, and (2) it provides a standard way of incorporating powerful machine-learning methods into segmentation so that complex image phenomenon can be learned easily from a training set.

The technique is used to segment cardiac ultrasound images sequences which have substantial spatio-temporal contrast variation that is cumbersome to model. Experimental results show that the method gives accurate segmentations of the endocardium in spite of the contrast variation.

1. Introduction

Classical maximum-a-posteriori (MAP) segmentation requires generative models for images. In practice, the MAP approach has a serious practical limitation: generative models for complex images are hard to create. Moreover, they usually have additional auxiliary parameters which have to be estimated simultaneously with the segmentation.

This paper suggests an alternative to segmenting complex images – abandon the MAP approach and directly model the posterior probability of segmentation. By "directly model" we mean modeling the posterior probability without using Bayes theorem. The direct modeling approach has advantages: (1) It limits the number of auxiliary parameters. (2) It provides a standard way of incorporating machine-learning into segmentation thereby allowing the use of powerful non-linear methods.

We propose a class of posterior models that we call *penalized logistic models*. These models are used in a standard level-set framework with a shape prior. Our main application for the method is segmenting B-mode cardiac ultrasound image sequences. The contrast across the endocardium in these sequences (see the top rows of figures 4 and 5) changes spatially as well as temporally throughout the cardiac cycle. This contrast variation is due to a complex interaction of the propagating ultrasound wave and the fiber geometry of the myocardium. A generative model for such images is possible (see [11] for example), but has many additional parameters to optimize over. In comparison, as we show in 5, direct posterior modeling can accurately segment the B-mode image sequence without additional parameters.

1.1. Previous Work

Due to space limitations, we only provide a brief literature review. An overview of different approaches to ultrasound segmentation can be found in [10]. MAP ultrasound segmentation has received considerable attention, especially in the cardiac domain, e.g. [13]. Contrast variation in the myocardium is a complex phenomenon, related to the interaction of the ultrasound wave and directional fibers in the myocardium. The study [11] attempts to segment cardiac ultrasound images in the presence of contrast inhomogeneity using a MAP framework.

Logistic models are discussed in detail in classical statistics [9], but surprisingly they are not commonly used in image segmentation. One exception is [4]. Direct models of posteriors have been used for *image recognition* in com-

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

puter vision under the rubric of *conditional* or *discriminative random fields* [8, 6, 7, 14]. These models are typically second-order quadratic Markov random fields.

Machine-learning is receiving increasing attention as a tool for segmentation [15, 16], but we are not aware of any work that uses machine-learned penalized logistic models in segmentation. Non-logistic machine-learning tools have been proposed for ultrasound segmentation, e.g. [1]. These methods are neither probabilistic nor level-set based.

We use two machine learning techniques, the support-vector machine [3] and the import vector machine [17]. Both depend on the "kernel trick" [12] to provide highly non-linear models which are nevertheless computationally tractable.

2. Kernel Logistic Models and the IVM

We start with a brief tutorial on kernel logistic models based on [17]. Suppose that x, y are $\mathcal{R}^n$-valued and $\{0,1\}$-valued random variables respectively. The *kernel logistic model* assumes that the ratio of the posterior probabilities $p(y=1 \mid x)$ and $p(y=0 \mid x)$ is given by

$$\log \frac{p(y=1 \mid x)}{p(y=0 \mid x)} = f(x) = \sum_{i=1}^{M} \alpha_i K(\tilde{x}_i, x), \qquad (1)$$

where f is the *logistic regression function*, K is the kernel, and $\tilde{x}_i$ are the *import vectors*. The model of eq. (1) gives the posterior probability $p(y \mid x) = \frac{(1-y)+y\exp(f(x))}{1+\exp(f(x))}$.

Learning f in eq. (1) is equivalent to learning $M, \{\alpha_i\}, \{\tilde{x}_i\}$ from a training set $\{x_i, y_i \mid i = 1, \ldots, T\}$. Learning is carried out by maximizing the conditional log-likelihood $\sum_{i=1}^{T} \log p(y_i \mid x_i)$ with respect to $M, \{\alpha_i\}, \{\tilde{x}_i\}$. This is a complex non-linear optimization problem; nevertheless, a greedy but effective algorithm is available for it in the machine-learning literature [17].

The algorithm uses a result of kernel function theory which states that the optimizing import vectors are a subset of the training samples $\{x_i\}$. The algorithm begins with a null set of import vectors. It iteratively increases the set by adding to it one training set vector per iteration. The vector that is added is the one that causes the most increase in log-likelihood $\sum_{i=1}^{T} \log p(y_i \mid x_i)$. The algorithm terminates when the increase in the log-likelihood falls below a threshold. A brief description of the algorithm is given below. For further details, see [17] :

Algorithm (IVM):

(1) Let $X = \{x_i\}$ be the set of x_i's in the training set and V be the set of import vectors. Initialize $V = \emptyset$.

(2) For every $x_i \in X - V$, create the set $A_i = V \bigcup x_i$.

(3) Regard each A_i as the import vector set and for it use gradient ascent to find $\{\alpha_i\}$ that maximize $\sum_{i=1}^{T} \log p(y_i \mid x_i)$.

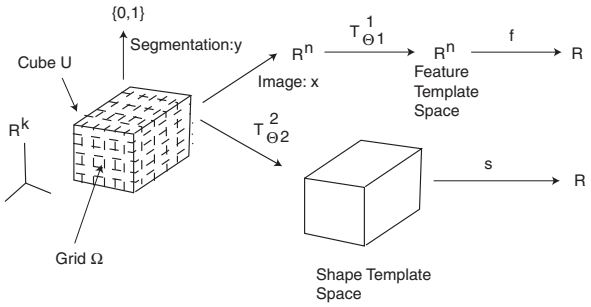


Figure 1. The Penalized Logistic Model.

(4) Set V to that A_i which gives the maximum value of $\sum_{i=1}^{T} \log p(y_i \mid x_i)$. This increases the import vector set by one element, namely the "best" element of $X - V$.

(5) Terminate if the increase in the log-likelihood $\sum_{i=1}^{T} \log p(y_i \mid x_i)$ falls below a threshold, else go to (2).

Because the best training sample is chosen at every step, the number of import vectors is usually quite small. The resulting logistic model is referred to as an *import vector machine* (IVM).

We use Gaussian radial basis functions as the kernel for the IVM so that $K(\tilde{x}_i, x) = \exp(-\|\tilde{x}_i - x\|^2/2\sigma^2)$.

3. Segmentation with Logistic Models

3.1. Penalized Logistic Model

We now turn to using logistic models and the IVM for segmentation. To do this, we first normalize the images using two template spaces: a *feature template space* and a *shape template space* (shown in figure 1). The feature template space is used to normalize image gray level values and the shape template space is used to normalize segmentation shape. As shown in figure 1, Ω is a finite grid of image pixels in a cube U of $\mathcal{R}^k$. The variable u stands for pixels, i.e. grid vertices, of Ω, The set of all u is finite and each u has k coordinate functions $a_1(u), \ldots, a_k(u)$, where $a_n(u)$ is the coordinate of u along the nth axis.

The image data is a vector-valued function $x : \Omega \to \mathcal{R}^n$, and the segmentation is given by the indicator function $y : \Omega \to \{0,1\}$ of one of the segmented regions. The feature template space is another copy of $\mathcal{R}^n$ and a transformation $T^1_{\theta_1} : \mathcal{R}^n \to \mathcal{R}^n$, which is parameterized by θ_1, maps the image values into the feature template space. Finally, $f : \mathcal{R}^n \to \mathcal{R}$ is the logistic regression function defined in the feature template space.

The shape feature space is another copy of the cube $U \in \mathcal{R}^k$. We define a *mean shape function* $s : U \to \mathcal{R}$ and pull it back onto Ω using a second transformation $T^2_{\theta_2} : \Omega \to U$ which is parameterized by θ_2 and is invertible.

Setting $\theta = (\theta_1, \theta_2)$, image segmentation is obtained by maximizing the posterior log-likelihood $\log p(y, \theta \mid x)$ with

respect to y and θ. To do so, we write $p(y, \theta \mid x) = p(y \mid \theta, x) \times p(\theta \mid x)$ from which we get

$$\log p(y, \theta \mid x) = \log p(y \mid \theta, x) + \log p(\theta \mid x). \quad (2)$$

The terms on the right hand side of (2) are set as follows:

The $\log p(y \mid \theta, x)$ term : We set

$$\log p(y|\theta, x) = \sum_{u \in \Omega} \log p(y(u)|\theta_1, x(u)) - \gamma(y, \theta_2), \quad (3)$$

where

$$p(y(u)|\theta_1, x(u)) = \frac{(1 - y(u)) + y(u) \exp(f(T^1_{\theta_1}(x(u))))}{1 + \exp(f(T^1_{\theta_1}(x(u))))} \quad (4)$$

and the function f is given by the kernelized model of equation (1). The $\gamma(y, \theta_2)$ term in eq. (3) is a regularization penalty term and we set it to:

$$\begin{aligned} \gamma(y, \theta_2) &= \lambda_1 \times \text{length (bndry}(\{u \mid y(u) = 1\}\text{)}) \\ &+ \lambda_2 \sum_{u \in \Omega} (y(u) - s(T^2_{\theta_2}(u)))^2. \end{aligned} \quad (5)$$

The first term on the right hand side of equation (5) is equal to λ_1 times the length of the boundary of the indicator function y and is a roughness penalty term. The second term is a shape prior term. It measures the deviation of the segmentation of y from the indicator function s of a pulled-back "mean shape" $s : U \rightarrow \{0, 1\}$. Of course, more complex shape priors can be used, but the mean shape prior is known to be useful in cardiac ultrasound segmentation [2]. We adopt it for simplicity.

The $\log p(\theta \mid x)$ term : This is the log-posterior of θ given x. We assume that we have an estimator $\theta^*(x)$ (we use support vector regression as the estimator in experiments), and use the normal distribution for $p(\theta \mid x)$, i.e.

$$\log p(\theta \mid x) = -\lambda_3 \|\theta - \theta^*(x)\|^2, \quad (6)$$

where $\lambda_3 > 0$ is a constant. This term is motivated by the well-known observation that the posterior distribution of many commonly used estimators is asymptotically normal.

Equations (2-6) define the *penalized logistic model* for image segmentation.

3.2. Segmenting with the penalized logistic function

The penalized logistic model can be used in a level set segmentation scheme. Let $\phi : \Omega \rightarrow \mathcal{R}$ be a level set function and let H be a smooth Heaviside function. Then using $y(u) = H(\phi(u))$ we can write the log-likelihood $\log p(y, \theta \mid x)$ of eq. (2) as the objective function $J(\phi, \theta \mid x)$:

$$\begin{aligned} & J(\phi, \theta \mid x) = \log p(y \mid \theta, x) \\ = & \sum_{u \in \Omega} \left\{ H(\phi(u)) f(T^1_{\theta_1}(x(u))) + \log \frac{1}{1 + \exp(f(T^1_{\theta_1}(x(u))))} \right\} \\ & -\gamma(H(\phi), \theta_2) - \lambda_3 \|\theta - \theta^*(x)\|^2, \end{aligned} \quad (7)$$

where, taking δ_0 to be the smooth Dirac delta function

$$\begin{aligned} \gamma(H(\phi), \theta_2) &= \lambda_1 \sum_{u \in \Omega} \delta_0(\phi(u)) |\nabla \phi(u)| \\ &+ \lambda_2 \sum_{u \in \Omega} (H(\phi(u)) - s(T^2_{\theta_2}(u)))^2. \end{aligned}$$

The segmentation is found by gradient ascent with respect to ϕ and θ:

$$\frac{\partial \phi}{\partial t} = \nabla_\phi J(\phi, \theta \mid x) \quad (8)$$

$$\frac{\partial \theta}{\partial t} = \nabla_\theta J(\phi, \theta \mid x), \quad (9)$$

where the expressions for $\nabla_\phi J(\phi, \theta \mid x)$ and $\nabla_\theta J(\phi, \theta \mid x)$ are given in the Appendix of the paper. Note that our convention is that θ is a row vector, hence the expression for $\nabla_\theta J(\phi, \theta \mid x)$ in the Appendix is a row vector.

3.3. Comments on and Extensions of the model

For a given $T^1_{\theta_1}$, the value of the regression function in eq. (4) at u only depends on $x(u)$. At first glance, this seems too simple to model complex images. But it is easy to extend this so that the regression function becomes more sophisticated. Two extensions are useful:

Neighborhood values: Suppose we have a scalar-valued image x and we filter it with a set of possibly non-linear filters to create a vector-valued image $\hat{x}$, called the *extended image*. Each pixel of the extended image contains neighborhood information from the corresponding pixel in x (see figure 2). Taking the extended image as the input image makes the regression function use information from the neighborhood of a pixel rather than just the gray level at a pixel. Theoretically, this step can be rigorously justified with *conditional random fields* [8].

Spatial location: When there is no contrast variation in an image, whether a pixel belongs to a region depends solely on the gray level of the pixel. When there is contrast variation, the same gray level has to be interpreted differently in different parts of the image, i.e. the regression function is now required to be a function of the gray level as well as the spatial location of the pixel.

To make the regression function depend on spatial location we simply add pixel coordinates to the extended image.

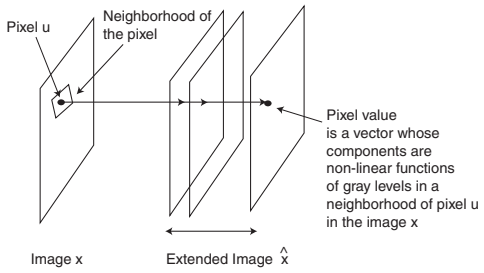


Figure 2. The Extended Image.

Given an image x, we define the extended image $\hat{x}$ by

$$\hat{x}(u) = \Big[x(u),\ (\text{avg } x)(u), (\text{std } x)(u),\ a_1(u),\ \ldots,\ a_k(u)\Big]^T \quad (10)$$

where $(\text{avg } x)(u)$ and $(\text{std } x)(u)$ are the Gaussian weighted mean and standard deviation of the image in a neighborhood of u, and $a_1(u), \ldots, a_k(u)$ are the pixel coordinates of u. By using this extended image as the input image, the regression function can use neighborhood information as well as location information.

4. Cardiac Ultrasound Segmentation

We use the above method to segment the endocardium in human cardiac ultrasound image sequences. We assume that each sequence has K frames and that all sequences begin at end-systole and stop at end-diastole. Thus, the grid Ω is three dimensional with the third dimension being time. Because all frames are synchronized from end-systole to end-diastole, the time coordinate is already "normalized". Given an image sequence $x : \Omega \to \mathcal{R}$ we form the extended image sequence $\hat{x}$ according to eq. (10). We refer to the first three components of the extended image as *intensity components* and the last three as the *coordinate components*.

4.1. Transformations

We have to specify two transformations $T^1_{\theta_1}$ and $T^2_{\theta_2}$ to proceed. We take the transformation $T^2_{\theta_2}$ to be:

$$T^2_{\theta_2}(u) = [a_1(u)\ a_2(u)\ a_3(u)]A + B, \text{ where}$$
$$A = \begin{bmatrix} p & s & 0 \\ -s & p & 0 \\ r & u & 1 \end{bmatrix}, B = \begin{bmatrix} v \\ w \\ 0 \end{bmatrix}^T, \quad (11)$$

so that $\theta_2 = (p, s, \ldots, v, w)$. This transformation preserves the time coordinate, $a_3(u)$, while transforming each frame in the sequence by a similarity transform.

We set the transformation $T^1_{\theta_1}$ to transform the intensity components and the coordinate components of the extended image by two independent transforms. For simplicity, the coordinate component transformation is set identical to $T^2_{\theta_2}$. That is,

$$T^1_{\theta_1}(\hat{x}(u)) = \Big[ax(u) + b,\ (\text{mean } ax + b)(u),\ (\text{std } ax + b)(u),\ T^2_{\theta_2}(u)\Big]. \quad (12)$$

We refer to the a, b parameters as *intensity normalization parameters*. Thus, the total set of parameters is $\theta = (a, b, p, \ldots, w)$. This completely defines the model.

We will need the derivatives of $T^1_{\theta_1}$ and $T^2_{\theta_2}$ with respect to θ_1 and θ_2 for later use. We denote the derivatives as:

1. The 6×8 matrix $\frac{\partial T^1_{\theta_1}}{\partial \theta_1}$ whose ij^{th} entry $(\frac{\partial T^1_{\theta_1}}{\partial \theta_1})_{ij}$ is the partial derivative of the i^{th} component of $T^1_{\theta_1}$ with respect to the j^{th} component of θ_1. We omit the formula for the matrix. It can be obtained by a straight forward partial differentiation of eq. (12).

2. The 3×6 matrix $\frac{\partial T^2_{\theta_2}}{\partial \theta_2}$ whose ij^{th} entry $(\frac{\partial T^2_{\theta_2}}{\partial \theta_2})_{ij}$ is the partial derivative of the i^{th} component of $T^2_{\theta_2}$ with respect to the j^{th} component of θ_2.

4.2. Learning

To use the model, we have to learn the logistic regression function and the estimator θ^* from a training set. This is done as follows:

(1) Calculate θ and the mean shape function for the training set: The training set contains image sequences x_i and segmentations $y_i,\ i = 1, \ldots, M$. We first calculate the spatial transformation parameters that affinely align segmentations $y_2, \ldots, y_M$ to y_1. The transform for the first segmentation is assumed to be identity, $T^2_{\theta_{2,1,}} = id$, and for $i = 2, \ldots, M$ the parameters $\theta_{2,i}$ are chosen to minimize $\|y_1 - \left(T^2_{\theta_{2,i}}\right)^{-1}(y_i)\|^2$.

After alignment, the mean shape function is calculated by $s = H(1/M \sum_{i=1}^{M} \left(T^2_{\theta_{2,i}}\right)^{-1}(y_i))$, where H is the smooth Heaviside function.

Finally, the parameters a, b in the transformation $T^1_{\theta_1}$ (see eq.(12)) are determined by setting the mean value of the first intensity component of the exteded image (i.e. the main value of the gray level of the original image) in blood pool and the myocardium to fixed values (20 and 70 respectively).

This completely determines all parameters θ_i for all i.

(2) Learn the logistic regression function: Using $\theta_{1,i}, \theta_{2,i}$ determined above, the training images and manual segmentations are normalized by insertion in feature and shape template spaces and the IVM algorithm of section 2 is used to learn the logistic regression function f.

(3) Learn the estimator $\theta^*(x)$: The parameter estimator $\theta^*(x)$ is created by using support vector regressions with Gaussian radial basis function kernels [5]. The gray level transformation parameters are learned by a support vector regression from the histogram h_i of the first component of the extended image to the (a_i, b_i) values calculated in step (1) above.

The estimator for transformation parameters $(p, \ldots, w)$ are learned as follows: The extended image x_i is tiled with polar co-ordinate tiles whose origin is the center of the image, and the mean value of the first intensity component is calculated in each tile. The vector of the mean values from all tiles gives the overall brightness pattern of the image and captures information about the location, size and orientation of the left ventricular cavity and the myocardium. The transformation parameters are learned by a support vector regression of the $\theta_{2,i}$ calculated in step (1) above with the vector of mean values.

These support vector regressions give the estimator $\theta^*(x)$.

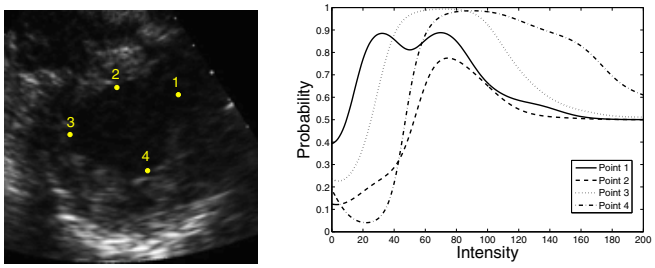
5. Experimental Results

We evaluated our algorithm using 2-D clinical short-axis echocardiographic image sequences. A total of 20 sequences were acquired from an Acuson Sequoia imaging system. The sequences were of different subjects obtained from a clinical database. All sequences had 4 frames with each frame 140×140 pixels. All sequences started at end-systole and stopped at end-diastole. 10 sequences were randomly selected as the training data set and their endocardiums were manually outlined. Training was carried out as explained in section 4. The σ of the smoothing Gaussian in the extended image was set to 7 and the Gaussian radial basis functions were used as kernels in the IVM and SVM with $\gamma = 1/2\sigma^2$ set to 0.001. The parameters $\lambda_i, i = 1, 2, 3$ were set as 2,1,1, respectively.

The learned regression function and estimator $\theta^*(x)$ was used to segment the remaining 10 image sequences. The level set was initialized by two mouse clicks in image sequence to obtain the blood pool center and an orientation.

The goal of the experiments was to demonstrate the main claims of this paper, which are (1) The penalized logistic model can account for complex image patterns without auxiliary parameters; in particular it can account for spatio-temporal contrast variation in cardiac ultrasound images, (2) This approach gives accurate segmentations.

Our first evaluation was to check whether the IVM learned to model the contrast inhomogeneity. Figure 3(a) shows the first frame of the image sequence in the top row of Fig. 4. The local contrast between the blood pool and the myocardium is greater near locations marker 2 and 4 than near locations marked 1 and 3. Figure 3(b) shows the posterior probability of the pixel belonging to the myocardium



(a) (b)

Figure 3. Short-axis cardiac ultrasound image. (a) The contrast along the endocardium is inhomogeneous. The contrast is higher at locations 2 and 4 than at 1 and 3, and (b) the probability of four points.

as a function of image intensity as learned by the regression function at locations 1,2,3,4. (The regression function depends on all components of the extended image. Figure 3(a) shows the dependence on the first component with the coordinate components set to the coordinates of the locations 1,2,3 and 4.) Note that the posterior probability function is similar for locations 1 and 3 and also for locations 2 and 4, but the two sets of functions are quite different. The posterior probability of the pixel belonging to the myocardium equals 0.5 at locations 1 and 3 when the intensity is 10 and 25 respectively, and at locations 2 and 4 when the intensity is 53 and 48 respectively. This shows that the IVM is willing to classify less brighter pixels at locations 1 and 3 as myocardium than at locations 2 and 4, i.e., the IVM has learned to account for spatial contrast variation without extra parameters that need to be optimized over during segmentation.

The ability to capture temporal variation is shown in figure 4. The top row of the figure shows a typical test sequence. The endocardium near the 2 o'clock position has poor contrast in the first two frames but becomes noticeable in the last frame, demonstrating spatio-temporal contrast change. To test whether the logistic model learned these changes, we visualized the posterior probability of a pixel belonging to the myocardium as calculated in the final iteration of the segmentation. That is, for every pixel u in the image sequence, we measured $\exp(f(T^1_{\theta_1}(u)))/(1 + \exp(f(T^1_{\theta_1}(u))))$ taking the value of θ_1 from the last iteration. The probabilities are shown in the second row of the figure where the probability values are color-mapped blue→purple→red where purple is the posterior probability of 0.5 and pixels that are redder have higher probability of belonging to the myocardium (bluer pixels have a higher probability of belonging to the blood pool). The second row shows clearly that the posterior probabilities allow for spatio-temporal variation – the blood pool/myocardium boundary is clearly evident across the whole sequence in spite of spatio-temporal contrast variation. The third row

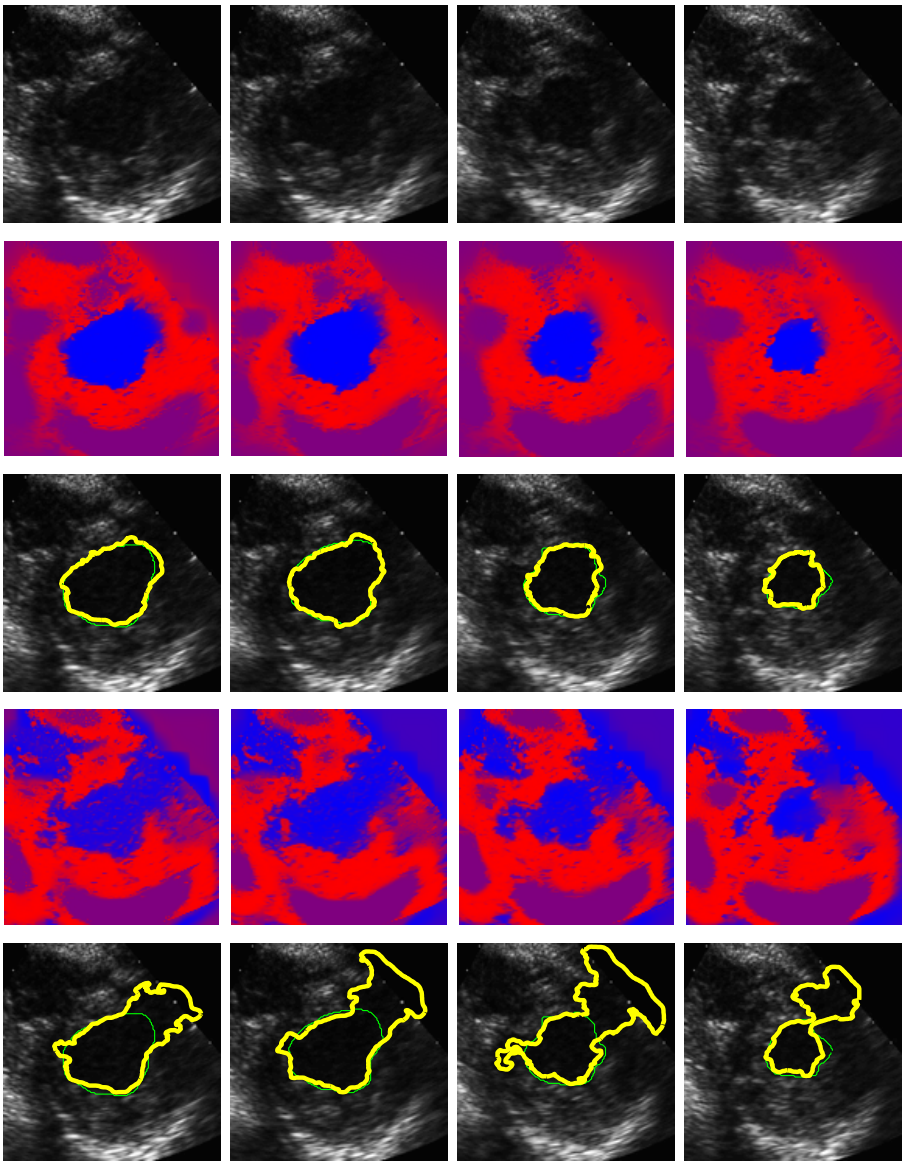


Figure 4. The segmentation of a sequence of echocardiographic images. First row: four frames in the sequence. Second row: the posterior probability map. Blue indicates the ventricular cavity, and red indicates the myocardium. Third row: the results of segmentation: algorithm (yellow) and manual (green). Fourth row: the posterior probability by a spatially homogenous model. Fifth row: segmentation using the spatially homogenous model.

of the figure shows algorithm segmentation results in yellow with corresponding manual segmentation in green. The mean distances between the two contours in the four frames are $1.64,\ 1.18,\ 1.63$, and 1.37 pixels respectively while the Hausdorff distances are $4.98,\ 3.80,\ 6.76$, and 6.40 pixels respectively. The two sets of contours are quite close given that the manual segmentation contains some inaccuracy and the size of the speckle grain is about 7 pixels $\times$ 7 pixels.

To further demonstrate the importance of taking spatio-temporal contrast variation into account, we excluded the coordinate components (the $a_1(u), \ldots, a_n(u)$ components in eq. (10)) from the extended image. From the argument in section 3.3 we know that eliminating coordinates eliminates all spatio-temporal variation in the model. The fourth row of figure 4 shows the posterior probability of pixels belonging to the myocardium calculated by this reduced model in the last iteration of the segmentation. Notice the "leakage" of the blue pixels (pixels with high posterior probability of belonging to the blood pool) near the 2 o'clock position. The last row of figure 4 shows the segmentation achieved by using the reduced model. The segmentation has also leaked into myocardium, clearly showing the need for taking contrast variation into account.

Figure 5 shows segmentation of another test sequence

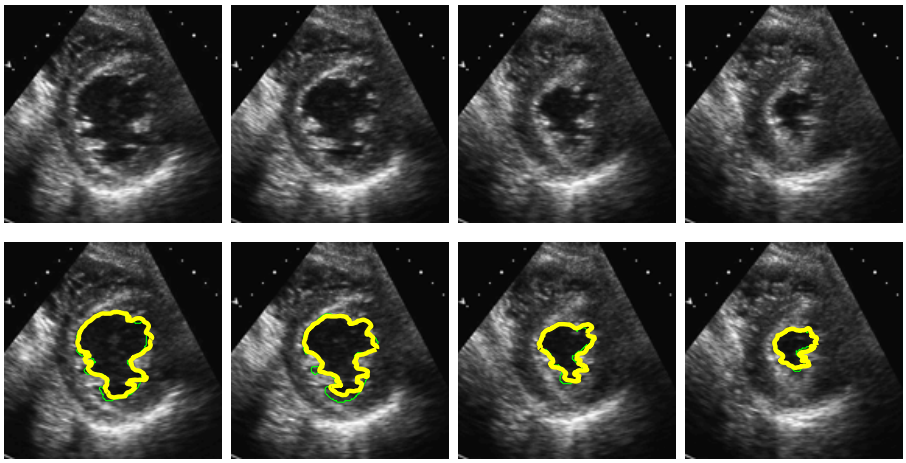


Figure 5. The segmentation of a second sequence. Top row: four frames in the sequence. Bottom, segmentation: algorithm (yellow) and manual (green).

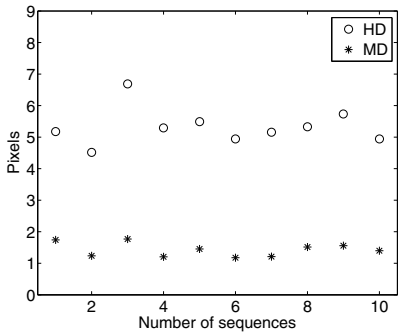


Figure 6. Comparison of algorithm and manual segmentations. HD: Hausdorff distance, MD: mean distance.

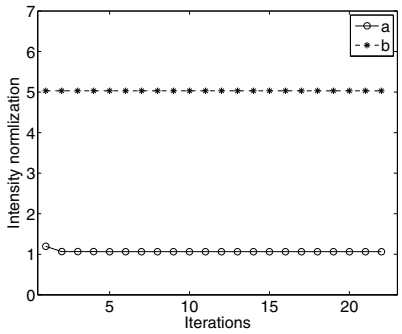


Figure 7. Fast convergence of the intensity normalization coefficients.

(the color row showing posterior probabilities is omitted to conserve space). The results show that the algorithm is able to segment it quite well.

The accuracy of all 10 test segmentations is summarized in figure 6. The figure shows the Hausdorff distance and the mean distance between the algorithm segmentation and manual segmentations for all 10 test sequences. All algorithm segmentations were within 7 pixels of manual segmentations.

The computational time for each iteration is 6.0 sec for a MATLAB implementation. The segmentation generally converges in 20 iterations. It is expected that the computation will be faster with C++ implementation.

A final point. Classical MAP models often get trapped in local maxima. This happens because in the early iterations MAP algorithms do not have good estimates of θ. In contrast, our algorithm initializes θ to the support vector estimate $\theta^*(x)$. Hence, good estimates are available for θ from the very beginning and θ converges to the final value quickly. Figure 7 illustrates this by showing the value of the intensity normalization parameters a and b (refer to equation (12)) as a function of iteration number. The parameter values have converged in the first 3 iterations. In the remaining iterations, the algorithm is essentially only evolving the level set. Thus, the risk of getting trapped in a local maximum is greatly reduced for this approach.

6. Conclusion

We proposed a methodology for directly modeling the posterior probability of segmentation using machine-learned penalized logistic models. This methodology does not require a generative data model and enables the use of powerful machine-learning techniques such as IVM and SVM in segmentation. The algorithm was used to segment ultrasound image sequences which exhibit considerable spatio-temporal contrast variation. The algorithm successfully learned the contrast variation from a training set and segmented test sequences with an accuracy of 7 pixels compared to manual segmentations. Further, the algorithm demonstrated rapid convergence of auxiliary parameters.

Acknowledgements

This research was supported by the grant R01 HL077810-02 from National Heart, Lung, and Blood Institute.

References

[1] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu. Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree. *IEEE Trans. on Medical Imaging*, 27:1342–1355, 2008.

[2] Y. Chen, F. Huang, H. D. Tagare, and M. Rao. A coupled minimization problem for medical image segmentation with priors. *International Journal of Computer Vision*, 71:259–272, 2007.

[3] N. Cristianini and J. Shawne-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[4] M. A. T. Figueiredo. Bayesian image segmentation using gaussian field priors. *In CVPR Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer, 2001.

[6] X. He, R. S. Zemel, and M. M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2004.

[7] S. Kumar and M. Hebert. Discriminative random fields. *International Journal of Computer Vision (IJCV)*, 68:179–201, 2006.

[8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proc. Int. Conf. on Machine Learning*, 2001.

[9] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall/CRC, 1989.

[10] J. Noble and D. Boukerroui. Ultrasound image segmentation: a survey. *IEEE Trans. Med. Imaging*, 25:987–1010, 2006.

[11] X. Qian, H. Tagare, and Z. Tao. Segmentation of rat cardiac ultrasound images with large dropout regions. *IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis*, 2006.

[12] B. Schlkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2001.

[13] Z. Tao, H. Tagare, and J. Beaty. Evaluation of four probability distribution models for speckle in clinical cardiac ultrasound images. *IEEE Trans. on Medical Imaging*, 25:1483–1491, 2006.

[14] M. Tappen, K. G. Samuel, C. Dean, and D. Lyle. The logistic random field – a convenient graphical model for learning parameters for mrf-based labeling. *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008.

[15] Z. Tu, C. Narr, P. Dollar, I. Dinov, P. Thompson, and A. Toga. Brain anatomical structure segmentation by hybrid discriminative/generative models. *IEEE Trans. on Medical Imaging*, 27:495–508, 2008.

[16] B. G. M. S. Y. Zheng, A. Barbu and D. Comaniciu. Four-chamber heart modeling and automatic segmentation for 3-d cardiac ct volumes using marginal space learning and steerable features. *IEEE Trans. on Medical Imaging*, 27:1668–1681, 2008.

[17] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14(1):185205, 2005.

Appendix

Formulae for gradients

The gradients in eq.s (8-9) are as follows:

$$\begin{aligned}&\nabla_\phi J(\phi,\theta \mid x)\\ =\;& \left(f(T^1_{\theta_1}(x)) - \lambda_1 \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - 2\lambda_2(H(\phi) - s(T^2_{\theta_2}))\right)\\ &\times\delta_0(\phi)\end{aligned}$$

and,

$$\begin{aligned}&\nabla_\theta J(\phi,\theta \mid x)\\ =\;& \sum_{u\in\Omega}\left\{H(\phi(u)) - \frac{\exp(f(T^1_{\theta_1}(x(u))))}{1+\exp(f(T^1_{\theta_1}(x(u))))}\right\}\\ &\times\frac{\partial f(T^1_{\theta_1}(x(u)))}{\partial\theta_1}\\ &+2\lambda_2\sum_{u\in\Omega}(H(\phi(u)) - s(T^2_{\theta_2}(u)))\nabla s\frac{\partial(T^2_{\theta_2})}{\partial\theta_2} - 2\lambda(\theta-\theta^*),\end{aligned}$$

where, ∇s is the gradient of s expressed as a row vector and assuming Gaussian radial basis functions for the kernel K,

$$\begin{aligned}&\frac{\partial f(T^1_{\theta_1}(x(u)))}{\partial\theta_1} =\\ &\frac{1}{\sigma_r^2}\sum_{i=1}^{M}\alpha_i K(T^1_{\theta_1}(x(u)),\tilde{x}_i)(\tilde{x}_i - T^1_{\theta_1}(x(u)))\frac{\partial T^1_{\theta_1}(x(u))}{\partial\theta_1}.\end{aligned}$$

where, $\frac{\partial T^1_{\theta_1}(x(u))}{\partial\theta_1}$ and $\frac{\partial T^2_{\theta_2}(x(u))}{\partial\theta_2}$ are the partial differential matrices mentioned at the end of section 4.1.