Discriminative Learning for Deformable Shape Segmentation: A Comparative Study

Jingdan Zhang[1], Shaohua Kevin Zhou[1], Dorin Comaniciu[1], and Leonard McMillan[2]

[1] Integrated Data Systems Department, Siemens Corporate Research
Princeton, NJ 08540, USA

[2] Department of Computer Science, UNC Chapel Hill
Chapel Hill, NC 27599, USA
{jingdan.zhang, shaohua.zhou, dorin.comaniciu}@siemens.com, mcmillan@cs.unc.edu

Abstract. We present a comparative study on how to use discriminative learning methods such as classification, regression, and ranking to address deformable shape segmentation. Traditional generative models and energy minimization methods suffer from local minima. By casting the segmentation into a discriminative framework, the target fitting function can be steered to possess a desired shape for ease of optimization yet better characterize the relationship between shape and appearance. To address the high-dimensional learning challenge present in the learning framework, we use a multi-level approach to learning discriminative models. Our experimental results on left ventricle and left atrium segmentation from ultrasound images and facial feature point localization demonstrate that the discriminative models outperform generative models and energy minimization methods by a large margin.

1 Introduction

Deformable shape segmentation is a long-standing challenge in computer vision and medical imaging. The challenge arises from mainly two aspects: (i) modeling deformable shape and (ii) characterizing the relationship between shape and appearance. Segmentation algorithms must address both aspects successfully. In the paper, we study the latter from a discriminative learning perspective.

Deformable shape can be represented either explicitly or implicitly. Explicit shape representation includes parametric curve/mesh [1], landmark-based model [2], etc. Implicit representation includes level set [3], M-rep [4], etc. In the paper, we focus on landmark-based explicit representation, in which prior knowledge about the shape can be encoded by principal component analysis (PCA), similar to active shape model (ASM) [2]. To have a robust segmentation performance, prior knowledge is important to dealing with the complex shape variations by constraining the shape deformation to a compact model space spanned by a few parameters (*e.g.,* the dominant principal components.). For other shape representations, prior knowledge can be encoded differently, e.g. using bending energy, minimum curve length, etc.

Shape segmentation can be considered as seeking in the model space the shape model best fitting an image. In order to determine how well a hypothesis model matches the image, we need to build a fitting function to characterize the relationship between shape and image appearance. A good fitting function should satisfy two requirements. First, the function can well differentiate the correct solution from its background in the model space. Second, the function can effectively guide the search algorithms to the correct solution.

The fitting function can be learned from a given set of example images with ground truth annotations. In previous work, generative models are commonly used in learning; examples of generative models include ASM [2] and active appearance model (AAM) [5]. Generative models learn the relationship between the ground truth shapes and their appearances to characterize the underlying generating process of data population. However, they have difficulty in satisfying the two requirements mentioned above. A generative function does not typically represent the background and therefore is sub-optimal in differentiating the ground truth shapes from their background. Also generative learning has difficulty in controlling the overall shape of the fitting function. The learned functions often have local extremes, cause difficulties for optimization algorithms.

The fitting function can be also constructed as the energy function in an energy minimization approach, such as active contour model [1] and level set algorithm [6]. The local shape priors, including the elasticity and stiffness of the shape, are crafted into energy functions, which unfortunately also suffer from the same problems as before. They cannot guarantee to produce the lowest energy at the ground truth position and they are likely to have local minima around strong image edges.

Given sufficient training examples, discriminative learning approaches can provide better fitting functions. Discriminate learning has been applied successfully to object detection applications [7–9], in which the problem is formulated as a classification problem. In training, the image patches containing the target object are considered as positives, and the patches containing background as negatives. In testing, the target object is detected by scanning, using the trained classifier, the test image over an exhaustive range of similarity transformation. The computational load of the classification approach is proportional to the dimensionality of parameter space. The main challenge of extending the classification approach to deformable shape segmentation is the high dimensionality of the model space, which makes the exhaustive search prohibitive. In [10], a fitting function is learned using a classification method and the learned function is optimized via gradient ascent. The learned fitting function is not smooth and may have local maxima, making it difficult for gradient ascent to find the correct solution.

Recently, discriminative learning has been incorporated into generative models or energy functions to improve the segmentation performance. In [11], boundary detectors are trained to replace the generative models in ASM for better locating the boundary of heart chambers. In [12], a foreground/background classifier is plugged into an energy function to provide the evidence of whether the current pixel belongs to the target object or not. The fitting functions built by these approaches improve the segmentation results. However, they could still have local extremes.

A regression based approach was proposed to learn the fitting function [13]. The target fitting function is constrained to be unimodal and smooth in the model space, which can be used by local optimization algorithms to efficiently estimate the correct solution. The algorithm demonstrated superior performance on segmenting corpus callosum border from clean/noisy images, segmenting the left ventricle endocardial wall from echocardiogram, and localizing facial feature points.

In this paper, we present a comparative study on how to apply three discriminative learning approaches – classification, regression, and ranking – to deformable shape segmentation. By using discriminative learning in the model space, the fitting function can be learned in a steerable manner. We discuss how to extend the classification approach from object detection to deformable object segmentation. We also propose a ranking based approach to learning the fitting function: The fitting function is trained to produce the highest score around the ground truth and also possesses a desired shape to guide optimization algorithms to the ground truth. To address the high-dimensional learning challenge present in the learning framework, we apply a multi-level approach to learn discriminative models. In section 2, we discuss how to solve the segmentation via classification, regression, and ranking approaches. We also compare these three approaches in terms of learning complexity and computational cost at the segmentation stage. In section 3, we address the common challenge of the three approaches, namely learning in a high dimensional model space. In section 4, we detail the ranking based approach. In section 5, we compare the performance the three approaches on various test datasets.

2 Discriminative learning approaches

A shape in an image I can be parameterized by a set of continuous model parameters, C, which contains both rigid and non-rigid components. Given an image I and a hypothesis model C, we can extract a feature image $x(I, C)$ to describe the image appearance associated with C. For conciseness, we use x instead of $x(I, C)$ when there is no confusion in the given context.

There are a variety of ways of building shape models and computing feature images. Though the discriminative learning approaches presented are not bound to a specific shape model, in our implementation we represent a shape by a set of control points. A shape model is built by aligning the training shapes using the generalized Procrustes analysis [2] and applying PCA to the aligned shapes. The model C is defined as $(t_x, t_y, \theta, s, b_1, b_2, \ldots)$, including pose parameters (2D translation, rotation, and scale) and shape parameters corresponding to a reduced set of eigenvectors associated with the largest eigenvalues. We follow the strategy proposed in [13] to obtain a feature image $x(I, C)$ for computational efficiency. For the shape C with M control points, the feature image x is composed by $M + 1$ image patches cropped from I, as shown in Fig. 1. Other approaches involving more computations are can also be used, such as image warping based on linear interpolation [5] or thin plate spline [14].

A supervised learning approach attempts to train a fitting function $f(x(I, C))$ based on a set of training images $\{I\}$ and their corresponding ground truth shape models $\{\underline{C}\}$. The desired output of f is specific to the discriminative approach.

Classification

A classification approach is to learn a classifier f to indicate whether a hypothesis shape C correctly represents the one in an image I or not. The desired output y of f is

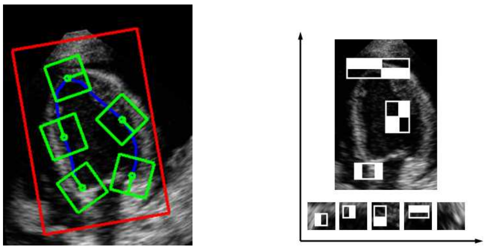

Fig. 1. The feature image x associated with a hypothesis model C. The contour represented by the model C is plotted as blue line. The subimage enclosed by the red box contains global fitness information. The subimges enclosed by the green boxes contain the local fitness information. The image x is composed by subimages with normalized orientation as shown on the right. Examples of Haar-like features are also shown in x.

a binary value. Whether a feature image $x(I, C)$ is positive or negative is determined by the distance between C and the ground truth model $\underline{C}$ in the image I:

$$y = \begin{cases} 1 & \text{if } \parallel C - \underline{C} \parallel \leq \epsilon \\ -1, & \text{otherwise} \end{cases}, \tag{1}$$

where ϵ is a threshold that determines the aperture of f. The learned $f(x(I, C))$ is like a boxcar function around the ground truth. Fig. 2(A) shows an ideal learned function f when C is one dimensional. Because the learned f only provides binary indication, the exhaustive search is necessary to estimate the solution, which is computationally prohibitive when the dimensionality of the model C is high.

Regression

A regression approach is to learn a regressor f with real-valued output to indicate the fitness of a hypothesis model C to an image I. The desired output y of f can be designed to facilitate the searching process at the testing stage. In [13], y is set to be a normal distribution:

$$y = \mathcal{N}(C; \underline{C}, \Sigma), \tag{2}$$

where Σ is a covariance matrix determining the aperture of f. The ideally learned f has a smooth and unimodal shape, *e.g.*, a 1D example shown in Fig. 2(B). The function f learned in this way can be effectively optimized by general-purpose local optimization techniques, such as gradient descent or simplex, due to the guidance provided by the gradient of f. However, compared with a classification approach, the desired output is more complicated and, hence, more information needs to be learned at the training stage as it requires the regressor to produce a desired real value for each point in the model space. Learning a regressor in a high-dimensional model space is challenging. Recently, an image-based regression algorithm using boosting methods was proposed in [15] and successfully applied to different applications [16, 13].

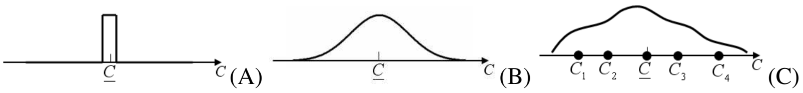

Fig. 2. The learned $f(I, C)$ when C is one dimensional: (A) a classification approach, (B) the regression approach in [13], and (C) the ranking approach. The ground truth of the model is $\underline{C}$.

Ranking

Discriminative learning via ranking is originally proposed to retrieve information based on user preference [17]. In segmentation applications, ranking approaches are used to retrieve candidate shapes from the shape database containing example shapes [18, 14].

In this paper, we propose a ranking approach to learning partial ordering of points in the model space. The ordering learned by the ranking function provides essential information to guide the optimization algorithm at the testing stage. Unlike a regression approach, which enforces the regressor to produce an exact value at each point in the model space, ranking only tries to learn relative relations of paired points in the model space. Let (C_0, C_1) be a pair of points in the model space and its associated feature image pair (x_0, x_1). The ordering of x_0 and x_1 is determined by their shape distance to the ground truth: the one closer to the ground truth has a higher order. We learn a ranking function f to satisfy the constraint:

$$\begin{cases} f(x_0) > f(x_1) & \text{if } \| C_0 - \underline{C} \| < \| C_1 - \underline{C} \| \\ f(x_0) < f(x_1) & \text{if } \| C_0 - \underline{C} \| > \| C_1 - \underline{C} \| \\ f(x_0) = f(x_1), & \text{otherwise} \end{cases} \tag{3}$$

Fig. 2(C) illustrates the basic idea of the ranking approach. There are five points in the 1D model space and $\underline{C}$ is the ground truth. At the training stage, a ranking function f is learned to satisfy the ordering constrains: $f(x(I, \underline{C})) > f(x(I, C_2))$, $f(x(I, C_2)) > f(x(I, C_1))$, $f(x(I, \underline{C})) > f(x(I, C_3))$, and $f(x(I, C_3)) > f(x(I, C_4))$. Similar to the regression approach [13], the learned ranking function f is unimodal, which is desired for local optimization techniques. However, the amount of information to be learned in ranking is less than the one in regression. The regression approach learns the full ordering of points in the model space, while the ranking approach only learns partial, pairwise ordering.

We employ the boosting principle to learn our ranking function by selecting relative features to form an additive committee of weak learners. Each weak learner, based on a Haar-like feature that can be computed rapidly, provides a rough ranking. The learned ranking function combines the rough ranking from weak learners and provides the robust ordering information in the shape model space. We will discuss the detailed implementation of the ranking algorithm in section 4.

3 Learning in a high dimensional space

The first step toward learning a discriminative function is to sample training examples in the model space. Due to the curse of dimensionality, the number of training examples should be exponential to the model dimensionality to ensure training quality. This poses a big challenge to apply discriminative learning for deformable segmentation applications, in which the dimensionality of the model space is usually high. Another challenge is that it is increasingly difficult to discriminate the correct solution from its background, when the background points get closer to the solution. In this situation, the image appearance of the background points becomes more and more similar to that of the correct solution. Due to these two challenges, learning a single function in the whole model space to accurately distinguish the solution from its background is ineffective.

We use the multi-level approach proposed in [13] to learn a series of discriminative functions f_k, $k = 1, \dots, K$, each of which focusing on a region that gradually narrows down to the ground truth. Let $\varOmega_k$ be the focus region of f_k in the model space, which is defined within an ellipsoid centered at the ground truth:

$$\varOmega_k = \{C = (c_1, c_2, ..., c_Q) | \sum_{q=1}^{Q} (c_q - \underline{c}_q)^2 / r_{k,q}^2 \leq 1\} \tag{4}$$

where Q is the dimensionality of the model space and $R_k = (r_{k,1}, \dots, r_{k,Q})$ defines the range of the focus region. The focus regions are designed to have a nested structure gradually shrinking to the ground truth:

$$\varOmega_1 \supset \varOmega_2 \supset \dots \supset \varOmega_K \supset \underline{\varOmega} \ni \underline{C}, \tag{5}$$

where $\varOmega_1$ defines the initial region of the model parameters. It should be big enough to include all the possible solutions in the model space. The final region $\underline{\varOmega}$ defines the desired segmentation accuracy.

In segmentation applications, the initial focus region $\varOmega_1$ is highly elongated due to the variation in parameter range. It is desirable to first decrease the range of the parameters with a large initial range. The evolution of the range is designed as:

$$r_{k+1,q} = \begin{cases} r_k^{\max}/\gamma & \text{if } r_{k,q} > r_k^{\max}/\gamma \\ r_{k,q} & \text{otherwise} \end{cases}, \tag{6}$$

where $r_k^{\max}$ is the largest value in R_k and γ is a constant controlling the shrinking rate of focus regions (we empirically set $\gamma = 2.9$ for all experiments). Geometrically, the region gradually shrinks from a high-dimensional ellipsoid to a sphere, and then shrinks uniformly thereafter. Fig. 3(A) shows the evolution of the focus regions in a 2D example.

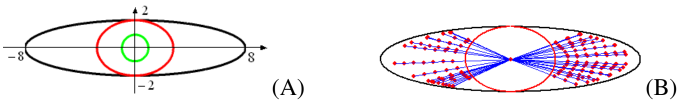

Fig. 3. (A) The three nested focus regions defined by R_1(black), R_2(red) and R_3(green), (B) The result of 2D sampling used for the ranking approach. The lines connect the points on the same ray.

At the testing stage, we apply optimization algorithms sequentially to the learned functions to refine the segmentation results. At the kth stage, we want the solution fallen within the region Ω_k to be pushed into the region Ω_{k+1}. In order to achieve this, the learned function f_k should be able to differentiate the instances in the region Ω_{k+1} from those in the region $\Omega_k - \Omega_{k+1}$ and provide effective guidance to the optimization algorithms especially in the region $\Omega_k - \Omega_{k+1}$. Data sampling strategies should be accordingly designed.

For the classification approach, the positive examples are sampled from the region Ω_{k+1} and the negatives from the region $\Omega_k - \Omega_{k+1}$. The choice of shrinking rate γ is balanced by two factors. A large γ means a small positive region which requires a fine search grid to detect the solution at the testing stage. This causes high computational expense at the testing stage. On the other hand, a small γ means a large positive region in which the image appearance of the instances has large variation. This causes confusion to the classifier at the training stage.

For the regression approach, gradient sampling is proposed [13]: the learned regressors provide guidance to optimization algorithms based on local gradient. Because the regressor f_k has large gradient in the region $\Omega_k - \Omega_{k+1}$, more training examples are drawn from the region $\Omega_k - \Omega_{k+1}$ to insure the training quality in this region.

The ranking approach is to learn the partial ordering of instances in the model space. Because we perform a line-searching type of optimization, the ordering of instances along the rays starting from the ground truth is the most important. This ordering provides the essential information to guide the optimization algorithms to the ground truth. Also by learning the ordering information from enough rays, the learned ranking function is unimodal which has a global optimum at the ground truth.

We propose a sampling algorithm to sample training pairs for training the ranking function f_k. First, we select a ray starting from the ground truth with random direction. Second, we sample $J+1$ points $\{C_0, C_1, \ldots, C_J)\}$ on the selected ray, where C_0 is at the ground truth and the remaining J points are sampled from the line segment in the region $\Omega_k - \Omega_{k+1}$. These points are ordered based on the distance to the ground truth. The parameter J is proportional to the length of the line segment. The reason of sampling only from the line segment is that the ordering on this part of the ray is most important for training f_k, which is used to push the solution from the region $\Omega_k - \Omega_{k+1}$ to Ω_{k+1}. Finally, from the training image I, we draw J pairs of training examples $\{(x(I, C_j), x(I, C_{j-1})), j = 1, \ldots, J)\}$, where $x(I, C_{j-1})$ should be ranked above $x(I, C_j)$. We continue this process to sample as much rays as possible that can be fitted into computer memory. Fig. 3(B) shows the sampling result in a 2D model space.

4 Ranking using boosting algorithm

In this section, we present a ranking algorithm based on RankBoost [17] to learn the ordering of the sampled image pairs. Mathematically, we learn a ranking function that minimizes the number of image pairs that are mis-ordered by the learned function. Let Φ be the sampled training set and $(x_0, x_1) \in \Phi$ be a pair of images. Following the sampling strategy proposed in the previous section, x_1 should be ranked above x_0, otherwise a penalty $D(x_0, x_1)$ is imposed. We use equal weighted penalty $D(x_0, x_1)$ in our experiments. The penalty weights can be normalized over the whole training set to a probability distribution $\sum_{(x_0,x_1)\in\Phi} D(x_0, x_1) = 1$.

```
1. Given: initial distribution $D$ over $\Phi$.
2. Initialize: $D_1 = D$.
3. For $t = 1, 2, \ldots, T$
   – Train weak learner using distribution $D_t$ to get weak ranking $g_t$.
   – Choose $\alpha_t \in \mathbf{R}$.
   – Update: $D_{t+1}(x_0, x_1) = \frac{D_t(x_0,x_1)\exp[\alpha_t(g_t(x_0)-g_t(x_1))]}{Z_t}$ where $Z_t$ is a normalization
     factor (chosen so that $D_{t+1}$ will be a distribution.)
4. Output the final ranking: $f(x) = \sum_{t=1}^{T} \alpha_t g_t(x)$.
```

Fig. 4. The RankBoost algorithm [17].

The learning goal is to search for a ranking function f that minimizes the ranking loss,

$$rloss_D(f) = \sum_{(x_0,x_1)\in\Phi} D(x_0, x_1)[\![f(x_0) \geq f(x_1)]\!], \tag{7}$$

where $[\![\pi]\!]$ is defined to be 1 if the predicate π holds and 0 otherwise. In RankBoost, the ranking function $f(x)$ takes an additive form:

$$f_t(x) = f_{t-1}(x) + \alpha_t g_t(x) = \sum_{i=1}^{t} \alpha_i g_i(x), \tag{8}$$

where each $g_i(x)$ is a weak learner residing in a dictionary set $\mathcal{G}$. It maps a feature image x to a real-valued ranking score. The strong learner $f(x)$ combines the weighted ranking scores from weak learners to obtain a robust ranking. Boosting is used to iteratively select weak learners by leveraging the additive nature of $f(x)$: at iteration t, one more additive term $\alpha_t g_t(x)$ is added to the ranking function $f_{t-1}(x)$. The weak learner $g_t(x)$ is selected from the set $\mathcal{G}$ and its associated weight α_t is computed to minimize the ranking loss

$$(g_t, \alpha_t) = \arg \min_{g\in\mathcal{G},\alpha\in\mathbf{R}} \sum_{(x_0,x_1)\in\Phi} D(x_0, x_1)[\![f_{t-1}(x_0)+\alpha g(x_0) \geq f_{t-1}(x_1)+\alpha g(x_1)]\!]. \tag{9}$$

The RankBoost algorithm is shown in Fig. 4. We now discuss the choice of weak learners below.

Weak learner

The input to a weak learner is a feature image x. We use Haar-like features as primitives to construct the dictionary set $\mathcal{G}$. Each weak learner $g(x)$ is associated with a Haar-like feature $h(x; \eta)$, where η specifies the attribute of the feature, including feature type and window position/size. We further restrict that the features must be contained within one of the image patches in x. Fig. 1 shows some examples of the Haar-like features. By choosing Haar-like features with different attributes, we obtain the over-complete feature representation of the image x. These features can be computed rapidly using a set of pre-computed integral images with different orientations [7].

For each feature $h(x;\eta)$, we use a 1D binary function $g(x;\eta)$ with L bins to produce a weak ranking:

$$g(x;\eta) = \beta_l; \;\; \text{if } h(x;\eta) \in (u_{l-1}, u_l] \tag{10}$$

where $\{u_l; l = 0, \ldots, L\}$ evenly divide the output range of the feature $h(x;\eta)$ to L bins and $\beta_l \in \{0, 1\}$ is the value of the lth bin.

Based on the discussion in [17], when the output of a weak learner has range $[0, 1]$, the weak learner should be trained to maximize the function:

$$r = \sum_{(x_0,x_1)\in\Phi} D(x_0, x_1)(g(x_1;\eta) - g(x_0;\eta)). \tag{11}$$

Following the definition in (10), the function r can be rewritten as

$$r = \sum_{l=1}^{L} \beta_l e_l, \;\; e_l := \sum_{h(x_1)\in(u_{l-1},u_l]} D(x_0, x_1) - \sum_{h(x_0)\in(u_{l-1},u_l]} D(x_0, x_1). \tag{12}$$

It is easy to show that in order to maximize r, the lth bin value should be determined as:

$$\beta_l = \{ \begin{matrix} 1 & \text{if } e_l > 0 \\ 0, & \text{otherwise} \end{matrix}, \tag{13}$$

At each boosting iteration, we exhaust all weak learners and find the one that produces the largest r value. Then the associated weight α is computed as

$$\alpha = \frac{1}{2} \ln \left(\frac{1 + r_{\max}}{1 - r_{\max}} \right). \tag{14}$$

5 Experiments

We tested the proposed discriminative learning approaches on three problems. The boosting principle is used in all three approaches to select and combine weak learners. For classification, we implemented the cascade of boosted binary classifiers based on Adaboost [7], which has been successfully applied to fast object detection. For regression, the regression based on boosting proposed in [13] was applied. To fairly compare these three algorithms, we used the same set of Haar-like features discussed in Section 4. We also compared the three algorithms with other alternative approaches, such as ASM [2] and AAM [5]. In order to enhance the performance of ASM, we also implemented an enhanced ASM version that replaces the regular edge computation by boundary classifiers, which is similar to the approach proposed in [11, 12].

To handle the challenge of learning in a high dimensional space, we used the multilevel approach to learn a series of discriminative functions. In training, the initial error ranges of the parameters are set to control the sampling range. The initial error ranges of the shape parameters are assumed to be $3\sqrt{\lambda}$, where λ's are eigenvalues from PCA. We sampled as many training examples as computer memory allows. For our computer with 2GB memory, about 400K training examples are used. In testing, for each test image we randomly generated an initial contour, whose pose parameters are within the error range

(A) LV	ASM	enhanced ASM	Classification	Regression	Ranking
level 1	n/a	n/a	15.85±5.51 15.15±4.65	11.09±4.31 10.43±3.11	10.12±3.26 9.69±2.69
final level	26.20±17.64 23.43±12.03	17.91±6.80 17.07±5.82	14.77±6.53 13.86±5.25	10.07±4.52 9.41±3.06	9.93±3.56 9.37±2.62
time (s)	0.94	1.43	18.7	3.15	2.86
(B) LA	ASM	enhanced ASM	Classification	Regression	Ranking
level 1	n/a	n/a	16.37±5.96 15.60±4.95	14.72±13.59 12.10±4.02	11.66±6.14 10.79±3.69
final level	30.14±17.72 28.01±12.44	18.66±10.09 16.93±5.83	15.95±6.78 15.17±5.95	14.12±16.04 11.03±4.64	11.40±6.75 10.44±4.14
time (s)	0.75	1.26	18.1	2.80	2.18
(C) AR	AAM		Classification	Regression	Ranking
level 1	n/a		18.28±7.07 17.23±5.29	17.40±6.34 16.59±5.30	14.80±5.63 13.99±4.49
level 2	n/a		12.94±6.11 11.88±3.60	8.39±4.09 7.72±1.92	6.84±2.48 6.47±1.87
final level	19.70±23.83 15.87±17.23		11.66±6.77 10.53±4.35	5.76±3.99 5.10±1.38	5.79±2.95 5.31±2.07
time (s)	0.91		29.4	4.70	3.49

Table 1. *The mean and standard deviation of the segmentation errors. In each cell, there are two rows: the first row reports the mean and standard deviation obtained using all testing data and the second row using 95% of testing data (excluding 5% outliers). For ASM and AAM, we applied multi-resolution searching but only reported the benchmarks of the final results.*

defined in training and shape parameters are zeros (mean shape). Starting from an initial solution, the learned functions are sequentially applied to refine the solution. For the classification approach, we exhaustively searched around the initial solution and found the candidate having the highest classification probability as the starting point for the next level. For the regression and ranking approaches, we used the simplex optimization method [19] due to its tolerance to shallow maxima caused by image noise.

5.1 Endocardial wall segmentation: left ventricle

Segmenting the endocardial wall in echocardiographic images is a challenging task due to the large shape/appearance variation of the heart chambers and signal dropouts in images. In [13], the fitting functions trained by regression are used to locate the endocardial wall of the left ventricle (LV) in an apical four chamber (A4C) view. In this experiment, we followed the exact setting in [13].

The data set has 528 A4C images from different patients. The LV walls are annotated by experts using contours with 17 control points. The size of the LV in an image is roughly 120×180 pixels. Half of the dataset is used for training and the remaining half for testing. The initial range of the pose is set as $[50, 50, \pi/9, 0.2]$, which means 50 pixels in translation, 20 degrees in rotation, and 20% in scale in the extreme. The model C includes five shape parameters which account for 80% of the total shape variation. In training, two levels of discriminative functions are learned.

The segmentation error is defined as the average Euclidean distance between corresponding control points of the segmented shape and the ground truth. In testing, we set the initial pose of a testing image to be a random perturbation of the ground truth within the initial range $[50, 50, \pi/9, 0.2]$. The average initial error of all testing images

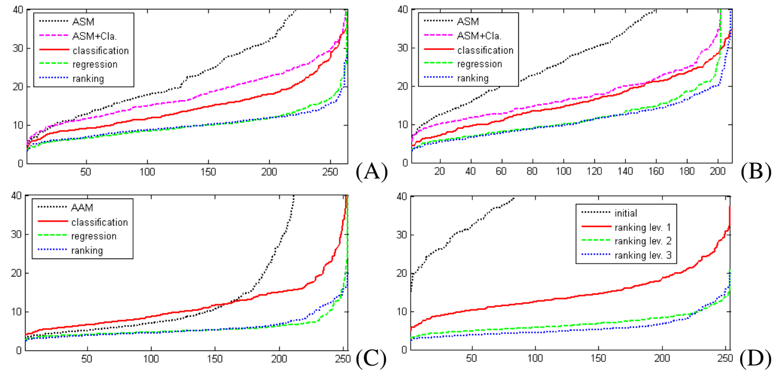

Fig. 5. Sorted errors of the experiment results. The horizontal axes are testing numbers and vertical axes are segmentation errors. (A) LV segmentation, (B) LA segmentation, (C) facial feature localization, and (D) the errors of the multi-level refinement using the ranking approach in the third experiment.

is 27.16 pixels. We tested the classification, regression and ranking algorithms, along with ASM and its enhanced version. Table 1(A) shows the mean and standard deviation of the test errors of the above algorithms. The average computational time is also listed in the table. Fig. 5(A) is a plot of the sorted errors, where points on the curve with the same horizontal position do not necessarily correspond to the same test case. Fig. 6 shows some segmentation results using the ranking approach.

5.2 Endocardial wall segmentation: left atrium

In this experiment, we tested the algorithms on segmenting the endocardial wall of the left atrium (LA) in the apical two chamber (A2C) view. LA segmentation is even harder than LV segmentation. The LA appearance is more noisy because the LA, being in the far field of the ultrasound probe, is more difficult to image. We collected 417 A2C images with the LA walls annotated by experts using 17 control points. The LA roughly occupies 120×120 in an image. We used 208 images in training and remaining 209 images in testing. The initial range of the pose is set as $[50, 50, \pi/9, 0.2]$. The model C includes four shape parameters which account for 88% of the total shape variation. In training, we trained two levels of discriminative functions. In testing, the average initial error is 26.82 pixels. We used the same experimental setting as in the LV segmentation. The benchmarks are shown in Table 1(B) and Fig. 5(B).

5.3 Facial feature localization

In the third experiment, we tested the performance of the algorithms on the AR face database [20]. There is a total of 508 images with annotations which include 22 control

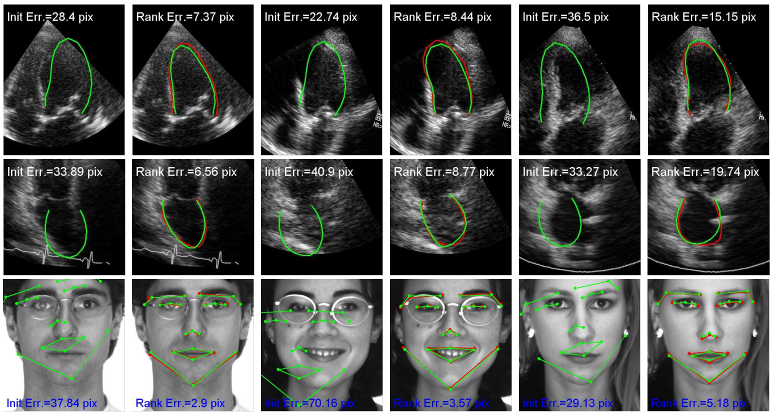

Fig. 6. Segmentation results with a variety of errors obtained by the ranking approach. For each pair of images, the left one shows the initial contour position and the right shows the segmentation results after the multi-level refinement. The initial positions and the results are green line. The ground truths are red line.

points[3]. The color images were converted to gray-scale. The size of a face is roughly 250×300 in an image. We used half of the data for training and half for testing. Examples of the same subject were not used in both training and testing data. The initial range of the pose is $[100, 100, \pi/9, 0.2]$. The model C includes 5 shape parameters which account for 73% of the total shape variation. In training, we trained three levels of discriminative functions. In testing, the average initial error is 47.19 pixels. We used AAM [21] for comparison. The benchmarks are shown in Table 1(C) and Fig. 5(C). Fig 5(D) shows the errors after each level of refinement when using the ranking functions in testing.

Fig. 7 shows the 2D slices of learned classifiers and ranking functions on a testing image. These slices are obtained by varying the 1st and the 5th parameters of the model in the error range while fixing the remaining parameters as the ground truth, where the 1st is a translation parameter and the 5th is a shape parameter corresponding to the largest eigenvalue. The learned functions have desired shapes which make optimization algorithms perform well on this testing image.

5.4 Discussion

In the three experiments, the segmentation algorithms using discriminative fitting functions consistently outperform the previous algorithm by a large margin. The performance of the ASM algorithm is boosted by using discriminative boundary classifiers; however, it still suffers from the local extremes because the boundary classifier is local. The relative poor performance of the classification approach is due to the coarse search

[3] The annotations are provided by Dr. Cootes, which is available at http://www.isbe.man.ac.uk/˜bim.

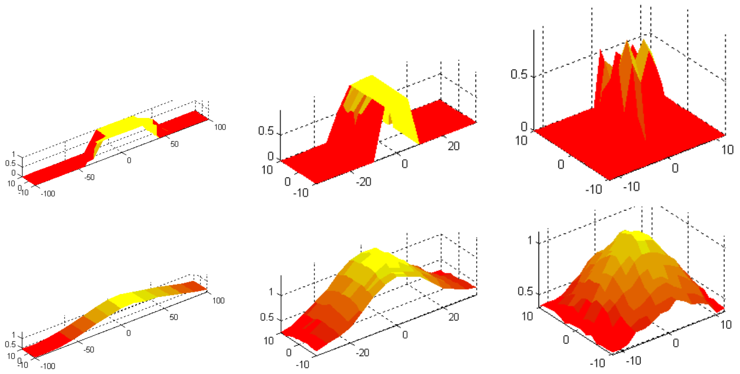

Fig. 7. The 2D slices of the learned classifiers (top row) and ranking functions (bottom row) on a testing image. The left column shows the first level, the middle shows the second level, and the right shows the third level.

grid in exhaustive search. If we use a fine search grid, the segmentation accuracy is expected to improve. It is interesting to see the performance of the algorithms specifically designed to train classifiers in a high dimensional model space, such as marginal space learning [11]. The ranking approach outperforms the regression approach, especially in the challenging situation, such as LA segmentation. The main reason might be that ranking only attempts to learn a partial ordering information in the model space and hence its learning complexity is lower than regression. We will verify this as future work. Like all discriminative learning problems, the discriminative learning approaches suffer from the problem of overfitting especially when the variation of training data cannot totally covers that of testing. Further, the number of sampled data points is hardly sufficient when the model space is high. Because of these problems, the fitting function does not have desired shape on some test data and the local optimization algorithm fails to converge to the ground truth.

Recently, a ranking based algorithm for face alignment was independently proposed [22]. It presents a ranking approach to learning an alignment score function and compares with a classification based algorithm [10]. Compared with the method in [22], we use a different ranking algorithm and apply the multi-level approach to improve the segmentation accuracy. We also compare more algorithms on different kind of data.

6 Conclusions

We have presented a discriminative learning framework for deformable shape segmentation and shown that all three discriminative methods, classification, regression, and ranking, can be applied. We have also addressed how to sample a high-dimensional space and proposed a RankBoost algorithm that does feature selection. Finally, we have demonstrated that the discriminative models outperform generative models and energy minimization methods by a large margin in our experiments on segmentation of left ventricle and left atrium from ultrasound images and facial feature point localization. In the future, we will further investigate how to sample a high-dimensional space more efficiently and extend this framework to arbitrary shape representations.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Computer Vision **1(4)** (1988) 321–331
2. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models–their training and application. Computer Vision and Image Understanding **61(1)** (1995) 38–59
3. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. Journal of Computation Physics **79** (1988) 1249
4. Pizer, S., Fletcher, P., Joshi, S., Thall, A., Chen, Z., Fridman, Y.: Deformable m-reps for 3D medical image segmentation. Int. J. Computer Vision **55** (2003) 85–106
5. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. Pattern Anal. Machine Intell **23(6)** (2001) 681–685
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Computer Vision **22(1)** (1997) 61–79
7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR. (2001)
8. Georgescu, B., Zhou, X.S., Comaniciu, D., Gupta, A.: Database-guided segmentation of anatomical structures with complex appearance. In: Proc. CVPR. (2005)
9. Zhang, J., Zhou, S., McMillan, L., Comaniciu, D.: Joint real-time object detection and pose estimation using probabilistic boosting network. In: Proc. CVPR. (2007)
10. Liu, X.: Generic face alignment using boosted appearance model. In: Proc. CVPR. (2008)
11. Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D.: Fast automatic heart chamber segmentation from 3D ct data using marginal space learning and steerable features. In: Proc. ICCV. (2007)
12. Tu, Z., Zhou, X.S., Comaniciu, D., Luca, B.: A learning based approach for 3D segmentation and colon detagging. In: Proc. European Conf. Computer Vision. (2006)
13. Zhang, J., Zhou, S., Comaniciu, D., McMillan, L.: Conditional density learning via regression with application to deformable shape segmentation. In: Proc. CVPR. (2008)
14. Zheng, Y., Zhou, X.S., Georgescu, B., Zhou, S., Comaniciu, D.: Example based non-rigid shape detection. In: Proc. European Conf. Computer Vision. (2006)
15. Zhou, S., Gerogescu, B., Zhou, X., Comaniciu, D.: Image-based regression using boosting methods. In: Proc. ICCV. (2005)
16. Zhou, S., Comaniciu, D.: Shape regression machine. In: Proc. IPMI. (2007)
17. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Machine Learning Research **4(6)** (2004) 933–970
18. Athitsos, V., Alon, J., S. Sclaroff, G.K.: Boostmap: A method for efficient approximate similarity rankings. In: Proc. CVPR. (2004)
19. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical recipes in C (2nd edition), Cambridge University Press (1992)
20. Martinez, A.M., Benavente, R.: The AR face database. (1998)
21. Stegmann, M.B., Ersboll, B.K., Larsen, R.: FAME–a flexible appearance modeling environment. IEEE Trans. Medical Imaging **22(10)** (2003) 1319–1331
22. Wu, H., Liu, X., Doretto, G.: Face alignment via boosted ranking model. In: Proc. CVPR. (2008)