Discriminative Parameter Estimation for Random Walks Segmentation

Pierre-Yves Baudin[1,2,3,4,5,6], Danny Goodman[1,2,3], Puneet Kumar[1,2,3], Noura Azzabou[4,5,6], Pierre G. Carlier[4,5,6], Nikos Paragios[1,2,3], and M. Pawan Kumar[1,2,3]

[1] Center for Visual Computing, École Centrale Paris, FR
[2] Université Paris-Est, LIGM (UMR CNRS), École des Ponts ParisTech, FR
[3] Équipe Galen, INRIA Saclay, FR
[4] Institute of Myology, Paris, FR
[5] CEA, I^2 BM, MIRCen, IdM NMR Laboratory, Paris, FR
[6] UPMC University Paris 06, Paris, FR

Abstract. The Random Walks (RW) algorithm is one of the most efficient and easy-to-use probabilistic segmentation methods. By combining contrast terms with prior terms, it provides accurate segmentations of medical images in a fully automated manner. However, one of the main drawbacks of using the RW algorithm is that its parameters have to be hand-tuned. we propose a novel discriminative learning framework that estimates the parameters using a training dataset. The main challenge we face is that the training samples are not fully supervised. Specifically, they provide a hard segmentation of the images, instead of a probabilistic segmentation. We overcome this challenge by treating the optimal probabilistic segmentation that is compatible with the given hard segmentation as a latent variable. This allows us to employ the latent support vector machine formulation for parameter estimation. We show that our approach significantly outperforms the baseline methods on a challenging dataset consisting of real clinical 3D MRI volumes of skeletal muscles.

1 Introduction[1]

The Random Walks (RW) algorithm is one of the most popular techniques for segmentation in medical imaging [5]. Although it was initially proposed for interactive settings, recent years have witnessed the development of fully automated extensions. In addition to the contrast information employed in the original formulation [5], the automated extensions incorporate prior information based on appearance [4] and shape [1].

It has been empirically observed that the accuracy of the RW algorithm relies heavily on the relative weighting between the various contrast and prior terms. Henceforth, we refer to the relative weights of the various terms in the RW objective function as parameters. At present, researchers either rely on a user to

[1] Supplementary materials at: `http://hal.inria.fr/hal-00830564`

K. Mori et al. (Eds.): MICCAI 2013, Part III, LNCS 8151, pp. 219–226, 2013.

hand-tune the parameters or on exhaustive cross-validation [1,4]. However, both these approaches quickly become infeasible as the number of terms in the RW objective function increase.

In contrast to the RW literature, the problem of parameter estimation has received considerable attention in the case of discrete models such as CRFs [9]. Recent years have witnessed the emergence of structured-output support vector machine (Structured SVM) as one of the most effective discriminative frameworks for supervised parameter estimation [10,11]. Given a training dataset that consists of pairs of input and their ground-truth output, structured SVM minimizes the empirical risk of the inferred output with respect to the ground-truth output. The risk is defined by a user-specified loss function that measures the difference in quality between two given outputs.

We would like to discriminatively learn the parameters of the RW formulation. To this end, a straightforward application of structured SVM would require a training dataset that consists of pairs of inputs as well as their ground-truth outputs—in our case, the *optimal* probabilistic segmentation. In other words, we require a human to provide us with the output of the RW algorithm for the best set of parameters. This is an unreasonable demand since the knowledge of the optimal probabilistic segmentation is as difficult to acquire as it is to hand-tune the parameters itself. Thus we cannot directly use structured SVM to estimate the desired parameters.

In order to handle the above difficulty, we propose a novel formulation for discriminative parameter estimation in the RW framework. Specifically, we learn the parameters using a weakly supervised dataset that consists of pairs of medical acquisitions and their hard segmentations. Unlike probabilistic segmentations, hard segmentations can be obtained easily from human annotators. We treat the optimal probabilistic segmentation that is *compatible* with the hard segmentation as a latent variable. Here, compatibility refers to the fact that the probability of the ground-truth label (as specified by the hard segmentation) should be greater than the probability of all other labels for each pixel/voxel. The resulting representation allows us to learn the parameters using the latent SVM formulation [3,8,12].

While latent SVM does not result in a convex optimization problem, its local optimum solution can be obtained using the iterative concave-convex procedure (CCCP) [13]. The CCCP involves solving a structured SVM problem, which lends itself to efficient optimization. In order to make the overall algorithm computationally feasible, we propose a novel efficient approach for ACI based on dual decomposition [2,7]. We demonstrate the benefit of our learning framework over a baseline structured SVM using a challenging dataset of real 3D MRI volumes.

2 Preliminaries

We will assume that the input $\mathbf{x}$ is a 3D volume. We denote the i-th voxel of $\mathbf{x}$ as $\mathbf{x}(i)$, and the set of all voxels as $\mathcal{V}$. In a hard segmentation, each voxel is assigned a label $s \in \mathcal{S}$ (for example, the index of a muscle). We will use $\mathbf{z}$ to

represent the human annotation (that is, the class labels of the voxels in $\mathbf{x}$) in binary form:

$$\mathbf{z}(i,s) = \begin{cases} 1 & \text{if voxel } i \in \mathcal{V} \text{ is of class } s \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In other words, the binary form $\mathbf{z}$ of the annotation specifies delta distribution over the putative labels for each voxel. Our training dataset is a collection of training images $\mathbf{x}$ and hard segmentations $\mathbf{z}$: $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{z}_k)\}_k$. Note that we use subscript k to denote the input index within a dataset, and parenthetical i to denote a voxel within a particular input.

2.1 Random Walks Segmentation

The RW algorithm provides a probabilistic—or soft—segmentation of an input $\mathbf{x}$, which we denote by $\mathbf{y}$, that is,

$$\mathbf{y}(i,s) = \Pr[\text{voxel } i \text{ is of class } s], \forall i \in \mathcal{V}, s \in \mathcal{S}. \tag{2}$$

When using one contrast term and one prior model, the RW algorithm amounts to minimizing the following convex quadratic objective functional:

$$E(\mathbf{y}, \mathbf{x}) = \mathbf{y}^\top L(\mathbf{x})\,\mathbf{y} + w^{\text{prior}} \|\mathbf{y} - \mathbf{y}_0\|^2_{\Omega_0(\mathbf{x})}, \tag{3}$$

$$= \mathbf{y}^\top L(\mathbf{x})\,\mathbf{y} + E^{\text{prior}}(\mathbf{y}, \mathbf{x}). \tag{4}$$

Here, $\mathbf{y}_0$ is a reference prior probabilistic segmentation dependent on appearance [4] or shape [1], and $\Omega_0(\mathbf{x})$ is a diagonal matrix that specifies a voxel-wise weighting scheme for $\mathbf{x}$. The term $L(\mathbf{x})$ refers to a combinatorial Laplacian matrix defined on a neighborhood system $\mathcal{N}$ based on the adjacency of the voxels. It is a block diagonal matrix—one block per label—with all identical blocks, where the entries of the block $L^b(\mathbf{x})$ use the typical Gaussian kernel formulation (see [5]). The relative weight w^{prior} is the parameter for the above RW framework. The above problem is convex, and can be optimized efficiently by solving a sparse linear system of equations. We refer the reader to [1,5] for further details.

2.2 Parameters and Feature Vectors

In the above description of the RW algorithm, we restricted ourselves to a single Laplacian and a single prior. However, our goal is to enable the use of numerous Laplacians and priors. To this end, let $\{L_\alpha\}_\alpha$ denote a known family of Laplacian matrices and $\{E_\beta(\cdot)\}_\beta$ denote a known family of prior energy functionals. In section 4, we will specify the family of Laplacians and priors used in our experiments. We denote the general form of a linear combination of Laplacians and prior terms as:

$$L(\mathbf{x}; \mathbf{w}) = \sum_\alpha w_\alpha L_\alpha(\mathbf{x}), E^{\text{prior}}(\cdot, \mathbf{x}; \mathbf{w}) = \sum_\beta w_\beta E_\beta(\cdot, \mathbf{x}), \mathbf{w} \geq 0. \tag{5}$$

Each term $E_\beta(\cdot, \mathbf{x})$ is of the form:

$$E_\beta(\mathbf{y}, \mathbf{x}) = \|\mathbf{y} - \mathbf{y}_\beta\|^2_{\Omega_\beta(\mathbf{x})}, \tag{6}$$

where $\mathbf{y}_\beta$ is the β-th reference segmentation and $\Omega_\beta(\mathbf{x})$ is the corresponding voxel-wise weighting matrix (which are both known). We denote the set of all parameters as $\mathbf{w} = \{w_\alpha, w_\beta\}_{\alpha,\beta}$. Clearly, the RW energy (4) is linear in $\mathbf{w}$, and can therefore be formulated as:

$$E(\mathbf{y}, \mathbf{x}; \mathbf{w}) = \mathbf{y}^T L(\mathbf{x}; \mathbf{w})\, \mathbf{y} + E_{\text{prior}}(\mathbf{y}, \mathbf{x}; \mathbf{w}) \,, \tag{7}$$

$$= \mathbf{w}^T \psi(\mathbf{x}, \mathbf{y}) \,, \tag{8}$$

where $\psi(\mathbf{x}, \mathbf{y})$ is known as the joint feature vector of $\mathbf{x}$ and $\mathbf{y}$. Note that by restricting the parameters to be non-negative (that is, $\mathbf{w} \geq \mathbf{0}$), we ensure that the energy functional $E(\cdot, \mathbf{x}; \mathbf{w})$ remains convex.

2.3 Loss Function

As mentioned earlier, we would like to estimate the parameters $\mathbf{w}$ by minimizing the empirical risk over the training samples. The risk is specified using a loss function that measures the difference between two segmentations. In this work, we define the loss function as the number of incorrectly labeled voxels. Formally, let $\hat{\mathbf{y}}$ denote the underlying hard segmentation of the soft segmentation $\mathbf{y}$, that is, $\hat{\mathbf{y}}(i, s) = \delta(s = \text{argmax}_{s \in \mathcal{S}} \mathbf{y}(i, s))$, where δ is the Kronecker function. The loss function is defined as

$$\Delta(\mathbf{z}, \mathbf{y}) = 1 - \frac{1}{|\mathcal{V}|} \hat{\mathbf{y}}^T \mathbf{z} \,, \tag{9}$$

where $\mathcal{V}$ is the set of all voxels, and $|\cdot|$ denotes the cardinality of a set.

3 Parameter Estimation Using Latent SVM

Given a dataset $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{z}_k), k = 1, \cdots, N\}$, which consists of inputs $\mathbf{x}_k$ and their hard segmentation $\mathbf{z}_k$, we would like to estimate parameters $\mathbf{w}$ such that the resulting inferred segmentations are accurate. Here, the accuracy is measured using the loss function $\Delta(\cdot, \cdot)$. Formally, let $\mathbf{y}_k(\mathbf{w})$ denote the soft segmentation obtained by minimizing the energy functional $E(\cdot, \mathbf{x}_k; \mathbf{w})$ for the k-th training sample, that is,

$$\mathbf{y}_k(\mathbf{w}) = \underset{\mathbf{y}}{\text{argmin}}\, \mathbf{w}^\top \psi(\mathbf{x}_k, \mathbf{y}) \,. \tag{10}$$

We would like to learn the parameters $\mathbf{w}$ such that the empirical risk is minimized. In other words, we would like to estimate the parameters $\mathbf{w}^*$ such that

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmin}} \frac{1}{N} \sum_k \Delta(\mathbf{z}_k, \mathbf{y}_k(\mathbf{w})) \,. \tag{11}$$

The above objective function is highly non-convex in $\mathbf{w}$, which makes it prone to bad local minimum solutions. To alleviate this deficiency, it can be shown that the following latent SVM formulation minimizes a regularized upper bound on the risk for a set of samples $\{(\mathbf{x}_k, \mathbf{z}_k), k = 1, \cdots, N\}$:

$$\min_{\mathbf{w} \geq \mathbf{0}} \lambda ||\mathbf{w}||^2 + \lambda' ||\mathbf{w} - \mathbf{w}_0||^2 + \frac{1}{N} \sum_k \xi_k \,, \tag{12}$$

$$\text{s.t.} \min_{\mathbf{y}_k, \Delta(\mathbf{z}_k, \mathbf{y}_k) = 0} \mathbf{w}^\top \psi(\mathbf{x}_k, \mathbf{y}_k) \leq \mathbf{w}^\top \psi(\mathbf{x}_k, \overline{\mathbf{y}}_k) - \Delta(\mathbf{z}_k, \overline{\mathbf{y}}_k) + \xi_k \,, \forall \overline{\mathbf{y}}_k, \forall k \,,$$

where the slack variable ξ_k represents the upper bound of the risk for the k-th training sample. Note that we have added two regularization terms for the parameters $\mathbf{w}$. The first term $||\mathbf{w}||^2$, weighted by hyperparameter λ, ensures that we do not overfit to the training samples. The second term $||\mathbf{w}-\mathbf{w}_0||^2$, weighted by hyperparameter λ', ensures that we do not obtain a solution that is very far away from our initial estimate $\mathbf{w}_0$. The reason for including this term is that our upper bound to the empirical risk may not be sufficiently tight. Thus, if we do not encourage our solution to lie close to the initial estimate, it may drift towards an inaccurate set of parameters. In section 4, we show the empirical effect of the hyperparameters λ and λ' on the accuracy of the parameters.

While the upper bound of the empirical risk derived above is not convex, it was shown to be a difference of two convex functions in [12]. This observation allows us to obtain a local minimum or saddle point solution using the CCCP algorithm [12,13], outlined in Algorithm 1, which iteratively improves the parameters starting with an initial estimate $\mathbf{w}_0$. It consists of two main steps at each iteration: (i) step 3, which involves estimating a compatible soft segmentation for each training sample—known as annotation consistent inference (ACI); and (ii) step 4, which involves updating the parameters by solving problem (13). In the following subsections, we provide efficient algorithms for both the steps.

Algorithm 1. The CCCP method for parameter estimation using latent SVM.

Input: Dataset $\mathcal{D}$, λ, λ', $\mathbf{w}_0$, ε

1: Set $t = 0$. Initialize $\mathbf{w}_t = \mathbf{w}_0$.
2: **repeat**
3: Compute $\mathbf{y}_k^* = \text{argmin}_{\mathbf{y}_k, \Delta(\mathbf{z}_k, \mathbf{y}_k)=0} \mathbf{w}_t^\top \psi(\mathbf{x}_k, \mathbf{y}_k), \forall k$.
4: Update the parameters by solving the following problem

$$\mathbf{w}_{t+1} = \underset{\mathbf{w} \geq \mathbf{0}}{\text{argmin}} \lambda||\mathbf{w}||^2 + \lambda'||\mathbf{w} - \mathbf{w}_0||^2 + \frac{1}{N}\sum_k \xi_k \,, \tag{13}$$
$$\text{s.t.} \mathbf{w}^\top \psi(\mathbf{x}_k, \mathbf{y}_k^*) \leq \mathbf{w}^\top \psi(\mathbf{x}_k, \overline{\mathbf{y}}_k) - \Delta(\mathbf{z}_k, \overline{\mathbf{y}}_k) + \xi_k, \forall \overline{\mathbf{y}}_k, \forall k \,,$$

5: $t = t + 1$
6: **until** The objective function of problem (12) does not decrease below tolerance ε.

3.1 Annotation Consistent Inference

Given an input $\mathbf{x}$ and its hard segmentation $\mathbf{z}$, ACI requires us to find the soft segmentation $\mathbf{y}$ with the minimum energy, under the constraint that it should be compatible with $\mathbf{z}$ (see step 3 of Algorithm 1). We denote the ground truth label of a voxel i by s_i, that is, $s_i = \text{argmax}_s \mathbf{z}(i, s)$, and the set of all voxels by $\mathcal{V}$. Using our notation, ACI can be formally specified as

$$\min_{\mathbf{y} \in \mathcal{C}(\mathcal{V})} \mathbf{y}^\top L(\mathbf{x}; \mathbf{w})\mathbf{y} + E^{\text{prior}}(\mathbf{y}, \mathbf{x}; \mathbf{w}) \,. \tag{14}$$

Here, $\mathcal{C}(\mathcal{V})$ is the set of all compatible probabilistic segmentations, that is,

$$\mathbf{y}(i,s) \geq 0, \forall i \in \mathcal{V}, \forall s \in \mathcal{S}\,, \tag{15}$$

$$\sum_{s \in \mathcal{S}} \mathbf{y}(i,s) = 1, \forall i \in \mathcal{V}\,, \tag{16}$$

$$\mathbf{y}(i,s_i) \geq \mathbf{y}(i,s), \forall i \in \mathcal{V}, \forall s \in \mathcal{S}\,. \tag{17}$$

Constraints (15) and (16) ensure that $\mathbf{y}$ is a valid probabilistic segmentation. The last set of constraints (17) ensure that $\mathbf{y}$ is compatible with $\mathbf{z}$. Note that in the absence of constraints (17), the above problem can be solved efficiently using the RW algorithm. However, since the ACI problem requires the additional set of compatibility constraints, we need to develop a novel efficient algorithm to solve the above convex optimization problem. To this end, we exploit the powerful dual decomposition framework [2,7]. Briefly, we divide the above problem into a set of smaller subproblems defined using overlapping subsets of variables. Each subproblem can be solved efficiently using a standard convex optimization package. In order to obtain the globally optimal solution of the original subproblem, we pass *messages* between subproblems until they agree on the value of all the shared variables. For details on the ACI algorithm, please refer to the supplementary materials of this paper.

3.2 Parameter Update

Having generated a compatible soft segmentation, the parameters can now be efficiently updated by solving problem (13) for a fixed set of soft segmentations $\mathbf{y}_k^*$. This problem can be solved efficiently using the popular cutting plane method (for details on this algorithm, please refer to [6]). Briefly, the method starts by specifying no constraints for any of the training samples. At each iteration, it finds the most violated constraint for each sample, and updates the parameters until the increase in the objective function is less than a small epsilon.

In this work, due to the fact that our loss function is not concave, we approximate the most violated constraint as the predicted segmentation, that is,

$$\overline{\mathbf{y}} = \operatorname*{argmin}_{\mathbf{y}} \mathbf{w}^\top \psi(\mathbf{x}, \mathbf{y})\,. \tag{18}$$

The above problem is solved efficiently using the RW algorithm.

4 Experiments

Dataset. The dataset consists of 30 MRI volumes of the thigh region of dimensions $224 \times 224 \times 100$. The various segments correspond to 4 different muscle groups together with the background class. We randomly split the dataset into 80% for training and 20% for testing. In order to reduce the training time for both our method and the baselines, we divide each volume into 100/2 volumes of dimension $224 \times 224 \times 2$.

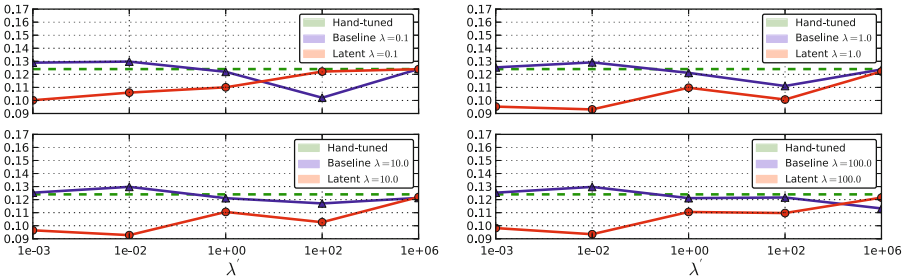

Fig. 1. Estimated risk $\Delta(\mathbf{y}_k^\star, \mathbf{y}_k(\mathbf{w}))$ for three different methods

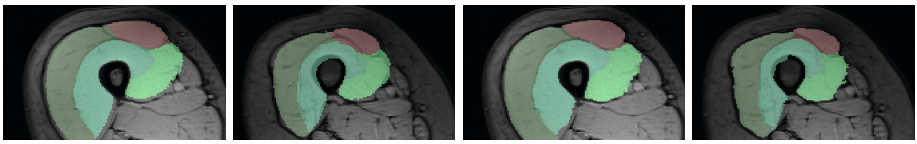

Fig. 2. Method comparison: (columns 1 & 2) segmentations using $\mathbf{w_0}$; (columns 3 & 4) segmentations using learned $\mathbf{w}$ using latent structured SVM. The latter are closer to expert segmentation.

Laplacians and Prior Terms. We use 4 different Laplacians (generated with different weitghing functions). Furthermore, we use two shape priors based on [1] and one appearance prior based on [4]. This results in a total of 7 parameters to be estimated.

Methods. The main hypothesis of our work is that it is important to represent the unknown optimal soft segmentation using latent variables. Thus we compare our method with a baseline structured SVM that replaces the latent variables with the given hard segmentations. In other words, our baseline estimates the parameters by solving problem (13), where the imputed soft segmentations $\mathbf{y}_k^*$ are replaced by the hard segmentations $\mathbf{z}_k$. During our experiments, we found that replacing the hard segmentation with a pseudo soft segmentation based on the distance transform systematically decreased the loss of the output. Thus the method refered to as "Baseline" uses a structured SVM with distance-tranform "softened" segmentations.

Results. Fig. 1 shows the test loss for three different methods: (i) the initial hand-tuned parameters $\mathbf{w}_0$; (ii) the baseline structured SVM with distance transforms; and (iii) our proposed approach using latent SVM. As can be seen from Fig. 1, latent SVM provides significantly better results than the baselines—even when using the distance transform. For the 4 x 5 hyperparameter settings that we report (that is, four different values of λ and 5 different values of λ'), latent SVM is significantly better than SVM in 15 cases, and significantly worse in only 2 cases. Note that latent SVM provides the best results for very small values of λ', which indicates that the upper bound on the empirical risk in tight. As expected, for sufficiently large values of λ', all the methods provide similar results.

For the best settings of the corresponding hyperparameters, the percentage of incorrectly labeled voxels as follows: (i) for $\mathbf{w}_0$, 13.5%; (ii) for structured SVM, 10.0%; and (iii) for latent SVM, 9.2%. Fig. 2 shows some example segmentations for the various methods.

5 Discussion

We proposed a novel discriminative learning framework to estimate the parameters for the probabilistic RW segmentation algorithm. We represented the optimal soft segmentation that is compatible with the hard segmentation of each training sample as a latent variable. This allowed us to formulate the problem of parameter estimation using latent SVM, which upper bounds the empirical risk of prediction with a difference of convex optimization program. Using a challenging clinical dataset of MRI volumes, we demonstrated the efficacy of our approach over the baseline method that replaces the latent variables with the given hard segmentations. The latent SVM framework can be used to estimate parameters with partial hard segmentations. Such an approach would allow us to scale the size of the training dataset by orders of magnitude.

References

1. Baudin, P.-Y., Azzabou, N., Carlier, P.G., Paragios, N.: Prior knowledge, random walks and human skeletal muscle segmentation. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 569–576. Springer, Heidelberg (2012)
2. Bertsekas, D.: Nonlinear Programming. Athena Scientific (1999)
3. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
4. Grady, L.: Multilabel random walker image segmentation using prior models. In: CVPR (2005)
5. Grady, L.: Random walks for image segmentation. PAMI (2006)
6. Joachims, T., Finley, T., Yu, C.N.: Cutting-plane training of structural SVMs. Machine Learning (2009)
7. Komodakis, N., Paragios, N., Tziritas, G.: MRF optimization via dual decomposition: Message-passing revisited. In: ICCV (2007)
8. Smola, A., Vishwanathan, S., Hoffman, T.: Kernel methods for missing variables. In: AISTATS (2005)
9. Szummer, M., Kohli, P., Hoiem, D.: Learning cRFs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
10. Taskar, B., Guestrin, C., Koller, D.: Max-margin Markov networks. In: NIPS (2003)
11. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML (2004)
12. Yu, C.N., Joachims, T.: Learning structural SVMs with latent variables. In: ICML (2009)
13. Yuille, A., Rangarajan, A.: The concave-convex procedure (CCCP). Neural Computation (2003)