Self-Tuning Semantic Image Segmentation

Sergey Milyaev[1,2], Olga Barinova[2]

[1] Voronezh State University
sergey.milyaev@gmail.com
[2] Lomonosov Moscow State University
obarinova@graphics.cs.msu.su

Abstract. In this paper we present a method for finding optimal parameters of graph Laplacian-based semantic segmentation. This method is fully unsupervised and provides parameters individually for each image. In the experiments on Graz dataset the accuracy of segmentation obtained with the parameters provided by our method is very close to the accuracy of segmentation obtained with the parameters chosen on the test set.

1 Introduction

Methods based on graph Laplacian (L2-norm regularization) have shown state-of-the-art results for interactive image segmentation [1] and image matting [2]. In [1] Grady suggested explanation of using Laplacians for interactive segmentation in terms of random walks. In [3] the use of graph Laplacian for interactive image segmentation was explained in terms of transductive inference. The parameters of graph Laplacian are usually chosen by validation on hold-out dataset. However, the optimal values of parameters can vary significantly from one image to another, therefore choosing the parameters individually for each image is desirable.

In this paper we consider the task of finding optimal parameters of graph Laplacian for semantic image segmentation. We propose a new method that tunes the parameters individually for each test image without using any ground truth segmentation. The idea of our method is based on the properties of graph Laplacian to approximate the Laplace-Beltrami operator studied in [4]. Proposed self-tuning method is computationally efficient and achieves performance comparable to choosing the parameters on the test set.

The remainder of the paper if organized as follows. In section 2 we describe the image segmentation framework used in this paper. In section 3 we present our method for unsupervised learning of graph Laplacian parameters. In section 4 we present the experimental evaluation of the proposed method.

2 Semantic segmentation framework

Let us denote $W : W_{ij} = \exp\left(-d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)^2\Big)$ - a weight matrix with Gaussian kernel. Let $g_i = \sum_j w_{ij}$ stand for a sum of W along the i-th row. Let D be

a diagonal matrix with values g_i on diagonal. Graph Laplacian is defined as a matrix $L = W - D$.

The methods for image segmentation and matting solve the following energy function with respect to vector $\mathbf{f} = (f_1, ..., f_N)$:

$$E(\mathbf{f}) = \sum_i c_i \left(f_i - y_i\right)^2 + \sum_{i,j} w_{ij} \left(f_i - f_j\right)^2 . \tag{1}$$

In the matrix form (1) takes the following form:

$$E(\mathbf{f}) = \left(\mathbf{f} - \mathbf{y}\right)^T C \left(\mathbf{f} - \mathbf{y}\right) + \mathbf{f}^T L \mathbf{f} , \tag{2}$$

where C denotes a square diagonal matrix with c_i on diagonal and $\mathbf{y}$ denotes an N-dimensional vector of initial likelihood scores y_i. This optimization problem reduces to solving a sparse linear system:

$$(L + C)\mathbf{f} = C\mathbf{y}. \tag{3}$$

The object/background segmentation algorithm then consists in: 1) computing graph Laplacian matrix L; 2) solving the sparse linear system (3); 3) thresholding the output. We assume that initial estimates y_i and confidences c_i are provided by local models (e.g. appearance model of a specific category).

This framework can be extended to a multi-class segmentation. Let K denote the number of labels corresponding to object categories. If we solve (3) for each label l vs all other labels $1, \cdots, l-1, l+1, \cdots, K$ and obtain the values $y_i^{(l)}$ for all image pixels; at the end, an i-th image pixel is assigned to the label l_{max}, where $l_{max} = arg \max_{l=1,\cdots,K} y_i^{(l)}$.

3 Self-tuning method

Suppose that the distance function d is represented as a weighted sum of metrics $d_i : \mathbb{R} \times \mathbb{R} \to R^+; i = 1, ..., K$:

$$d(\mathbf{x}_i, \mathbf{x}_j)^2 = \frac{1}{\epsilon} \sum_{k=1}^{K} \alpha_k d_k(\mathbf{x}_i, \mathbf{x}_j)^2 , \tag{4}$$

with fixed $\alpha_1 = 1$. Therefore the parameters of graph Laplacian $\alpha_i, i = 2, ..., l$ are the weights of features $\mathbf{x}^k, i = 2, ..., l$ and the kernel bandwidth ϵ. Below we show that optimal value of ϵ is determined by the values of $\alpha_i, i = 2, ..., l$.

Choosing the kernel bandwidth ϵ with fixed α. We start by fixing the parameters $\alpha_i, i = 2, ..., l$. As shown in [5], if we assume that L provides a good approximation of Laplace-Belrami operator then the following condition holds:

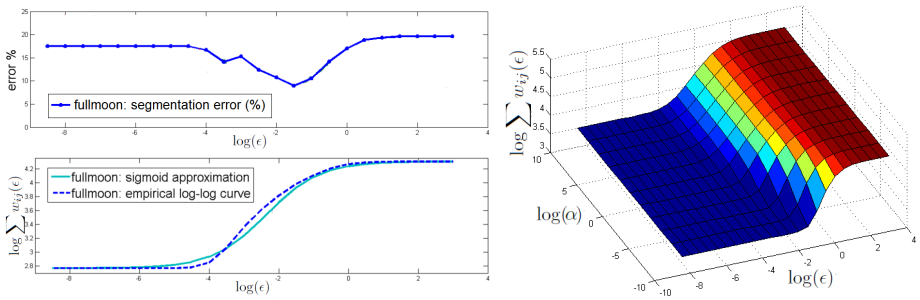
$$\log \sum_{i,j} w_{ij}(\epsilon) \approx m/2 \log(\epsilon) + \log \left(\frac{N^2 (2\pi)^{m/2}}{vol(M)} \right) , \tag{5}$$

where m is a dimensionality of corresponding manifold M and w_{ij} are the elements of the weight matrix W.

Consider the *logarithmic plot* of $\log \sum_{i,j} w_{ij}$ with respect to $\log \epsilon$. Figure (3) shows the plot of $\log \sum_{ij} w_{ij}$ with respect to $\log \epsilon$ and $\log \alpha$ for one image from GrabCut dataset. According to (5) if the approximation is good then the slope of this plot ϵ should be about the half dimensionality of corresponding manifold.

In the limit $\epsilon \to \infty$, $w_{ij} \to 1$, so $\sum_{ij} w_{ij} \to N^2$. On the other hand, as $\epsilon \to 0$, $w_{ij} \to \delta_{ij}$, so $\sum_{ij} w_{ij} \to N$. These two limiting values set two asymptotes of the plot and assert that logarithmic plot cannot be linear for all values of ϵ.

Therefore in order to get better approximation of Laplace-Beltrami operator with $\alpha_1, ..., \alpha_K$ fixed we have to choose the value of ϵ from the linear region of logarithmic plot. We use the point of maximum derivative as the point of maximum linearity.



(a) (b)

Fig. 1. (a) - *Top:* segmentation errors for the "fullmoon" image from GrabCut database with respect to $\log \epsilon$ (α is fixed). *Bottom:* Dashed line - logarithmic plot for the "fullmoon" image with respect to $\log \epsilon$ (α is fixed). The optimal value of ϵ is chosen in the point of maximum derivative of the logarithmic plot; Solid line - sigmoid fit of the logarithmic plot. (b) - The plot of $\log \sum_{ij} w_{ij}$ with respect to $\log \epsilon$ and $\log \alpha$. Th plot shown in shown in (3, bottom) corresponds to the 2-d slice of this 3-d plot for fixed α. Note that the slope of linear region are not constant for all values of α. We seek for α such that the slope in the linear region equals 0.5.

Unsupervised learning of $\alpha_1, ..., \alpha_K$ *and* ϵ. As follows from (5) the slope of the logarithmic curve near optimal value of ϵ has to be close to $m/2$, where m is the dimensionality of manifold M. In our case $m = 1$, therefore the slope of the logarithmic plot has to be 0.5. If the plot has different slope in the linear region, this indicates that the second term in (5) is large.

In order to find optimal values of $\alpha_2, ..., \alpha_K$ we solve the following optimization problem:

$$(\alpha_2^{(opt)}, ..., \alpha_K^{(opt)}) = arg \min_{\alpha_2,...,\alpha_K} \|S(\alpha_2, ..., \alpha_K) - 0.5\|, \tag{6}$$

where $S(\alpha_2, ..., \alpha_K)$ is the slope of the logarithmic plot in the point of maximum derivative.

$S(\alpha_2, ..., \alpha_K)$ can be estimated numerically. We can compute $\log \sum_{ij} w_{ij}$ for different values of ϵ and estimate the slope of this function in the point of maximum derivative. Therefore the optimization problem (6) can be solved using standard optimization methods, e.g. Nelder-Mead simplex method.

The unsupervised learning method for graph Laplacian therefore has two steps:

- Find $\alpha_2, ..., \alpha_K$ by solving optimization problem (6)
- Find ϵ with $\alpha_2, ..., \alpha_K$ as the point of maximum derivative of the logarithmic plot.

Implementation details For the experiments in this work we use the distance function from [3]:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{\|r_i - r_j\|^2}{\sigma_r^2} + \frac{\|x_i - x_j\|^2}{\sigma_g^2}, \tag{7}$$

where r encodes mean RGB color in the superpixel, x encodes coordinates of the center of the superpixel, $\sigma_r > 0$ and $\sigma_g > 0$ are the parameters of the method. The meaning $\sigma_r > 0$ and $\sigma_g > 0$ is the scale of chromatic neighbourhoods and the scale of geometric neighbourhoods respectively.

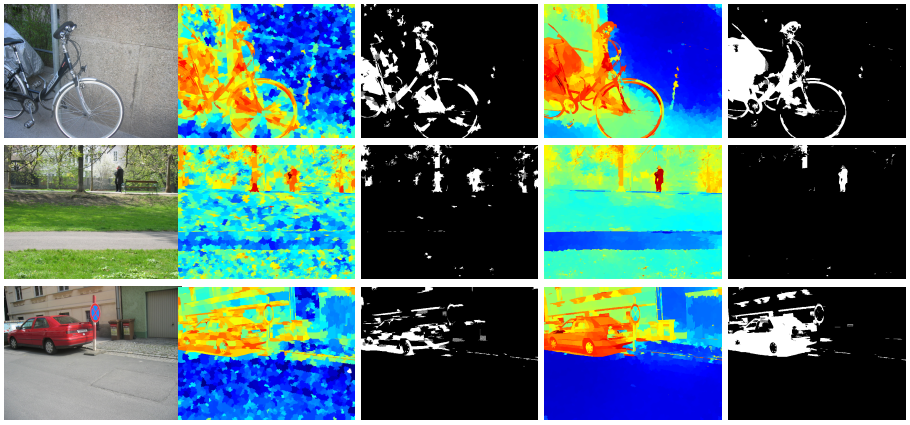
This distance function (7) can be rewritten in the form of (4) as follows:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\epsilon}\left(\|r_i - r_j\|^2 + \alpha \|x_i - x_j\|^2\right), \tag{8}$$

where $\epsilon = 0.5\sigma_r^2$ and $\alpha = \sigma_r^2/\sigma_g^2$. Therefore, the distance function has two parameters ϵ and α.

In the second step of the learning method (3) we use the sigmoid fit of the logarithmic plot. The shape of logarithmic plot can be approximated with a sigmoid function:$T(x) = \frac{A}{B+\exp(Cx+D)} + E$. Since the asymptotes of the sigmoid are set by (5) and the slope in the linear region of the sigmoid should be 0.5 the sigmoid has only one free parameter that controls the shift of the sigmoid along horizontal axis. Figure (3) illustrates the choice of ϵ according to sigmoid approximation.

In most cases the slope of the logarithmic plot $S(\alpha)$ is monotonic function of α. Monotonicity of $S(\alpha)$ allows using simple bin-search for optimization problem (6).



(a) *input image* (b) *local model* (c) *thresholded (b)* (d) *Laplacian* (e) *thresholded (d)*

Fig. 2. Results of SVM and graph Laplacian method for images from Graz dataset. (a) - input images of "bike", "person" and "cars" classes; (b) - real-valued output from local SVM model, color ranges from blue to red and encodes the real-valued output; (c) - results of thresholding the SVM outputs; (d) - real-valued output of graph Laplacian using SVM as a local model with the parameters learnt by our method, color ranges from blue to red and encodes the real-valued output; (e) - thresholded output of our method. Note how graph Laplacian refines the output from SVM. It doesn't oversmooth the result and preserves fine details like the wheel of the bike and the small figure of the person.

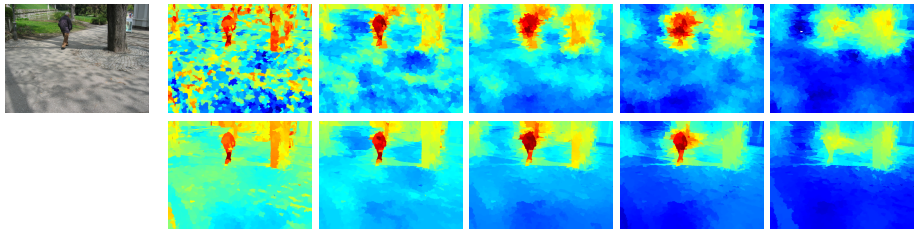
4 Experiments

In all experiments graph Laplacian operated with superpixels produced by image over-segmentation methods. Each superpixel was linked with a fixed number of it's nearest neighbours, and the distances to other superpixels were assumed infinite. For all experiments we used confidences that are a linear function of the outputs of local appearance models $c_i = 0.5(1 - |p_i - 0.5|)$.

Graz dataset [1] contains 1096 images of three classes: "person", "bike" and "car". In our experiments we solved a separate binary segmentation problem for each category. To measure the quality of segmentation we used a standard metric - percent of incorrectly classified pixels in the image.

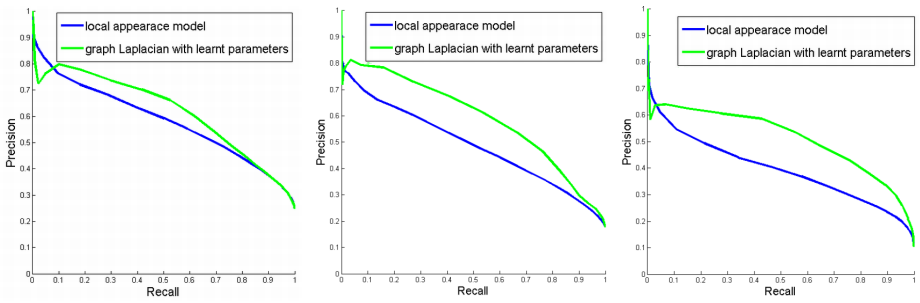
In our experiments we used an open-source VlBlocks toolbox [2], which implements the method described in [6]. We chose it for comparison for the following reasons. First, it allows using different local appearance models. The method has a parameter N meaning number of neighbouring superpixels which features are used for classification of each particular superpixel. So we report performance metrics for different values of N to illustrate the performance of proposed graph Laplacian framework applied to different local models. Second, the toolbox in-

[1] available at `http://www.emt.tugraz.at`
[2] code available at `http://vlblocks.org/index.html`



(a) *input image* (b) *N=0* (c) *N=1* (d) *N=2* (e) *N=3* (f) *N=4*

Fig. 3. Results of using different local models. The first row shows real-valued output of local appearance models. The color ranges from blue to red and encodes the real-valued output from the segmentation framework. The second row shows results of our method. Parameter N sets the size of superpixel neighborhood in the local model. The effect of using graph Laplacian is better visible for smaller N.



(a) *"bike" class* (b) *"person" class* (c) *"car" class*

Fig. 4. Precision-recall curves for "bike", "person" and "car" classes of Graz dataset. *Blue curves* - local appearance model (N=0); *Green curves* - graph Laplacian with learnt parameters.

cludes implementation of discrete CRF with graph-cut inference, which we use for comparison. Note, this CRF model uses similar types of features (color and spatial coordinates of superpixels) to those used in our graph Laplacian.

In our experiments on GrabCut dataset we used the same over-segmentation and the same local appearance model based on SVM as [6]. To obtain initial estimates y_i for graph Laplacian framework we scaled SVM outputs to $[0, 1]$ interval for each image.

In the first experiment the parameters ϵ and α were validated on the GrabCut dataset. In the second experiment we validated the parameters on the test set. In the third experiment we used our unsupervised learning method for choosing the parameters individually for each image. We also compared with Vlblocks implementation of CRF with graph-cut inference. The strategy for choosing internal parameters of CRF was the same as in [6].

Table 1 contains results of the comparison. Our unsupervised learning gives results comparable to upper bound on performance of graph Laplacian with

	N=0			N=1			N=2			N=3			N=4		
	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers
SVM	41.9	56.5	49.4	59.6	66.9	63.6	68.0	69.2	66.6	69.4	70.7	65.2	66.5	71.9	63.6
GraphCut	43.0	57.7	49.3	60.2	67.1	63.9	70.1	70.2	66.9	70.7	71.0	65.4	68.8	72.2	64.2
Ours (valid.GrabCut)	50.0	60.1	56.0	65.5	68.7	68.5	71.6	70.8	70.8	72.2	72.0	69.5	70.0	**73.2**	67.3
Ours (valid.testset)	**56.6**	**63.3**	**59.1**	66.3	68.4	68.8	71.9	70.4	70.4	72.6	71.2	69.4	70.8	72.2	68.0
Ours (learnt)	54.2	60.9	58.5	65.1	66.8	**69.4**	**72.0**	69.5	**71.3**	**73.3**	70.3	**70.2**	**71.4**	71.5	**68.9**

Table 1. Performance on Graz dataset at equal precision and recall rates for "cars", "bike" and "person" classes. First row: local appearance model (from VlBlocks toolbox). Second row: result of applying discrete CRF with graph cut inference (from VlBlocks toolbox). Third row: graph Laplacian with parameters validated on GrabCut dataset. Fourth row: graph Laplacian with parameters validated on the test set. Fifth row: graph Laplacian with parameters learnt individually for each image. For each appearance model used in our experiments (we varied the number of neighboring regions as in [6]) the best result is shown in **bold font**. Underlined are the best overall results.

fixed parameters from the second experiment. The value of performance gain compared to local appearance model differs for different values of parameter N. The smaller N is the smaller neighborhood is considered by low-level model, and the more significant is the gain in performance attained by both CRF and graph Laplacian.

The gain in performance of graph Laplacian is almost uniformly higher than the performance gain obtained by discrete CRF. Figure 2 shows results provided by local appearance model (SVM) and corresponding results of using graph Laplacian with learnt parameters. Figure 3 shows how the results vary for different local models.

The running time is the following: learning phase takes about 0.2 seconds on average, solving of linear system 3 takes about 0.02 seconds on average.

5 Conclusion

We presented a method for tuning internal parameters of graph Laplacian in a fully unsupervised manner individually for each test image. Proposed method has a low computational cost and shows better performance compared to discrete CRF with graph-cut inference. In the future work we plan to use more complex distance functions and investigate the case then distance function has more parameters.

References

1. Grady, L.: Random walks for image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **28**(11) (2006) 1768–1783
2. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. IEEE Trans. on Pattern Analysis and Machine Intelligence (2008)
3. Duchenne, O., Audibert, J.Y., Keriven, R., Ponce, J., Segonne, F.: Segmentation by transduction. In: CVPR. (2008)

4. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computataion **15** (2003) 1373–1396
5. Coifman, R.R., Shkolnisky, Y., Sigworth, F.J., Singer, A.: Graph laplacian tomography from unknown random projections. IEEE Trans. on Image Processing
6. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: ICCV. (2009)