Parameter Learning for CRF-based Tissue Segmentation of Brain Tumors

Raphael Meier[1], Venetia Karamitsou[1], Simon Habegger[2], Roland Wiest[2], and Mauricio Reyes[1]

[1] Institute for Surgical Technologies and Biomechanics, University of Bern
[2] Inselspital, Bern University Hospital, Switzerland
raphael.meier@istb.unibe.ch

Abstract. In this work, we investigate the potential of a recently proposed parameter learning algorithm for Conditional Random Fields (CRFs). Parameters of a pairwise CRF are estimated via a stochastic subgradient descent of a max-margin learning problem. We compared the performance of our brain tumor segmentation method using parameter learning to a version using hand-tuned parameters. Preliminary results on a subset of the BRATS2015 training set show that parameter learning leads to comparable or even improved performance. Future work will include training on the complete data set and the use of more elaborate loss functions suitable for brain tumor segmentation.

1 Introduction

Brain tumor segmentation yields information about the volume of a tumor and its position relative to neighboring possibly eloquent brain areas. Alternatively, such information can only be obtained via time-consuming and subjective manual segmentation. Consequently, fully-automatic segmentation methods applicable in a wide range of domains such as neurooncology, neurosurgery and radiotherapy are in high demand.

The development of new brain tumor segmentation methods has been fostered through the MICCAI Brain Tumor Segmentation (BRATS) Challenge [4], which was held for the first time during MICCAI 2012. Several previously published segmentation methods rely on the use of structured prediction including approaches such as Markov or Conditional Random Fields (CRFs) (e.g. [7, 3]). However, parameters for those models are often hand-tuned rather than estimated from training data. Recently, an efficient method for parameter learning in CRFs applicable to volumetric imaging data was proposed [2]. In this paper, we investigate a modification of our previous segmentation method [3] employing the learning algorithm of [2].

2 Methods

Our current segmentation method (proposed in [3]) encompasses a preprocessing, a feature extraction step followed by a voxel-wise classification and a spatial regularization. The features try to capture visual cues of appearance and image context relevant for discriminating the different tissue classes. Classification is performed by a decision forest. Spatial regularization is formulated as an energy-minimization problem of a CRF. In the remainder of this paper, we present a modification of the spatial regularization used so far.

Structural MRI. Our approach relies on four different MRI sequences, namely T_1-, T_1-post contrast-, T_2-, $FLAIR$-weighted images. We assume that these images are co-registered and organized as a vector image, where every voxel contains the four different MR intensity values. We refer to this image as $X = \left\{\mathbf{x}^{(i)}\right\}_{i \in V}$, where voxel i is represented by a feature vector $\mathbf{x}^{(i)} \in \mathbb{R}^4$ and V denotes the set of all voxels in X. The corresponding tissue label map of X is denoted by $Y = \left\{y^{(i)}\right\}_{i \in V}$ with $y^{(i)}$ being a scalar tissue label (e.g. 1=necrosis, 2=edema, etc.). We consider seven possible tissue classes ($|\mathcal{L}|$=7): three unaffected (gray matter, white matter, csf) and four tumor tissues (necrosis, edema, enhancing and non-enhancing tumor). All possible labelings are contained in $\mathcal{Y}$.

Conditional Random Field. A CRF models a parametrized conditional probability $p\left(Y|X,\mathbf{w}\right) = \frac{1}{Z(X,\mathbf{w})} \exp\left(-E(X,Y,\mathbf{w})\right)$ where $Z(X,w)$ is the partition function. The energy $E(X,Y,\mathbf{w})$ depends linearly on the unknown parameters $\mathbf{w}$. In general, given the parameter vector $\mathbf{w}$, a CRF can predict the labeling Y of a given input image X by minimizing the energy, i.e. $Y^\star = \arg\min_{Y \in \mathcal{Y}} E(X,Y,\mathbf{w})$.

Energy Function. We employ an energy function associated with a pairwise CRF: $E(X,Y,\mathbf{w}) = \sum_{i \in V} D_i(\mathbf{x}^{(i)}, y^{(i)}) + \sum_{(i,j) \in E} B_{i,j}(\mathbf{x}^{(i)}, y^{(i)}, \mathbf{x}^{(j)}, y^{(j)})$. The unary potentials D_i and pairwise potentials $B_{i,j}$ are expressible as an inner product between the parameter vector $\mathbf{w}$ and a feature map ψ_i or $\psi_{i,j}$, respectively [2]. For a given feature vector $\mathbf{x}^{(i)}$, we can define the feature map $\psi_i = \left[I(y^{(i)}=1)(-\log(p(y^{(i)}=1|\mathbf{x}^{(i)}))), \cdots, I(y^{(i)}=7)(-\log(p(y^{(i)}=7|\mathbf{x}^{(i)})))\right]^T$ by using the indicator function I (returns a value of 1 if the argument is true). The posterior probability $p(y^{(i)}|\mathbf{x}^{(i)})$ is output by the decision forest classifier. Consequently, the cost of assigning label y to voxel i is smaller the more confident the prediction of the decision forest is. The pairwise feature map is given by $\psi_{i,j} = \left[I(y^{(i)}=a, y^{(i)}=b)(1 - I(y^{(i)}=y^{(j)})) \exp\left(-\left\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right\|_\infty\right)\right]_{(a,b) \in \mathcal{L}^2}$ which is defined for all possible label pairs in $\mathcal{L}$. The term $1 - I(y^{(i)} = y^{(j)})$ establishes a Potts-like model. The exponential term penalizes large intensity discontinuities between neighboring voxels. Potentials can now be expressed as an inner product between parameter vector and feature map, i.e. $\langle \mathbf{w}, \psi \rangle$. Furthermore, let $\Psi^D = \sum_{i \in V} \psi_i$ and $\Psi^B = \sum_{(i,j) \in E} \psi_{i,j}$. Given the parameter vector $\mathbf{w} = \left[(\mathbf{w}^D)^T, (\mathbf{w}^B)^T\right]^T$, the energy function can then be rewritten as $E(X,Y,\mathbf{w}) = \left\langle \mathbf{w}^D, \Psi^D \right\rangle + \left\langle \mathbf{w}^B, \Psi^B \right\rangle$.

Parameter Learning. For estimating the parameter vector $\mathbf{w}$, we use the recently proposed method by Lucchi et al. [2] which builds on the max-margin formulation for parameter learning [6]. Essentially, learning is posed as a quadratic program with soft margin constraints. The objective function is minimized via stochastic subgradient descent in which iteratively a training example $\left(X^{(n)}, Y^{(n)}\right)$ is chosen, the subgradient with respect to this example computed and the weight vector updated accordingly (see algorithm 1). The objective function for $\left(X^{(n)}, Y^{(n)}\right)$ is defined as $f(\mathbf{w}, n) = l\left(Y^{(n)}, Y^{\star}, \mathbf{w}\right) + \frac{1}{2C}\left\|\mathbf{w}\right\|^2$ with l being the hinge loss[3]. The task-specific loss is defined as $\Delta\left(Y^{(n)}, Y\right) = \sum_{i \in V} I\left(y^{(i)} \neq y^{(n),(i)}\right)$ and measures the dissimilarity between a labeling Y and its ground truth $Y^{(n)}$. In contrast to [5], the method of Lucchi et al. aims at an increased reliability in the computation of the subgradient by the use of working sets of constraints $\mathcal{A}^n$. For every iteration, loss-augmented inference is performed to obtain a current estimate of the labeling $Y^{\star} = \arg\min_{Y \in \mathcal{Y}}\left(E(X, Y, \mathbf{w}) - \Delta\left(Y^{(n)}, Y\right)\right)$ (step 4). The set $\mathcal{A}^{n'}$ contains all labelings (constraints) Y which are violated (i.e. $l(Y, Y^{(n)}, \mathbf{w}) > 0$) (step 7). The subgradient is then computed as an average subgradient over all violated constraints (step 8).

Algorithm 1 Subgradient Method with Working Sets [2]

1: Training data $\mathcal{S} = \left\{(X^{(i)}, Y^{(i)}) : i = 1, ..., m\right\}, \beta := 1, \mathbf{w}^{(1)} := \mathbf{0}, t := 1$
2: **while** $(t < T)$ **do**
3: Pick randomly an example $(X^{(n)}, Y^{(n)})$ from $\mathcal{S}$
4: $Y^{\star} = \arg\min_{Y \in \mathcal{Y}}(E(X, Y, \mathbf{w}) - \Delta(Y^{(n)}, Y))$
5: $\mathcal{A}^n := \mathcal{A}^n \cup \{Y^{\star}\}$
6: $\mathcal{A}^{n'} := \left\{Y \in \mathcal{A}^n : l(Y, Y^{(n)}, \mathbf{w}^{(t)}) > 0\right\}$
7: $\eta^{(t)} := \frac{\beta}{t}$
8: $\mathbf{g}^{(t)} := \frac{1}{\mathcal{A}^{n'}} \sum_{Y \in A^{n'}} \left(\Psi^D(Y^{(n)}) + \Psi^B(Y^{(n)}) - \left(\Psi^D(Y) + \Psi^B(Y)\right) + \frac{1}{C}\mathbf{w}\right)$
9: $\mathbf{w}^{(t+1)} := \mathcal{P}\left[\mathbf{w}^{(t)} - \eta^{(t)}\mathbf{g}^{(t)}\right]$
10: $t := t + 1$
11: **end while**

For performing loss-augmented inference, we employed the Fast-PD algorithm proposed by Komodakis et al. [1]. Fast-PD requires $B_{i,j}(\cdot, \cdot) \geq 0$.[4] The update of the weights (step 9) can potentially violate this constraint. Thus, we apply a projection $\mathcal{P}$ to ensure the compatibility of the weights $\mathbf{w}$ with Fast-PD.

3 Results

We evaluated our method via a 5-fold cross-validation on a subset of the BRATS2015 training data, encompassing 20 high-grade glioma cases (part of the former

[3] $l(Y^{(n)}, Y^{\star}, \mathbf{w}) = [E(X^{(n)}, Y^{(n)}, \mathbf{w}) + \Delta(Y^{(n)}, Y) - E(X^{(n)}, Y^{\star}, \mathbf{w})]_+$
[4] Fast-PD requires $B_{i,j}$ to define a semi-metric.

BRATS2013 training set). The performance of the presented method was compared against our previous approach using hand-tuned CRF parameters (baseline). Quantitative results are presented in table 1.

Region	Dice coefficient	Absolute volume error $[mm^3]$
Complete tumor (CRF+Learning)	$(0.887, 0.35)/(0.885, 0.35)$	$(10276, 41871)/(11078, 41257)$
Complete tumor (CRF Baseline)	$(0.888, 0.353)/(0.886, 0.353)$	$(9029, 42199)/(9029, 42001)$
Tumor core (CRF+Learning)	$(0.784, 0.912)/(0.793, 0.538)$	$(6504, 29505)/(6472, 29505)$
Tumor core (CRF Baseline)	$(0.789, 0.915)/(0.79, 0.58)$	$(6057, 32954)(6017, 32954)$
Enhancing tumor (CRF+Learning)	$(\mathbf{0.811}, 0.918)/(\mathbf{0.812}, 0.827)$	$(2784, 29875)/(2825, 29875)$
Enhancing tumor (CRF Baseline)	$(0.767, 0.942)/(0.768, 0.852)$	$(2485, 36986)/(2041, 36986)$

Table 1: Results of evaluation on subset of BRATS2015 training set. Performance measures are given as (median, range=max-min). Left tuple: Results for all 20 cases. Right tuple: Results after removal of outlier "brats_2013_pat0012_1 ".

4 Discussion and Future Work

The preliminary results indicate that learning CRF parameters from data instead of hand-tuning them can lead to comparable or even improved performance. Future work for our final submission will include training on the complete BRATS2015 training set and the investigation of more elaborate task-specific loss functions.

Acknowledgments. This project has received funding from the European Unions Seventh Framework Programme for research, technological development and demonstration under grant agreement N°600841.

References

1. Komodakis, N., Tziritas, G.: Approximate Labeling via Graph Cuts based on Linear Programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(8), 2007.
2. Lucchi, A., Marquez-Neila, P., Becker, C., Li, Y., Smith, K., Knott, G., Fua, P.: Learning Structured Models for Segmentation of 2D and 3D Imagery. IEEE Transactions on Medical Imaging (March), 2014.
3. Meier, R., Bauer, S., Slotboom, J., Wiest, R., Reyes, M.: Appearance- and Context-sensitive Features for Brain Tumor Segmentation. MICCAI BRATS Challenge Proceedings, 2014.
4. Menze, B.H., Jakab, A., et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). TMI 2014.
5. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: (Online) Subgradient Methods for Structured Prediction. Artificial Intelligence and Statistics, 2007.
6. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support Vector Machine Learning for Interdependent and Structured Output Spaces. ICML 2004.
7. Zhao, L., Wu, W., Corso, J.J.: Semi-Automatic Brain Tumor Segmentation by constrained MRFs using Structural Trajectories. MICCAI 2013.