HAL
open science

Efficient Learning for Discriminative Segmentation with Supermodular Losses

Jiaqian Yu, Matthew Blaschko

To cite this version:

Jiaqian Yu, Matthew Blaschko. Efficient Learning for Discriminative Segmentation with Supermodular Losses. British Machine Vision Conference, Sep 2016, York, United Kingdom. hal-01349000

HAL Id: hal-01349000

https://inria.hal.science/hal-01349000v1

Submitted on 26 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Learning for Discriminative Segmentation with Supermodular Losses

Jiaqian Yu
jiaqian.yu@centralesupelec.fr

CentraleSupélec
Université Paris-Saclay & Inria
Châtenay-Malabry, France

Matthew B. Blaschko
matthew.blaschko@esat.kuleuven.be

Center for Proc. Speech and Images
Dept. Elektrotechniek - ESAT
KU Leuven, Belgium

Abstract

Several supermodular losses have been shown to improve the perceptual quality of image segmentation in a discriminative framework such as a structured output support vector machine (SVM). These loss functions do not necessarily have the same structure as the segmentation inference algorithm, and in general, we may have to resort to generic submodular minimization algorithms for loss augmented inference. Although these come with polynomial time guarantees [11, 12, 15], they are not practical to apply to image scale data. Many supermodular losses come with strong optimization guarantees, but are not readily incorporated in a loss augmented graph cuts procedure. This motivates our strategy of employing the alternating direction method of multipliers (ADMM) decomposition for loss augmented inference. In doing so, we create a new API for the structured SVM that separates the maximum a posteriori (MAP) inference of the model from the loss augmentation during training. In this way, we gain computational efficiency, making new choices of loss functions practical for the first time, while simultaneously making the inference algorithm employed during training closer to the test time procedure. We show improvement both in accuracy and computational performance on the Microsoft Research Grabcut database and a brain structure segmentation task, empirically validating the use of a supermodular loss during training, and the improved computational properties of the proposed ADMM approach over the Fujishige-Wolfe minimum norm point algorithm.

1 Introduction

Discriminative structured prediction is a valuable tool in computer vision that has been applied to a wide range of application areas, and in particular object detection and segmentation [2, 6, 25, 27, 28, 35]. It is frequently applied using variants of the structured output support vector machine (SVM) [38, 39] in which a domain specific discrete loss function is upper bounded by a piecewise linear surrogate. In the case of image segmentation, this discrete loss function has frequently been taken to be the Hamming loss, which simply counts the number of incorrect pixels (see e.g. [2, 35]). Following the principle of empirical risk minimization, one might expect that minimization of the desired loss at training time would lead to the best performing loss at test time. However, it has recently been shown that in the

© 2016. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

finite sample regime, minimizing a different loss can lead to better performance even when measured using Hamming loss [27]. In that work, a supermodular loss function was employed, and a custom graph cuts solution was found to the loss augmented inference problem necessary for computation of a subgradient or cutting plane of the learning objective [16].

Several non-modular loss functions have been considered in the context of image segmentation, e.g. the intersection over union loss in the context of a Bayesian framework [24], an area/volume based label-count loss that enforces high-order statistics [28], or a layout-aware loss function that takes into account the topology/structure of the object [27]. A message passing based optimization scheme is proposed for optimizing several families of structured loss functions [36, 37], which assumes the loss function is constructed by a grammar for which the productions specify function composition [37]. In general, it is a time consuming process to develop custom loss-augmented solvers for different combinations of loss functions and inference procedures.

An alternative approach is to resort to generic submodular optimization algorithms, such as that of Iwata [15] which has complexity $\mathcal{O}(n^4T + n^5 \log M)$, or Orlin [26] with complexity $\mathcal{O}(n^6 + n^5T)$, where T is the time for a single function evaluation and M is an upper bound on the absolute value of the function. Although these optimization algorithms are polynomial, the exponent is sufficiently large as to render them infeasible for images of even less than one megapixel. In practice, the Fujishige-Wolfe minimum norm algorithm [11, 12] is empirically faster [9]. However, we will show that even this state of the art optimization strategy is infeasible for relatively small consumer images.

Specific subclasses of submodular functions come with lower complexity optimization algorithms, and we should be able to exploit these known classes in a general learning framework. Examples include decomposable submodular functions [23, 34], several notions of symmetry [17, 29], and graph partition problems [10, 18]. A problem with the current API for loss augmented inference is that it is assumed that the loss function will decompose with a structure compatible to that of the inference problem. We address the case that this assumption does not hold and that separate efficient optimization procedures are available for the loss and for inference.

We propose to use Lagrangian splitting techniques to separate loss maximization from the inference problem. Strategies such as dual decomposition have become popular in Markov Random Fields (MRF) inference [19], while later developments such as the alternating direction method of multipliers (ADMM) [5, 6] have improved convergence guarantees. Other strategies involving a quadratic penalty term have also been proposed in the literature (although still with the assumption that the loss decomposes as the inference) [22]. We make use of ADMM to separate these inference problems and apply them to a supermodular loss function that cannot be straightforwardly incorporated in a submodular graph partition problem for loss augmented inference. Instead we allow separate optimization strategies for the loss maximization and inference procedures yielding substantially improved computational performance, while making feasible the application of a wide range of supermodular loss functions by changing a single line of code.

2 Methods

We discriminatively train a graph cuts based segmentation system using a structured SVM [39]. We construct a supermodular loss function that is solvable with graph cuts, but that when incorporated in a joint loss-augmented inference leads to non-submodular potentials which

causes graph cuts based optimization to fail. We therefore use an ADMM based decomposition strategy to perform loss augmented inference. This strategy consists of alternatingly optimizing the loss function and performing maximum *a posteriori* (MAP) inference, with each process augmented by a quadratic term enforcing the labeling determined by each to converge to the optimum of the sum.

The structured output SVM is a discriminative learning framework that has been applied in diverse computer vision applications. Given a training set of labeled images $\{(x_1, y_1^*), \ldots, (x_n, y_n^*)\} \in (\mathcal{X} \times \mathcal{Y})^n$, where $\mathcal{Y} = \{-1, 1\}^p$ for a binary segmentation problem, it optimizes a regularized convex upper bound to a structured loss function, $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. Δ measures the mismatch between a ground truth labeling and a hypothesized labeling. With Δ provided as an input, the structured SVM with margin rescaling minimizes [39]:

$$\min_{w,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \qquad \text{s.t. } \forall i, \tilde{y}_i \in \mathcal{Y}, \tag{1}$$

$$\langle w, \phi(x_i, y_i^*) - \phi(x_i, \tilde{y}_i)\rangle \geq \Delta(y_i^*, \tilde{y}_i) - \xi_i \tag{2}$$

In the case of image segmentation, we may interpret $\langle w, \phi(x, y)\rangle$ as a function that is monotonic in the log probability of the joint configuration of observed and unobserved variables (x, y) as determined by a CRF [21]. Under this interpretation, a standard definition of ϕ is

$$\phi(x, y) := \begin{pmatrix} \sum_{j=1}^{p} \phi_u(x, y^j) \\ \sum_{(k,l)\in\mathcal{E}} \phi_p(x, y^k, y^l) \end{pmatrix} \tag{3}$$

where ϕ_u determines a vector of features, a linear combination of which form the unary potentials of the CRF, and ϕ_p determines the pairwise potentials over a model specific edge set $\mathcal{E}$. In this work, we have set $\phi_p(x, \cdot, \cdot) : \{-1, 1\}^2 \to \{0, 1\}^3$ to map to an indicator vector of three cases: (i) $y^k = y^l = -1$, (ii) $y^k \neq y^l$, or (iii) $y^k = y^l = +1$, and have placed hard constraints on the corresponding entries of w in the optimization of the structured SVM to ensure that the pairwise potentials in the energy minimization problem remain submodular [41].

During training of the structured SVM, we must perform *loss augmented inference* in order to compute a subgradient of the loss. In the case of margin rescaling, this consists of computing

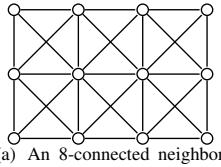
$$\arg\max_{\tilde{y}\in\mathcal{Y}} \langle w, \phi(x, \tilde{y})\rangle + \Delta(y^*, \tilde{y}). \tag{4}$$

If $\mathcal{Y}$ is isomorphic to $\{-1, 1\}^p$ for some p, $\Delta(y^*, \cdot)$ will be isomorphic to a set function $\ell : \mathcal{P}(V) \to \mathbb{R}_+$ where $\mathcal{P}(V)$ is the power set of a base set with $|V| = p$. In particular, we are interested in Δ corresponding to a supermodular set function ℓ [33, 40]:

Definition 1 (Supermodularity)**.** *A supermodular function is a set function $\ell : \mathcal{P}(V) \to \mathbb{R}$ which satisfies: for every $A, B \subseteq V$ with $A \subseteq B$ and every $v \in V \setminus B$ we have that $\ell(A \cup \{v\}) - \ell(A) \leq \ell(B \cup \{v\}) - \ell(B)$. A function is submodular if its negative is supermodular.*

As we have guaranteed that maximization of $\langle w, \phi(x_i, \tilde{y}_i)\rangle$ with respect to $\tilde{y}$ corresponds to a submodular minimization problem, the loss augmented inference as in Equation (4) remains a submodular minimization, when Δ is supermodular and can be aligned with the inference, and therefore polynomial time solvable. By contrast, non-supermodular Δ result in NP-hard optimization problems in general.

Modular loss functions, such as Hamming loss, can be incorporated into the unary potentials in a graph cuts optimization framework for loss augmented inference. However, the



(a) An 8-connected neighborhood is used in the construction of the loss function.

$$E = -\overbrace{\begin{pmatrix} w_{00} & w_{01} \\ w_{10} & w_{11} \end{pmatrix}}^{\text{inference pairwise potential}} - \underbrace{\begin{pmatrix} 0 & \gamma \\ 0 & 0 \end{pmatrix}}_{\text{loss pairwise potential}}$$

(b) Pairwise potential construction for an edge with $y^{*k} = +1$ and $y^{*l} = -1$ following the loss function in Equation (5).

Figure 1: Non-submodularity of the joint loss augmented inference procedure using the same mapping to a set function for inference and loss functions..

formulation of loss augmented inference with supermodular losses as a graph cuts problem is not straightforward. Moreover, while supermodular loss functions guarantee polynomial time solvability, they do not do so with low order polynomial guarantees in general. We have observed that the Fujishige-Wolfe algorithm is infeasible to apply even in the case of sub-megapixel images, and scales poorly for useful supermodular loss functions. Consequently, we develop a general framework for decomposing loss augmented inference based on ADMM. This framework solely relies on a loss function being able to be efficiently optimized in isolation using a specialized solver specific to the loss function.

2.1 A supermodular loss function for binary image segmentation

We propose a loss function that is itself optimizable with graph cuts. The loss simply counts the number of incorrect pixels plus the number of pairs of neighboring pixels that both have incorrect labels

$$\Delta(y^*, \tilde{y}) = \sum_{j=1}^{p} [y^{*j} \neq \tilde{y}^j] + \sum_{(k,l)\in\mathcal{E}_\ell} \gamma [y^{*k} \neq \tilde{y}^k \wedge y^{*l} \neq \tilde{y}^l] \tag{5}$$

where $[\cdot]$ is Iverson bracket notation, $\mathcal{E}_\ell$ is a loss specific edge set and γ is a positive weight. We have used 8-connectivity for the loss function in the experiments (Figure 1(a)), referred to as "8-connected loss" in the sequel. We may identify this function with a set function to which the argument is the set of mispredicted pixels.

Proposition 1. *Maximization of the loss function in Equation (5) is equivalent to a supermodular function maximization problem.*

Proof sketch. Equation (5) is isomorphic to a binary random field model for which label is 1 iff a pixel has a different label from the ground truth. Neighboring pixels that both have label 1 contribute a positive amount to the energy, while all other configurations contribute zero. This corresponds to a supermodular function following Definition 1. □

This loss function emphasizes the importance of correctly predicting adjacent groups of pixels, e.g. those present in thin structures more than one pixel wide. While the pairwise potential in $\langle w, \phi(x,y)\rangle$ has a tendency to reduce the perimeter of the segment, the loss strongly encourages the correct identification of adjacent pixels. We will observe in the experimental results that the use of this loss function during training improves the test time prediction accuracy, even when measuring in terms of Hamming loss.

It may appear at first glance that the structure of this loss function is aligned with that of the inference, and that we can therefore jointly optimize the loss augmented inference with

a single graph cuts procedure. Indeed, the loss function is isomorphic to a supermodular set function, and the inference is isomorphic to a supermodular set function, both of which can be solved by graph cuts. However, the isomorphisms are not the same. The loss function maps to a set function by considering the set of pixels that are incorrectly labeled, while the inference maps to a set function by considering the set of pixels that are labeled as foreground. Shown in Figure 1 is the pairwise potential for an edge with $y^{*k} = +1$ and $y^{*l} = -1$. If we apply a single mapping, the inference procedure can be solved by graph cuts when the sum of the diagonal elements of E is less than the sum of the off diagonal elements. While it is enforced during optimization that $w_{00} + w_{11} - w_{01} - w_{10} \geq 0$, the presence of γ in the off diagonal, for which the exact position depends on the value of y^*, removes the guarantee of a resulting submodular minimization problem. We therefore consider a Lagrangian based splitting method to solve the loss augmented inference problem.

2.2 ADMM algorithm for loss augmented inference

Several Lagrangian based decomposition frameworks have been proposed, such as dual decomposition and ADMM [6], with the latter having improved convergence guarantees. We have also observed a substantial improvement in performance using ADMM over dual decomposition in our own experiments. Here we consider a splitting method to optimize the minimization of the negative of Equation (4), which is equivalent to finding the most violated constraint in cutting plane optimization:

$$\arg\min_{y_a, y_b} -\langle w, \phi(x, y_a)\rangle - \Delta(y^*, y_b) \quad \text{s.t. } y_a = y_b. \tag{6}$$

and we form the augmented Lagrangian as

$$\mathcal{L}(y_a, y_b, \lambda) = -\langle w, \phi(x, y_a)\rangle - \Delta(y^*, y_b) + \lambda^T (y_a - y_b) + \frac{\rho}{2}\|y_a - y_b\|_2^2 \tag{7}$$

where $\rho > 0$. (7) can be optimized in an iterative fashion by Algorithm 1 [6].

The saddle point of the Lagrangian will correspond to an optimal solution over a convex domain, while we are optimizing w.r.t. binary variables. Strictly speaking, we may therefore consider the linear programming (LP) relaxation of our loss augmented inference problem, followed by a rounding post-processing step. We use a standard stopping criterion as in [6]: the primal and dual residuals must be small with an absolute criterion $\varepsilon^{\text{abs}} = 10^{-4}$ and a relative criterion $\varepsilon^{\text{rel}} = 10^{-2}$. In practice, we have found that discretizing the quadratic terms and incorporating them into the unary potentials of the respective graph cuts problems is more computationally efficient, while yielding results that are nearly identical with exact optimization with a primal-dual gap of 0.01%. We show in the experimental results that this strategy yields results almost identical to those of an LP relaxation.

Algorithm 1 ADMM in scaled form for finding a saddle point of the Lagrangian in Eq. (7)

1: Initialization $u^0 = 0$
2: **repeat**
3: $\quad y_a^{t+1} = \arg\min_{y_a} -\langle w, \phi(x, y_a)\rangle + \frac{\rho}{2}(\|y_a - y_b^t + u^t\|_2^2)$
4: $\quad y_b^{t+1} = \arg\min_{y_b} -\Delta(y^*, y_b) + \frac{\rho}{2}(\|y_a^{t+1} - y_b + u^t\|_2^2)$
5: $\quad u^{t+1} = u^t + (y_a^{t+1} - y_b^{t+1})$
6: $\quad t = t + 1$
7: **until** stopping criterion satisfied

In general, we simply need task-specific solvers for lines 3 and 4 of Algorithm 1. These solvers need not use a single graph cut algorithm, and can therefore exploit any available

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2: Example images and the extracted features. 2(a) original RGB image; 2(b) groundtruth; 2(c) the user-labelled seeds; 2(d) the extended seeds; 2(e) the distance features to foreground seed based on RGB space; 2(f) the distance features to background seed based on RGB space; 2(g) the GMM appearance model based on RGB space; 2(h) the distance features to foreground seed based on the RGB-space GMM appearance.

structure even though it may not be present, or aligned, between the two subproblems. Although we have used this framework for the specific supermodular loss function described in the previous subsection, we note that this provides an API for the structured output SVM framework alternate to that provided by SVMstruct [39].

3 Experimental Results

In this section, we consider a foreground/background segmentation task. We compare the prediction using our proposed supermodular loss function with the prediction using Hamming loss. We show that: (i) our proposed splitting strategy is orders of magnitude faster than the minimum norm point algorithm; (ii) our strategy yields results nearly identical to a LP-relaxation while being much faster in practice; and (iii) training with the same supermodular loss as during test time yields better performance.

$\gamma = 0.25$		Eval.		
		Δ(1e3)	0-1(1e3)	IoU
Train.	Δ	$\mathbf{6.3562 \pm 1.065}$	3.3378 ± 0.5462	0.2111 ± 0.0152
	0-1	7.8641 ± 1.0437	4.1548 ± 0.5378	0.2399 ± 0.0170
$\gamma = 0.5$		Eval.		
		Δ(1e3)	0-1(1e3)	IoU
Train.	Δ	$\mathbf{9.0483 \pm 1.3457}$	3.2801 ± 0.4687	0.2079 ± 0.0155
	0-1	11.582 ± 1.5495	4.1548 ± 0.5378	0.2399 ± 0.0170
$\gamma = 1.0$		Eval.		
		Δ(1e3)	0-1(1e3)	IoU
Train.	Δ	$\mathbf{14.908 \pm 2.4102}$	3.4145 ± 0.4108	0.2084 ± 0.0160
	0-1	19.019 ± 2.5613	4.1458 ± 0.5378	0.2399 ± 0.0170

Table 1: The cross comparison of average loss values (with standard error) using different loss functions during training and during testing on the Grabcut dataset. Training with the same supermodular loss functions as used during testing yields the best results. Training with supermodular losses even outperforms the Hamming loss in terms of evaluating by Hamming loss.

Datasets The dataset provided by [4, 14] contains 151 images in total, including the color images in RGB space, the ground truth foreground/background segmentation and the user-labelled seeds (see Figure 2(a), Figure 2(b), and Figure 2(c), respectively). As we are discriminatively training a class specific segmentation system in our experiments, we focus on the images in which the foreground objects are *people*. We compute in total 18 unary features following [27]. Figure 2(e) to Figure 2(h) show examples of the extracted features.

We additionally utilise the Internet Brain Segmentation Repository (IBSR) dataset [30], which consists of T1-weighted MR images. Images and masks have been linearly registered and cropped to $145 \times 158 \times 123$. We choose one horizontal slice within each volume and we follow the feature extraction procedure as in [1].

Training and Testing We use the ADMM splitting strategy to solve the minimization problem in Equation (6). We use the GCMex - MATLAB wrapper for the Boykov-Kolmogorov graph cuts algorithm [7, 8, 13, 18] to solve the optimization problems on lines 3 and 4 in Algorithm 1, i.e. for the inference part and for the loss part separately. Results computed with different values of $\gamma > 0$ are shown in Table 1 and Table 2. During the training stage, we use $\rho = 0.1$ for the ADMM step-size parameter. The regularization parameter C in Equation (1) is chosen by cross-validation in the range $\{10^i | -2 \leq i \leq 2\}$. We additionally train and test with Hamming loss as a comparison. At test time, we have computed the unnormalized Hamming loss, the intersection over union loss (IoU), and our 8-connected loss for each training scenario. We have performed several random train-test splits in order to compute error bars on the loss estimates.

		Eval.		
$\gamma = 0.5$		Δ(1e3)	0-1(1e3)	IoU
Train.	Δ	$\mathbf{2.616 \pm 0.612}$	1.297 ± 0.224	0.169 ± 0.018
	0-1	2.885 ± 0.765	1.393 ± 0.279	0.173 ± 0.019

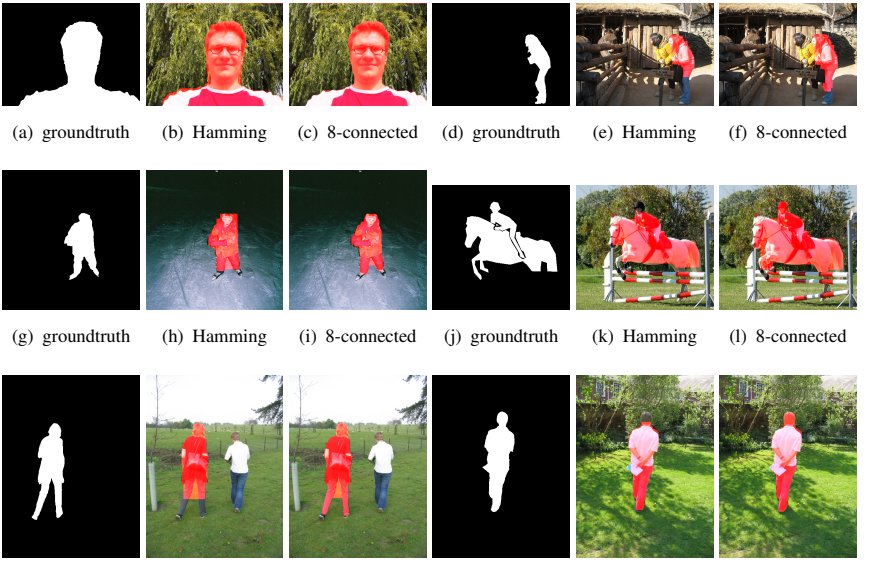
Table 2: The cross comparison of average loss values on IBSR dataset (cf. comments for Table 1).

Computation Time We compare the time of one calculation of the loss augmented inference by the ADMM algorithm and by the minimum norm point algorithm [12] (MinNorm). For MinNorm, we use the implementation provided in the SFO toolbox [20]. Although it has been proven that in t iterations, the MinNorm returns an $O(1/t)$-approximate solution [9], the first step of this algorithm is to find a point in the submodular polytope, which alone is computationally intractable even for small 600×400 pixel images. Therefore, we measure the computation time on downsampled images, showing the growth in computation as a function of image size (Figure 5 and Figure 6). The running times are recorded on a machine with a 3.20GHz CPU. Similarly, a dual-decomposition baseline took orders of magnitude longer computation than the ADMM approach, following known convergence results [6].

Results As shown in Table 1 and Table 2, training with the same supermodular loss as used for testing has achieved the best performance. Training with the supermodular loss even outperforms training with Hamming loss when measured by Hamming loss on the test set, with a reduction in error of 17.2%. A Wilcoxon sign rank test shows that training with Δ gives significantly better results in all cases ($p \leq 2 \times 10^{-3}$).We have additionally tried training with a joint graph cuts loss augmented inference using the pairwise potentials illustrated in Figure 1. However, due to the non-submodular potentials, the graph cuts procedure does not correctly minimize the energy resulting in incorrect cutting planes that causes optimization to fail after a small number of iterations. The performance of this system was effectively random, and we have not included these values in Table 1.

Qualitative segmentation results are shown in Figure 3, and in Figure 4 we show a pixelwise comparison of the predictions. The 8-connected loss achieves better performance on the foreground/background boundary, as well as on elongated structures of the foreground object, such as the head and legs, especially when the appearance of the foreground is similar to the background.

We measure the computation time for 120 calculations of the loss augmented inference by ADMM and MinNorm on different sized images. From Figure 5 and Figure 6 we can see that ADMM is always faster than the MinNorm by a substantial margin, and around 100 times faster when the problem size reaches 10^3. The computing time for both ADMM and MinNorm vary approximately linearly in log-log scale, while MinNorm has a higher



(a) groundtruth (b) Hamming (c) 8-connected (d) groundtruth (e) Hamming (f) 8-connected

(g) groundtruth (h) Hamming (i) 8-connected (j) groundtruth (k) Hamming (l) 8-connected

(m) groundtruth (n) Hamming (o) 8-connected (p) groundtruth (q) Hamming (r) 8-connected

Figure 3: The segmentation results of prediction trained with Hamming loss (columns 2 and 5) and our supermodular loss (columns 4 and 6). The supermodular loss performs better on foreground object boundary than Hamming loss does, and it achieves better prediction on the elongated structures of the foreground object e.g. the heads and the legs.

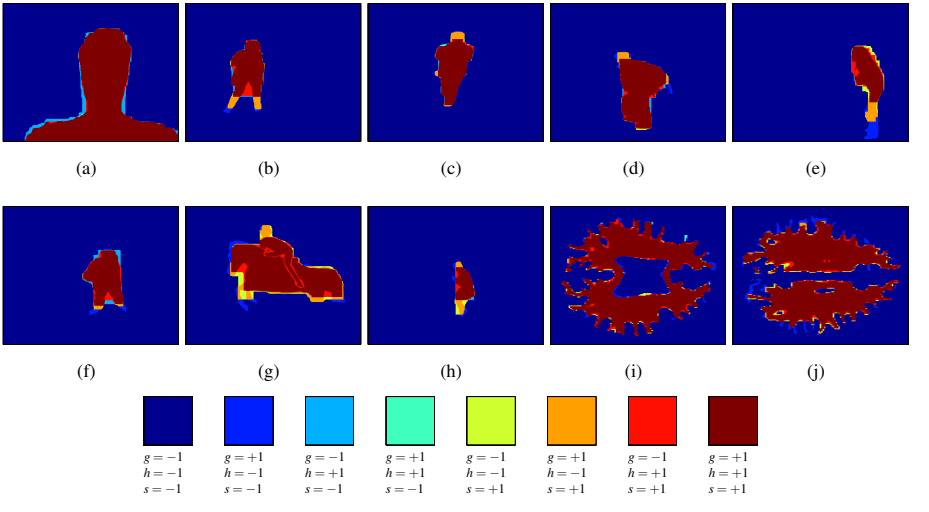
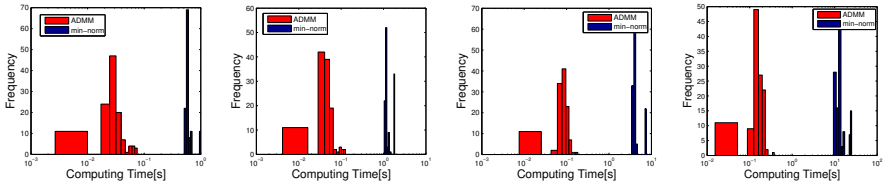


Figure 4: A pixelwise comparison of the ground truth (denoted g in the legend), the prediction from training with Hamming loss (denoted h) and the prediction when training with the proposed supermodular loss (denoted s). There are many regions in the set of images where the supermodular loss learns to correctly predict the foreground when Hamming loss fails (orange regions corresponding to $g = +1$, $h = -1$, and $s = +1$). (a)-(h) show the semantic segmentation task [14], while (i)-(j) show the structural brain segmentation task [30].



(a) Size=600 (b) Size=1200 (c) Size=2400 (d) Size=4800

Figure 5: The computing time for the loss augmented inference, on different problem sizes. The red histograms stands for ADMM and the blue for MinNorm. The calculation by ADMM is always faster than by MinNorm, and there is no overlap between the computing time by the two methods.

	$-E$	size $=600$	size $=1200$	size $=2400$
ADMM	2.28 ± 0.58	0.035 ± 0.002	0.051 ± 0.002	0.864 ± 0.476
LP	2.29 ± 0.57	1.857 ± 0.128	3.946 ± 0.286	13.57 ± 1.359

Table 3: The comparison between ADMM and an LP relaxation for solving the loss augmented inference. The 1st column shows the optimal energy values (10^3) (Equation (4)); columns 2–4 show the computation time (s) for one calculation on downsampled images of varying size.

slope, suggesting a worse big-$\mathcal{O}$ computational complexity. We note that theoretical bounds on MinNorm are currently weak and the exact complexity is unknown [9]. Although it is immediately clear from Figure 6 that ADMM is substantially faster than the minimum norm point algorithm, we have performed Wilcoxon sign rank tests that show this difference is significant with $p < 10^{-20}$ in all settings.

We also ran a baseline comparing non-submodular loss augmented inference with the QPBO approach [31]. We computed pairwise energies as in Figure 1(a). QPBO found loss augmented energies across the dataset of $1.1 \times 10^6 \pm 3 \times 10^5$ while ADMM found loss augmented energies of $3.7 \times 10^6 \pm 8 \times 10^5$, a substantial improvement.

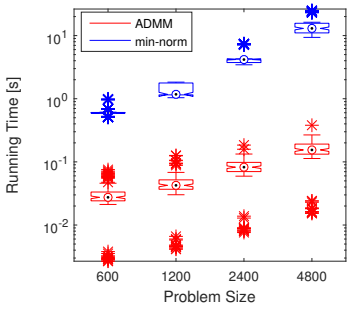


Figure 6: Computation time. ADMM is substantially faster than the min-norm algorithm.

Comparison to LP-relaxation We additionally compare ADMM to an LP relaxation procedure for the loss augmented inference to determine the accuracy of our optimization in practice, with using the 8-connected loss function and the Hamming loss (0-1). For the implementation of the LP relaxation, we use the `UGM` toolbox [32]. We show in Table 3 the comparison between using ADMM and the LP relaxation. The first column represents the energy achieved by the loss augmented inference (Equation (4)). We observe that the (maximal) energy achieved by ADMM is almost the same as the LP relaxation: a difference of 0.4%. Columns 2–4 show the computing time for one calculation of the loss augmented inference on the downsampled images. Using an LP relaxation, the computation time is orders of magnitude slower, growing as a function of the image size. ADMM provides a more efficient strategy without loss of performance.

4 Discussion and Conclusion

A somewhat surprising result in Table 1 is that training with the supermodular loss results in better performance *as measured by Hamming loss*. This has been previously observed with a different loss function by [27, 28], and indicates that in the finite sample regime a supermodular likelihood can result in better generalization performance. This holds, although the model space and regularizer were identical in both training settings. We believe that further exploration of the properties of supermodular loss functions is warranted in this regard.

Our results in terms of computation time give clear evidence for the superiority of ADMM inference when a specialized optimization procedure is available for the loss function. As shown in Figure 6, the Fujishige-Wolfe minimum norm point algorithm does not scale to typical consumer images (i.e. several megapixels), which indicates that loss functions for which a specialized optimization procedure is not available are likely infeasible for pixel level image segmentation without unprecedented improvements in general submodular minimization. Figure 6 shows that the log-log slope of the runtime for the min-norm point algorithm is higher than for ADMM, suggesting a worse computational compexity. One may wish to employ the result that early termination of the min-norm point algorithm gives a guaranteed approximation of the exact result, but even this is infeasible for images of the size considered here. Joint graph-cuts optimization for loss augmented inference results in non-submodular pairwise potentials and graph-cuts fails to correctly minimize the joint energy. As a result, a cutting plane optimization of the structured output SVM objective fails catastrophically, and the resulting accuracy is on par with a random weight vector.

In this work, we have shown that a supermodular loss function achieves improved performance both in qualitative and quantitative terms on a binary segmentation task. We observe that a key advantage of the proposed supermodular loss over modular losses, e.g. Hamming loss, is an improved ability to find elongated regions such as heads and legs, or thin articulated structures in medical images.

Previous to our work, specialized inference procedures had to be developed for every model/loss pair, a time consuming process. Our proposed ADMM algorithm provides a strategy to solve the loss augmented inference as two separate subproblems. This provides an alternate API for the structured output SVM framework to that of SVMstruct [39]. We envision that this can be of use in a wide range of application settings, and an open source general purpose toolbox for this efficient segmentation framework with supermodular losses is available for download from `https://github.com/yjq8812/efficientSegmentation`.

Acknowledgements

This work is funded by Internal Funds KU Leuven, ERC Grant 259112, FP7-MC-CIG 334380, and the Research Foundation Flanders (FWO) through project number G0A2716N. The first author is supported by a fellowship from the China Scholarship Council.

References

[1] Stavros Alchatzidis, Aristeidis Sotiras, and Nikos Paragios. Discrete multi atlas segmentation using agreement constraints. In *BMVC*, 2014.

[2] Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta,

Geremy Heitz, and Andrew Ng. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *CVPR*, volume 2, pages 169–176, 2005.

[3] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena, 1999.

[4] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, pages 428–441, 2004.

[5] Matthew B. Blaschko and Christoph H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[7] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *T-PAMI*, 26(9):1124–1137, 2004.

[8] Yuri Boykov, Olga Veksler, and Ramin Zabih. Efficient approximate energy minimization via graph cuts. *T-PAMI*, 20(12):1222–1239, 2001.

[9] Deeparnab Chakrabarty, Prateek Jain, and Pravesh Kothari. Provable submodular minimization using Wolfe's algorithm. In *NIPS*, 2014.

[10] G. Charpiat. Exhaustive family of energies minimizable exactly by a graph cut. In *CVPR*, 2011.

[11] Satoru Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980.

[12] Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

[13] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

[14] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, pages 3129–3136, 2010.

[15] Satoru Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4):833–840, 2003.

[16] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

[17] Vladimir Kolmogorov. Minimizing a sum of submodular functions. *Discrete Applied Mathematics*, 160(15):2246–2258, 2012.

[18] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *T-PAMI*, 26(2):147–159, 2004.

[19] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[20] Andreas Krause. SFO: A toolbox for submodular function optimization. *JMLR*, 11: 1141–1144, 2010.

[21] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[22] Ofer Meshi, Nathan Srebro, and Tamir Hazan. Efficient training of structured svms via soft constraints. In *AISTATS*, pages 699–707, 2015.

[23] Robert Nishihara, Stefanie Jegelka, and Michael I. Jordan. On the convergence rate of decomposable submodular function minimization. In *NIPS*, pages 640–648, 2014.

[24] Sebastian Nowozin. Optimal decisions from probabilistic models: the intersection-over-union case. In *CVPR*, 2014.

[25] Sebastian Nowozin and Christoph H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4): 185–365, 2011.

[26] James B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[27] Anton Osokin and Pushmeet Kohli. Perceptually inspired layout-aware losses for image segmentation. In *ECCV*, 2014.

[28] Patrick Pletscher and Pushmeet Kohli. Learning low-order models for enforcing high-order statistics. In *AISTATS*, 2012.

[29] Maurice Queyranne. Minimizing symmetric submodular functions. *Mathematical Programming*, 82(1-2):3–12, 1998.

[30] Torsten Rohlfing. Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable. *IEEE Transactions on Medical Imaging*, 31(2):153–163, 2012.

[31] Carsten Rother, Vladimir Kolmogorov, Victor Lempitsky, and Martin Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007.

[32] Mark Schmidt. UGM: Matlab code for undirected graphical models, 2012.

[33] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2004.

[34] Peter Stobbe and Andreas Krause. Efficient minimization of decomposable submodular functions. In *NIPS*, 2010.

[35] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning CRFs using graph cuts. In *ECCV*, pages 582–595, 2008.

[36] Daniel Tarlow and Richard S Zemel. Structured output learning with high order loss functions. In *AISTATS*, 2012.

[37] Daniel Tarlow, Inmar E. Givoni, and Richard S. Zemel. HOP-MAP: Efficient message passing with high order potentials. In *AISTATS*, 2010.

[38] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In *NIPS*, 2003.

[39] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. In *JMLR*, pages 1453–1484, 2005.

[40] Jiaqian Yu and Matthew B. Blaschko. Learning submodular losses with the Lovász hinge. In *ICML*, pages 1623–1631, 2015.

[41] Wojciech Zaremba and Matthew B. Blaschko. Discriminative training of CRF models with probably submodular constraints. In *WACV*, 2016.