Cutting-Plane Training of Non-associative Markov Network for 3D Point Cloud Segmentation

Roman Shapovalov [a] Alexander Velizhev [a,b]

[a] *Graphics & Media Lab, Lomonosov Moscow State University*
[b] *Moscow State University of Geodesy and Cartography*
{shapovalov, avelizhev}@graphics.cs.msu.ru

***Abstract*—We address the problem of object class segmentation of 3D point clouds. Each point of a cloud should be assigned a class label determined by the category of the object it belongs to. Non-associative Markov networks have been applied to this task recently. Indeed, they impose more flexible constraints on segmentation results in contrast to the associative ones. We show how to train non-associative Markov networks in a principled manner using the structured Support Vector Machine (SVM) formalism. In contrast to prior work we use the kernel trick which makes our method one of the first non-linear methods for max-margin Markov Random Field training applied to 3D point cloud segmentation. We evaluate our method on airborne and terrestrial laser scans. In comparison to the other non-linear training techniques our method shows higher accuracy.**

***Keywords*-semantic segmentation; LIDAR; conditional random field; structured learning; cutting-plane training**

I. Introduction

Light Detection and Ranging (LIDAR) technology has become commonly available recently. Airborne and terrestrial laser scanners produce a lot of data which need to be analyzed. Object class segmentation is an important step in the process of point cloud understanding. Each point of a scan should be assigned a class label determined by the category of the object it belongs to. For example, it is possible to split terrestrial scans into buildings, trees, cars, ground regions, etc. (Fig. 1). Machine learning methods are often used for these purposes: a classifier is built based on the features of a train scan. Afterwards, different point clouds are labeled by applying this classifier. Points can be treated independently (e.g. [1]), but since individual points are classified without context, their features should be computed within a large spatial support, which is computationally intensive and often does not afford the desired accuracy.

The Markov Random Field (MRF) framework provides a natural way of incorporating local context. In their seminal work Anguelov et al. propose to use associative Markov networks (AMN) for segmentation [2]. Associative Markov networks encourage neighboring points to have same class labels. The method only performs smoothing in an intelligent manner, but it improves the results significantly because individual classification is usually not robust. They minimize an energy function defined over the graph representing some neighborhood system on the points of a scan. The energy is a sum of potential functions of the features corresponding to individual nodes or edges extracted from the scan. Since the form of the energy function is simple enough, a graphcut-based method is used for minimization. The parameters of the energy are learned by reducing the training problem to quadratic programming (QP).

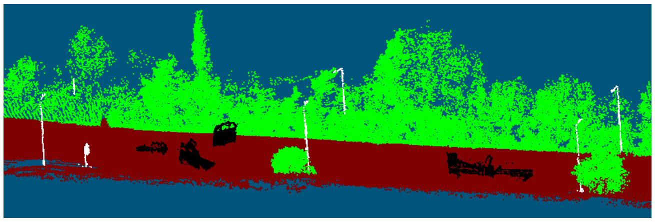


Figure 1. Typical terrestrial LIDAR scan. Hand-crafted labeling is color-coded: red — ground, black — vehicle, green — vegetation, white — pole.

In practice, complicated edge features are not very significant when the associative model is used. The model is not flexible enough to make use of edge features because they can only impose the intention to belong to the same class for neighboring points. As a result, in the early papers the constant pairwise feature is used [2], [3], i.e. pairwise potential serve only as a prior for class co-occurrence. However, Munoz et al. [4] use an anisotropic model, where pairwise potentials depend on the edge features, like in Conditional Random Fields (CRF). They also propose to use higher-order cliques within the CRF and show how to train the new model [5], but do not abandon the associativity constraint. In their model clique potentials (both pairwise and higher-order) are allowed to be positive *only* if all the variables in the clique share a common label, otherwise they are equal to zero. Associativity is indeed a hard constraint. In our prior work we have shown that it makes impossible to express asymmetric dependencies like “trees and buildings are usually above the ground” [6]. We use the general form of pairwise potentials. Our method is more exacting to a train set: interclass interactions should be well represented in addition to the intraclass ones learned by AMNs. Naïve Bayes is used to learn the parameters. The drawback of that approach is ignoring correlations between neighboring points during learning. Posner et al. also use a non-associative Markov network, which is a particular case of ours: they do not model the dependency of edge potential on the scan, which corresponds to using only prior distributions in our model [7]. Such simple Markov network does not need to be trained, only class frequencies should be estimated.

The problem of training non-associative Markov networks has *not* been deeply investigated. The essential difference is this form of energy is irregular, and there are no exact methods for minimizing it (c.f. Section II). While the accuracy of inexact methods is usually sufficient for inference, it may cause troubles during training. Finley and Joachims address the problem of structured learning when no exact inference algorithm is available [8]. Franc and Savchinskyy show how to train different categories of max-sum classifiers, including non-associative ones [9]. However, both papers are mostly theoretical, and the MRF formulations they use for experiments are simple.

We define more complicated pairwise potentials that handle the segmentation problem well and show how to train this non-associative Markov network in a principled manner using the structured Support Vector Machine (SVM) formalism [10], which is the main contribution of this paper. Since real scans are usually class-imbalanced, learning from imbalanced scans is an important practical issue. We show how to account the imbalance by changing the loss function. We also modify the original feature space with the radial basis function kernels, which is helpful when the assignment depends on features nonlinearly. In this sense our method is close to the work by Triebel et al. [3], who combine an AMN with a k nearest neighbor classifier. The difference is our support vectors are not only the correct training scans' labellings, but potentially all possible assignments over them (wrongly labeled scans contribute with negative weights). Since our method is sparse, only few wrongly labeled scans are usually chosen as support vectors.

CRFs are not the only way to account for spatial structure. Object candidate segmentation followed by segment classification is sometimes used [11], [12]. Recently, Xiong et al. proposed Stacked 3D Parsing for incorporating context into segmentation [13]. They learn relational information in coarse and fine scales and then combine that information.

We review the formulation of CRF used for object class segmentation in the following section. Section III describes the structured learning problem and techniques, including the cutting-plane training we adopted. Section IV reports on experimental results followed by a discussion and future work suggestions.

II. CRF FOR POINT CLOUD SEGMENTATION

The assumption that labels of neighboring points are correlated is extensively exploited for point cloud segmentation. Moreover, recent work takes into account the dependency of this interaction on the data. For example, if a point lies one meter above another, they are more likely to belong to the class "pole" altogether than if the second point is one meter to the right. This is usually modeled as a variant of Conditional Random Field. We follow this trend and model the posterior probability of the classification result $\mathbf{y} = \{y_1, \ldots, y_N\}$ given the scan features $\mathbf{x}$ as

$$\begin{aligned}\mathrm{P}(\mathbf{y}|\mathbf{x}) &= \frac{1}{Z(\mathbf{x})} \exp \Phi(\mathbf{x}, \mathbf{y}) \qquad (1)\\ &= \frac{1}{Z(\mathbf{x})} \prod_{i=1}^{N} \exp \phi_n(\mathbf{x}_i, y_i) \prod_{(i,j)\in\mathcal{E}} \exp \phi_e(\mathbf{x}_{ij}, y_i, y_j),\end{aligned}$$

where the partition function $Z(\mathbf{x}) = \sum_{\bar{\mathbf{y}}} \exp \Phi(\mathbf{x}, \bar{\mathbf{y}})$ is a sum over all possible label assignments $\bar{\mathbf{y}}$ that is independent of the assignment $\mathbf{y}$ itself. Each y_i is a random variable that corresponds to one of N points of the scan and takes a value in the range $\{1, \ldots, K\}$, which means one of the K predefined class labels. $\mathcal{E}$ is a neighboring system over points. Usually the edges connecting each point to its k nearest neighbors are added to this set.

We consider the linear form of the potential functions: $\Phi(\mathbf{x}, \mathbf{y}) = \mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \mathbf{y})$, where the vector-valued function Ψ defines the correspondence of the weights $\mathbf{w}$ to relevant features and classes. Node and edge potential functions are thus defined as $\phi_n(\mathbf{x}_i, y_i) = \sum_{k=1}^{K} \mathbf{w}_{n,k}^{\mathrm{T}} \mathbf{x}_i y_i^k$ and $\phi_e(\mathbf{x}_{ij}, y_i, y_j) = \sum_{k=1}^{K} \sum_{l=1}^{K} \mathbf{w}_{e,kl}^{\mathrm{T}} \mathbf{x}_{ij} y_i^k y_j^l$, where y_i^k is a binary indicator variable that is turned on if and only if the i-th node is labeled as belonging to the class k: $y_i^k \equiv [y_i = k]$. The weight vector is thus a concatenation of the weights for all the possible node and edge classes: $\mathbf{w} = [\mathbf{w}_{n,1}, \mathbf{w}_{n,2}, \ldots, \mathbf{w}_{n,K}, \mathbf{w}_{e,11}, \mathbf{w}_{e,12}, \ldots, \mathbf{w}_{e,KK}]$. Note that we redefine the inner product via kernels to go beyond this linear model, see Section III-C for details.

One does not need to know the distribution $\mathrm{P}(\mathbf{y}|\mathbf{x})$ to find its mode, i.e. the most probable assignment $\mathbf{y}$. For some particular cases, a good approximation (or even the global maximum) for

$$\arg\max_{\bar{\mathbf{y}}} \log \mathrm{P}(\bar{\mathbf{y}}|\mathbf{x}) \qquad (2)$$

could be found efficiently using message passing [14], linear programming relaxation [15], or graph cuts [16]. Since the structure of the CRF we use contains cycles, and the potential functions are not regular (non-associativity), loopy belief propagation [14] is not guaranteed to converge at all, and graph cut based methods are not applicable. We use sequential tree-reweighted message passing (TRW-S) for inference [17][1]. It minimizes the objective of the dual problem for the relaxation of (2), which upper bounds the maximum of (2). In practice, the relaxation is usually tight, so minimizing the upper bound leads to a solution having the probability close to optimal.

[1] We used the implementation by Vladimir Kolmogorov: http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/TRW-S.html

III. Structural SVM and cutting-plane training

A. Max-margin formulation

There is a number of ways to tune the model's weights $\mathbf{w}$. One of them is independent learning, i.e. to collect the statistics on the features for each class label (for a node) and each pair of class labels (for an edge) and train a linear classifier, which is shared among the factors. This approach does not take into account correlations between factors, which is often helpful for learning graphical model's parameters. It is natural to maximize the likelihood of weights given a training set of labeled scans. Suppose we have only one training scan $(\mathbf{x}, \mathbf{y})$ (the extension to multiple scans is straightforward). For pairwise CRFs, one needs to maximize

$$\mathrm{P}_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = \frac{\exp \mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \mathbf{y})}{\sum_{\bar{\mathbf{y}}} \exp \mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \bar{\mathbf{y}})}, \tag{3}$$

where the sum in the denominator is over all the possible assignments. In spite of the function is concave on $\mathbf{w}$, gradient-based methods are not suitable here due to intractable computation of the partition function on each iterationunless the graph is tree-structured (e.g. [18]) or some heuristic is used to approximate the likelihood (e.g. [19]).

Hinge loss is usually minimized instead. It corresponds to the margin maximization scheme with slack variables ξ:

$$\begin{aligned} &\min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} \mathbf{w}^{\mathrm{T}} \mathbf{w} + C\xi \\ &\text{s.t. } \mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \mathbf{y}) + \xi \geq \mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}}), \forall \bar{\mathbf{y}}. \end{aligned} \tag{4}$$

Here $\mathbf{w}^{\mathrm{T}}\mathbf{w}$ is a regularization term that penalizes vectors $\mathbf{w}$ of big norm, C trades off the importance of this penalty and the loss. $\Delta(\mathbf{y}, \bar{\mathbf{y}})$ measures the difference between two assignments, typically it is a Hamming distance: $\Delta(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i=1}^{N} [y_i \neq \bar{y}_i]$. This optimization problem is known as a structural Support Vector Machine (SVM) [10][2]. It is actually a standard quadratic optimization problem. Unfortunately, it contains exponentially many constraints (K^N for N variables in CRF and K class labels), so conventional QP techniques are not applicable. Therefore, all the structured learning methods exploit the structure of the problem. The common thing is they substitute the exponential number of linear constraints with a non-linear one:

$$\mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \mathbf{y}) + \xi \geq \max_{\bar{\mathbf{y}}} [\mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}})]. \tag{5}$$

Thus, the outer continuous optimization problem contains the inner integer one. Note that if the loss function could be factored into the sum over nodes and edges, the inner problem could be solved via combinatorial optimization. If the required regularity constraints are hold, the same CRF inference algorithm can be applied.

If point classes are represented in a training scan unevenly, the weights for underrepresented classes tend to be small since erroneous assignments in that nodes do not have significant impact to the loss. It is natural to set the penalty for misclassification in inverse proportion to the frequency of each class. Specifically, we use the weighted Hamming loss: $\Delta(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i=1}^{N} [y_i \neq \bar{y}_i] / freq(y_i)$, where $freq(k) = \sum_{i=1}^{N} [y_i = k]$ for some class label k. Scale of the frequency vector does not really matter since its change is equivalent to changing the constant C in (4), which is usually tuned on a validation set anyway. Our experiments show that such simple imbalance accounting scheme works well if the imbalance is not crucial (when it is still possible to collect enough statistics).

B. Optimization methods

Taskar et al. suggest to relax the inner problem to linear programming and then take the dual [20]. The resulting problem is thus reformulated as a quadratic programming problem over the weights and dual variables of the inner one and can be solved with a standard QP solver, which is tractable for medium-sized problems. This approach has been applied to point cloud segmentation successfully [2], [3].

Instead of taking dual of the relaxed inner problem it is possible to apply a gradient optimization technique to the problem (4) with the non-linear constraint (5). Subgradient formulation is both efficient and have intuitive appeal since it runs the inference algorithm being trained in the loop [21]. Whenever the slack variable ξ is non-negative (which corresponds to the linearly inseparable case, usual in practice), the constraint can be moved to the objective to get an unconstrained problem. One needs to minimize the regularized loss function:

$$\begin{aligned} c(\mathbf{w}) = &\frac{1}{2C} \mathbf{w}^{\mathrm{T}} \mathbf{w} + \\ &\max_{\bar{\mathbf{y}}} [\mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}})] - \mathbf{w}^{\mathrm{T}} \Psi(\mathbf{x}, \mathbf{y}). \end{aligned} \tag{6}$$

The weights vector is updated iteratively according to some subgradient of the loss $g_{\mathbf{w}} \in \partial c(\mathbf{w})$. It can be computed as $\frac{1}{2C} \mathbf{w} + \Psi(\mathbf{x}, \bar{\mathbf{y}}) - \Psi(\mathbf{x}, \mathbf{y})$, where $\bar{\mathbf{y}}$ is the optimal configuration w.r.t. the current weights. The method is adopted by Munoz et al. [4]. In the follow-up work they train CRF with potential functions of non-linear form [5]. Instead of computing a subgradient, they fit a function $h_t()$ having the form of regression tree to get the required update of the potentials. The resulting potential function is thus a weighted sum of such functions over the gradient descent steps t. Since the learning procedure reminds boosting, the algorithm is referred to as the functional gradient boosting learning.

Cutting-plane training is another iterative method for finding maximum margin [10]. The optimization problem (4) has exponentially many constraints, but not all of them contribute to the feasible polytope. The idea is to iteratively

[2] The special case of this problem when parameters of an MRF are tuned is also referred to as max-margin Markov network ($\mathrm{M}^3\mathrm{N}$) learning [20].

Algorithm 1 Cutting-plane algorithm for training structural SVM

```
1:  Input: labelled instance $(\mathbf{x}, \mathbf{y})$, parameters $C$, $\epsilon$.
2:  $\mathcal{W} \leftarrow \emptyset$, $\xi \leftarrow 0$
3:  repeat
4:      $\bar{\mathbf{y}} \leftarrow \arg\max_{\bar{\mathbf{y}}} [\mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}})]$
5:      if $\Delta(\mathbf{y}, \bar{\mathbf{y}}) - \mathbf{w}^{\mathrm{T}}[\Psi(\mathbf{x}, \mathbf{y}) - \Psi(\mathbf{x}, \bar{\mathbf{y}})] \geq \xi + \epsilon$ then
6:          $\mathcal{W} \leftarrow \mathcal{W} \cup \{\bar{\mathbf{y}}\}$
7:          $(\mathbf{w}, \xi) \leftarrow \arg\min_{\mathbf{w}, \xi \geq 0} \ \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\xi$
8:              s.t. $\mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \mathbf{y}) + \xi \geq$
9:                  $\mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}}), \forall \bar{\mathbf{y}} \in \mathcal{W}$
10:     end if
11: until $\mathcal{W}$ has not changed
```

add constraints to the polytope and update the weights. On each iteration the most violated constraint w.r.t. the current weights is added to a working set $\mathcal{W}$. The left hand sides of the constraints in (4) do not depend on the constraint, so the maximum violation is reached for $\bar{\mathbf{y}} = \arg\max_{\bar{\mathbf{y}}} [\mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}, \bar{\mathbf{y}})]$. This problem is called *loss augmented inference* and could be solved via the same inference algorithm that is used without the loss (assuming the loss is factored, i.e. decomposable to the sum over node and edge assignments). The algorithm terminates when the maximum violation does not exceed some precision parameter ϵ. For any fixed ϵ, the algorithm is proven to converge for a polynomial number of steps even if the inference is inexact [8]. See Algorithm 1 for pseudocode.

Cutting-plane CRF training has been applied to computer vision problems. Szummer et al. train CRFs for two-class image segmentation and one-image geometry recognition [22]. However, their energy have associative constraints, and nearly optimal inference can be performed with graph cuts or α-expansion. Franc and Savchinskyy describe different methods for training various max-sum classifiers, including non-associative ones [9]. Their experiments show that non-associative CRFs perform slightly worse in semantic image segmentation. It is explained by inexact inference procedure, which leads to finding non-optimal weights. They use only the constant pairwise feature, so non-associativity cannot help a lot, especially given they use a small training set, where the complicated pairwise interactions are underrepresented, and the classifier is thus overfit.

C. Dual QP and kernels

In practice the number of iterations until convergence is small, it seldom exceeds few hundred. On each step we have a quadratic program to solve (lines 7–9) with a reasonable number of constraints. It is possible to formulate the dual problem:

$$\begin{aligned} \max_{\alpha \geq 0} \ & \sum_{\bar{\mathbf{y}}} \alpha_{\bar{\mathbf{y}}} \Delta(\mathbf{y}, \bar{\mathbf{y}}) - \frac{1}{2} \sum_{\bar{\mathbf{y}}} \sum_{\bar{\mathbf{y}}'} \alpha_{\bar{\mathbf{y}}} \alpha_{\bar{\mathbf{y}}'} H(\bar{\mathbf{y}}, \bar{\mathbf{y}}') \\ \text{s.t.} \ & \sum_{\bar{\mathbf{y}}} \alpha_{\bar{\mathbf{y}}} = C, \end{aligned} \tag{7}$$

where α is a vector of dual variables, and the inner product is defined as

$$\begin{aligned} H(\bar{\mathbf{y}}, \bar{\mathbf{y}}') &= \Psi(\mathbf{x}, \mathbf{y})^{\mathrm{T}}\Psi(\mathbf{x}, \mathbf{y}) - \Psi(\mathbf{x}, \mathbf{y})^{\mathrm{T}}\Psi(\mathbf{x}, \bar{\mathbf{y}}') \qquad (8) \\ &- \Psi(\mathbf{x}, \bar{\mathbf{y}})^{\mathrm{T}}\Psi(\mathbf{x}, \mathbf{y}) + \Psi(\mathbf{x}, \bar{\mathbf{y}})^{\mathrm{T}}\Psi(\mathbf{x}, \bar{\mathbf{y}}') \\ &= K(\mathbf{x}, \mathbf{y}, \mathbf{x}, \mathbf{y}) - K(\mathbf{x}, \mathbf{y}, \mathbf{x}, \bar{\mathbf{y}}') \qquad (9) \\ &- K(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{x}, \mathbf{y}) + K(\mathbf{x}, \bar{\mathbf{y}}, \mathbf{x}, \bar{\mathbf{y}}'). \end{aligned}$$

Since feature vectors contribute to the problem (7) only via inner product, it is possible to generalize it using kernels (9) similar to the way it is done for classification SVM [10]. Instead of solving the primal problem on each iteration, it is also possible to solve the dual one. The solution is thus formulated in terms of support vectors. For each constraint $\bar{\mathbf{y}}$ that does not belong to the working set (and for some that belong), the corresponding $\alpha_{\bar{\mathbf{y}}} = 0$, so the method is sparse, and the number of support vectors is upper bounded by the number of iterations. Constraints in the working set are the most *unlikely* assignments, so intuitively the best assignment should be far from the support vectors.

Once kernel SVM is trained on a scan $(\mathbf{x}, \mathbf{y})$, one can find the most probable labeling (2) for a scan $\mathbf{x}'$ as

$$\arg\max_{\mathbf{y}'} \sum_{\bar{\mathbf{y}}} \alpha_{\bar{\mathbf{y}}} [K(\mathbf{x}', \mathbf{y}', \mathbf{x}, \mathbf{y}) - K(\mathbf{x}', \mathbf{y}', \mathbf{x}, \bar{\mathbf{y}})]. \tag{10}$$

Just like in the linear case, the same maximization problem (augmented with the loss $\Delta(\mathbf{y}, \mathbf{y}')$) serves as an oracle in Algorithm 1, line 4. In order to use combinatorial optimization, a kernel should be decomposable to factors. We use the radial basis function (RBF) kernel:

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}') = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{i'=1}^{N} \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}'_{i'}\|^2) y_i^k y_{i'}^k + \\ \sum_{k=1}^{K} \sum_{l=1}^{K} \sum_{(i,j)} \sum_{(i',j')} \exp(-\gamma \|\mathbf{x}_{ij} - \mathbf{x}'_{i'j'}\|^2) y_i^k y_j^l y_{i'}^k y_{j'}^l, \end{aligned} \tag{11}$$

where $(i, j) \in \mathcal{E}$ and $(i', j') \in \mathcal{E}'$ are the edges of the scans $\mathbf{x}$ and $\mathbf{x}'$ respectively, and γ is a parameter we set equal to 1.0.

Similar to the dot product in a Euclidean vector space, the kernel of similar elements in the transformed feature space is greater than for the distant ones. Intuitively, the maximum in (10) is reached for the assignment that is close to the way the training scan is labeled but far from the one for support vectors.

IV. Experiments

We evaluate our algorithm on airborne and terrestrial laser scans. Our experiments are aimed to show the advantage of non-associativity rather than to compete with state-of-the-art. We use the pointwise random forest classification of point features as the baseline and also show that cutting-plane approach with kernels[3] outperforms other non-linear methods for training max-margin Markov networks: functional gradient boosting for AMN learning [5][4] and naïve Bayes [6]. We run two series of experiments on the *Aerial* data set. For the first one we use constant unary potentials so that they do not affect the energy function. It is done to show the unique ability of our method to catch the information supplied by edge features. For the second series the unary potentials are assigned as logarithms of probabilistic random forest outputs, while pairwise are tuned via structured learning, which corresponds to the real-world usage of the algorithm. We also compare our method to the one with Hamming loss (without accounting imbalance) and to the linear structural SVM. We also apply our algorithm to more challenging *Road* data set and analyze its limitations.

A. Implementation details

We use the same features as in our prior work [6]. To train the random forest classifier we extract features of individual nodes using points in a vicinity of the point. We use a fixed-radius support region to compute the following features:

- spectral and directional features [4];
- variants of spin images;
- distribution of heights and related features [6].

For two neighboring points of the scan $\mathbf{p}$ and $\mathbf{q}$ with the approximated normals $\mathbf{n}_p$ and $\mathbf{n}_q$ the corresponding pairwise potentials' features are:

- cosine of the angle between approximated normals in the points: $\mathbf{n}_p^{\mathrm{T}}\mathbf{n}_q/(\|\mathbf{n}_p\|\|\mathbf{n}_q\|)$;
- difference in altitudes of the points $\mathbf{p}$ and $\mathbf{q}$ normalized by the distance between them: $(p_z - q_z)/\|\mathbf{p} - \mathbf{q}\|$.

We also use over-segmentation, which both speeds up the algorithm and makes edge features more informative. We use a variant of R-Tree for spatial indexing and for segmentation.[5] Points within each R-Tree leaf are combined to segments (the groups of about 50 neighboring points), for each segment the medoid is approximated as the scan point closest to the mean. For further processing an (approximate) medoid represents the whole group, i.e. the neighborhood graph is build over medoids, features are computed for medoids and links between them. Later the label assigned to a medoid is spread to the other points of the segment. Note that all the scan points participate in the support needed to compute features of individual medoids. It turns out that such over-segmentation is critical for non-associative Markov networks, which is discussed in Section V.

Since the original variant of functional gradient boosting does not handle imbalanced training data properly, we implemented a similar loss accounting scheme as described in the end of Section III-A. We run unexponentiated variant of the algorithm during $T = 100$ iterations with the decreasing step size $\alpha_t = 1/\sqrt{t}$. Following the authors, we estimate the regularization parameter on a validation set. We use the original naïve Bayes parameters as well: each distribution is approximated by a 10 bin histogram. The parameters of the structural SVM (C, ϵ) in our method were also estimated on the validation set.

B. Data sets

Aerial. We learn parameters of the algorithm on the airborne and test it on a similar one (Fig. 2a). The scans have been hand-labeled using three class labels: "ground", "building", "tree". Buildings are underrepresented (about 1/12, depending on the scan), the rest of the scan belongs to the ground and vegetation in approximately equal proportion. Each scan consists of about 100,000 points.

Road. Our terrestrial data contain 400,000 points scan for training and about 1 million for test (Fig. 1). Four classes are used: "ground", "vehicle", "tree", and "pole" (the latest includes both lamp and sign posts). There are only 0.2% of pole points, 5% of vehicles, 12% of vegetation, the rest is ground.

C. Results

The results of our first experiment are summarized in Table I. Since our data set is class-imbalanced, we report on precision and recall for each class individually rather than compute overall accuracy. We use the geometric mean of recalls (G-mean) as an overall performance measure, which treats all the classes equally [23]. Unsurprisingly, usage of meaningful unary potentials helps to achieve better performance, the improvement is more significant in the case of associative Markov networks trained with functional gradient boosting. This lag could be explained by the weakness of the associative pairwise potentials. Meanwhile, adding unary potentials does not make all the work. The results show that non-associative CRF yields better performance for both experiment settings. A visual example of the output is in Figure 2.

Our method is sparse: only 10 support vectors have been determined (although, each possible assignment could have become one). However, the kernel (11) contains the sum over all factors, so even handling single support vector is quite slow in comparison to the linear kernel, which is reduced to the set of weights. Unfortunately, the linear model behaves

[3] We use the implementation of cutting-plane optimization from the SVM$^{\mathrm{struct}}$ library: http://svmlight.joachims.org/.

[4] We use the implementation of inference and learning provided by the authors: http://www.cs.cmu.edu/~dmunoz/projects/m3n.html. We do not use higher-order cliques in this setting to compare only pairwise models.

[5] We use GML LidarK library: http://graphics.cs.msu.ru/en/science/research/3dpoint/lidark

Table I
PRECISIONS AND RECALLS FOR EACH OF THREE CLASSES AND THE G-MEAN RECALL FOR THE *Aerial* DATA SET. THE RESULTS OF RANDOM FOREST (UNARY), NAÏVE BAYES (BAYES), FUNCTIONAL GRADIENT BOOSTING (FUNC), AND OUR METHOD (SVM). -PW POSTFIX IS ADDED WHERE NO UNARY POTENTIALS WERE USED. THE RESULTS FOR DEGENERATIVE MODELS ARE IN THE LAST TWO ROWS: LINEAR STRUCTURAL SVM WITHOUT RBF KERNELS (SVM-LIN) AND KERNELIZED SVM WITH HAMMING LOSS (SVM-HAM).

Method	ground		building		tree		G-m
	pr	rec	pr	rec	pr	rec	rec
UNARY	0.992	0.952	0.576	0.688	0.890	0.892	0.836
BAYES-PW	0.985	0.979	0.493	0.698	0.898	0.809	0.821
FUNC-PW	0.911	0.975	0.578	0.545	0.923	0.850	0.767
SVM-PW	0.981	0.977	0.602	0.803	0.924	0.849	0.874
BAYES	0.983	0.978	0.496	0.779	0.917	0.789	0.844
FUNC	0.975	0.981	0.758	0.645	0.913	0.940	0.841
SVM	0.975	0.979	0.574	0.923	0.960	0.805	0.900
SVM-LIN	0.994	0.987	0.641	0.693	0.907	0.896	0.850
SVM-HAM	0.952	0.985	0.612	0.181	0.813	0.922	0.548

Table II
F-SCORES FOR THE RESULTS RETURNED BY SUBGRADIENT OPTIMIZATION (SUB, [4]) FUNCTIONAL GRADIENT BOOSTING (FUNC), AND OUR METHODS (SVM-LIN, SVM) ON THE *Road* DATA.

Method	ground	vehicle	tree	pole
SUB	0.974	0.302	0.497	0.138
FUNC	0.979	0.821	0.934	0.397
SVM-LIN	0.934	0.792	0.789	0.203
SVM	0.980	0.868	0.928	0.000

only slightly better than individual classification, when such features are used. For example, the neighboring ground regions have similar normals, and the cosine of the angle between them is close to zero, while big absolute values are unlikely for the "ground-ground" edges. Therefore, the linear model is unsuitable. The bottom row of Table I shows that usage of the weighted Hamming loss is crucial, in fact very few buildings are found with the vanilla Hamming loss.

We summarize results of the best performing methods on terrestrial data (kernel SVM and FGB, along with their linear counterparts) in Table II. Kernel structured SVM performs similarly to the functional gradient boosting on ground and tree classes and slightly better on vehicles. However, it fails to find poles, which are under-represented in the train set. The learned model has only 8 support vectors, which were enough to reach the requested precision $\epsilon = 10^4$. Decreasing ϵ might help to build a more flexible model, but it makes training intractable. So, our method performs poorly when a number of classes is big and/or some classes are extremely under-represented.

V. DISCUSSION

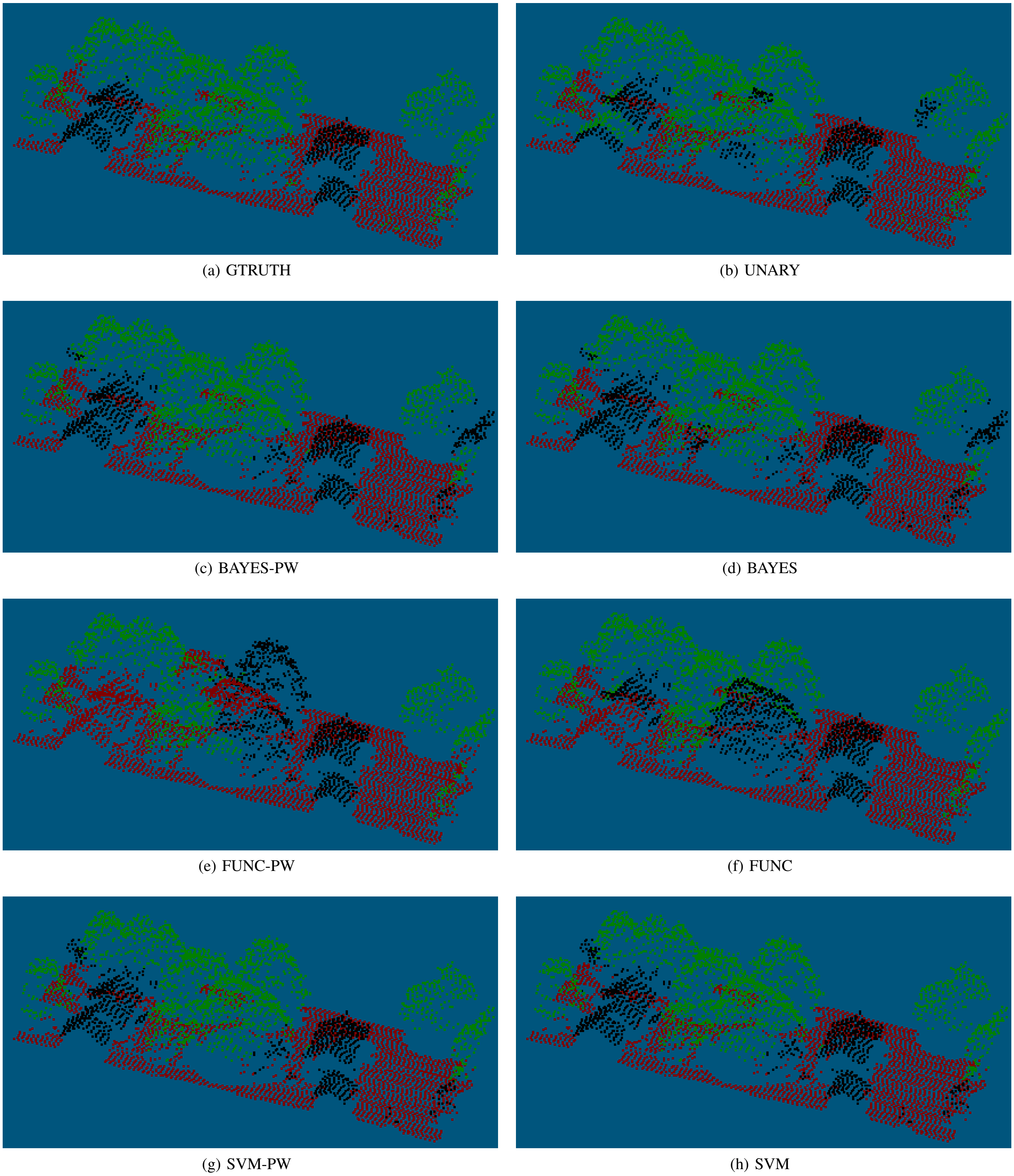
The experiment on *Aerial* data shows that non-associative Markov networks perform better than associative ones, especially when no meaningful unary potentials are available. Even simple naïve Bayes learning of pairwise potentials yields better results than powerful functional gradient boosting fettered with associativity. Motivation for using an associative model is usually small size of a training set hence the lack of statistics of the edges. We used a relatively small training scan (100,000 points), still collecting enough statistics. Our model turns out to be useful when the number of classes is low, and there are no very under-represented classes.

In our work edges connect quite distant points due to over-segmentation. Intuitively, heterogeneous edges occur more often because of that. Moreover, the features of such edges are more meaningful. For two neighboring points of a densely-scanned point cloud the features like the orientation and length of the segment that connects the points are useless. For instance, Anguelov et al. do not use any form of subsampling. They report that the constant pairwise feature performs better than meaningful ones, although they use meager associative Markov networks [2].

We use the TRW-S algorithm for inference as well as for separation oracle [17]. In spite of the oracle algorithm is approximate, the results are good in practice. Approximate inference makes our training algorithm *undergenerative*, which means on each step it finds the most violated constraint among the reduced set, and the constraint is thus feasible, but not necessarily is the most violated one [8]. The *overgenerative* alternative is to enhance the feasible region to make it possible to find the exact maximum efficiently. It could be done via a linear programming relaxation. If an LP relaxed solution is integral, it is also the optimum of the original problem, otherwise some variables could be assigned with fractional values. If constraints based on fractional assignments are allowed in the cutting-plane procedure, the theoretical properties are preserved [8]. However, if the approximation is tight in undergenerative approach (it is in our experiments), the resulting solution of (4) is guaranteed to be not far from optimal. Moreover, the residual can be estimated given only the precision for the last-chosen violated constraint. It can be upper bounded by the duality gap in TRW-S, which is often null on the latest iterations in our experiments. Thus, approximate inference is not a problem of method's accuracy, although it can slow it down.

VI. FUTURE WORK

There is number of ways to improve our method. First, it is possible to add new scan features. For example, a lot of scans contain color information nowadays, so joint color and shape descriptors are going to improve the performance. Second, there is a trend to use higher-order cliques for segmentation [5]. It is alluring to use non-associative potentials of higher-order cliques, but is unclear how to optimize energy of that kind. Finally, training can be sped up by exploiting the structure of the prediction task. For example, the DLPW method (dual loss, primal weights) proposed recently by Meshi et al. seems to be able to help with efficiency since it does not require solving the whole maximization problem on each iteration [24].



(a) GTRUTH

(b) UNARY

(c) BAYES-PW

(d) BAYES

(e) FUNC-PW

(f) FUNC

(g) SVM-PW

(h) SVM

Figure 2. Results on a part of the test scan where our method performs well. Red color denotes ground, black — building, green — vegetation. (a) Ground truth labeling. (b) Random Forest, or just unary potentials. (c)–(d) Naïve Bayes, without and with pairwise potentials. (e)–(f) Functional gradient boosting. (g)–(h) Cutting-plane training. *Better viewed in colour and magnified. The whole scan along with the results could be found in the supplementary material.*

Acknowledgements

We'd like to thank Dmitry Vetrov for proofreading and advices. The work has been partially supported by the federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013", project P1189.

References

[1] S. K. Lodha, D. M. Fitzpatrick, and D. P. Helmbold, "Aerial Lidar Data Classification using AdaBoost," in *IEEE 3DIM*, 2007, pp. 435–442. [Online]. Available: http://users.soe.ucsc.edu/~dph/mypubs/final3dim2007.pdf 1

[2] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data," in *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 169–176. [Online]. Available: http://ai.stanford.edu/~vasco/pubs/cvpr05.pdf 1, 3, 6

[3] R. Triebel, R. Shmidt, O. Mozos, and W. Burgard, "Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data," in *IJCAI*, Hyderabad, India, 2007, pp. 2225–2230. [Online]. Available: http://www.informatik.uni-freiburg.de/~omartine/publications/triebel2007ijcai.pdf 1, 2, 3

[4] D. Munoz, N. Vandapel, and M. Hebert, "Directional associative markov network for 3-d point cloud classification," in *3DPVT*, Atlanta, GA, 2008. [Online]. Available: http://www.cc.gatech.edu/conferences/3DPVT08/Program/Papers/paper200.pdf 1, 3, 5, 6

[5] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional Max-Margin Markov Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, Jun. 2009, pp. 975–982. [Online]. Available: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1039&context=robotics 1, 3, 5, 6

[6] R. Shapovalov, A. Velizhev, and O. Barinova, "Non-associative Markov networks for 3D point cloud classification," in *Photogrammetric Computer Vision and Image Analysis*, Paris, France, 2010. [Online]. Available: http://shapovalov.ro/papers/Shapovalov-et-al-PCV2010.pdf 1, 5

[7] I. Posner, M. Cummins, and P. Newman, "A generative framework for fast urban labeling using spatial and temporal context," *Autonomous Robots*, vol. 26, no. 2-3, pp. 153–170, Mar. 2009. [Online]. Available: http://www.robots.ox.ac.uk:5000/~mjc/Papers/AutonomousRobots_HIP_MJC_PNM_2009.pdf 2

[8] T. Finley and T. Joachims, "Training Structural SVMs when Exact Inference is Intractable," in *International Conference on Machine Learning*, New York, NY, 2008, pp. 304–311. [Online]. Available: http://www.joachims.org/publications/finley_joachims_08a.pdf 2, 4, 6

[9] V. Franc and B. Savchynskyy, "Discriminative learning of max-sum classifiers," *JMLR*, vol. 9, pp. 67–104, 2008. [Online]. Available: http://jmlr.csail.mit.edu/papers/volume9/franc08a/franc08a.pdf 2, 4

[10] T. Joachims, T. Finley, and C. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009. [Online]. Available: http://tfinley.net/research/joachims_etal_09a.pdf 2, 3, 4

[11] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based Recognition of 3D Point Clouds in Urban Environments," in *IEEE International Conference on Computer Vision*, Kyoto, Japan, 2009. [Online]. Available: http://www.cs.princeton.edu/gfx/pubs/Golovinskiy_2009_SRO/paper.pdf 2

[12] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha, "Scene Understanding in a Large Dynamic Environment through a Laser-based Sensing," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 127–133. [Online]. Available: http://www.poss.pku.edu.cn/Data/publications/icra10.pdf 2

[13] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-D Scene Analysis via Sequenced Predictions over Points and Regions," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011. [Online]. Available: http://www.cs.princeton.edu/courses/archive/spring11/cos598A/pdfs/Xiong11.pdf 2

[14] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation," in *NIPS*, Vancouver, 2001, pp. 689–695. [Online]. Available: http://eprints.kfupm.edu.sa/42528 2

[15] T. Werner, "A Linear Programming Approach to Max-sum Problem: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, 2007. [Online]. Available: http://cmp.felk.cvut.cz/ftp/articles/werner/Werner-PAMI-2007.pdf 2

[16] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001. [Online]. Available: http://www.csd.uwo.ca/~yuri/Papers/pami01.pdf 2

[17] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006. [Online]. Available: http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/TRW-S-PAMI.pdf 2, 6

[18] R. B. Rusu, A. Holzbach, N. Blodow, and M. Beetz, "Fast Geometric Point Labeling using Conditional Random Fields," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009. [Online]. Available: http://files.rbrusu.com/publications/Rusu09IROS_FPFH.pdf 3

[19] E. H. Lim and D. Suter, "3D terrestrial LIDAR classifications with super-voxels and multi-scale Conditional Random Fields," *Computer-Aided Design*, vol. 41, no. 10, pp. 701–710, Oct. 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0010448509000475 3

[20] B. Taskar, V. Chatalbashev, and D. Koller, "Learning associative Markov networks," in *ICML*, Banff, Alberta, Canada, 2004, pp. 102–109. [Online]. Available: http://www.seas.upenn.edu/~taskar/pubs/mmamn.pdf 3

[21] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "(Online) Subgradient Methods for Structured Prediction," in *International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007. [Online]. Available: http://www.ri.cmu.edu/pub_files/pub4/ratliff_nathan_2007_3/ratliff%_nathan_2007_3.pdf 3

[22] M. Szummer, P. Kohli, and D. Hoiem, "Learning CRFs using graph cuts," in *European Conference on Computer Vision*. Marseille, France: Springer, 2008, pp. 582–595. [Online]. Available: http://research.microsoft.com/en-us/um/people/pkohli/papers/skh_eccv08.pdf 4

[23] Y. Sun, M. S. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalanced class distribution," in *IEEE International Conference on Data Mining*, 2006, pp. 592–602. [Online]. Available: http://people.ee.duke.edu/~lcarin/ImbalancedClassDistribution.pdf 5

[24] O. Meshi, D. Sontag, T. Jaakkola, and A. Globerson, "Learning Efficiently with Approximate Inference via Dual Losses," in *ICML*, 2010. [Online]. Available: http://people.csail.mit.edu/dsontag/papers/MesSonJaaGlo_icml10.pdf 6