Max-Margin Markov Networks and Applications in 2D and 3D Image Segmentation

Agis Mesolongitis

May 17, 2010

Contents

Chapter 1

Introduction

This report deals with an approach to discriminative classification that uses both max-margin learning and a Markov Network framework and is referred to as Max-Margin Markov Networks (M^3Ns). To be more specific, this technique uses a Discriminative Model which is the posterior probability function of the Markov Network. We need to learn this probability function and then find the set of parameters that maximize this function (MAP estimation). In practical applications, the M^3Ns framework is used to give labels to a set of connected nodes, given a number of *features*, or *observations*. These features can be assigned to a node or a clique. So, we would like to map the set of all features to the set of all labels by maximizing the probability: $P(y|x;q)$, where y is the set of labels, x is the set of features and w is the set of the parameters.

In order to find the parameters, we learn the model using a technique borrowed from the Support Vector Machines learning procedure. We maximize the margin between the probability of the correct labels given the features and all the incorrect labels given the features. In mathematical notation, this margin can be defined as the lower bound of: $P(\hat{y}|x;w) - P(y|x;w), \forall y \neq \hat{y}$, where $\hat{y}$ is the correct label assignment. An already labeled training set is needed for the learning. Then, inference can be done using any way described in the literature.

One big advantage of this technique is that Max-Margin learning leads to an optimization function that contains the dot product of the features. Consequently, Kernel Methods can be used to generate more features and improve accuracy. Furthermore, it can be proven that MM learning results to a specific generalization bound. This bound defines a maximum value of a per-label loss function. The fact that this framework uses Markov Networks, makes the classifier more robust, as the interactions between data points are exploited efficiently.

For the problem of labeling, we consider the y labels to be discrete, having K possible values. So, 1-of-K representation will be used inside throughout this text. The next chapter is about general information about the structure and the details of Markov Networks used with the M^3Ns framework. Then, the procedure of Max-Margin Learning is presented and finally there are some applications in the field of 3D and 2D image labeling.

Chapter 2

Background Information: Markov Networks and Special Cases

2.1 Markov Networks

Markov Networks formulation is a way of representing the joint probability of a number of dependent random variables. This network is undirected. Also, in this paper If we define cliques inside this network and assign a potential function to each clique, the joint probability function is given:

$$P(\mathbf{y}) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{y}_c)$$

where:

- $\mathbf{y}$ are all the node variables
- $\mathbf{y}_c$ are the clique node variables
- $Z = \sum_{\mathbf{y}} \prod_{c \in C} \psi_c(\mathbf{y}_c)$

Pairwise Case A special case of the Markov Networks is the Pairwise Markov Network. In this case, cliques are defined only over nodes and edges. This simplified version will be used throughout the report to demonstrate the examined techniques. The joint probability is given by:

$$P(\mathbf{y}) = \frac{1}{Z} \prod_{i} \psi_n(y_i) \prod_{i,j} \psi_e(y_i, y_j)$$

Then, all these techniques can also apply for similar networks but with larger interactions instead of the edge interactions. The generalization is straightforward. This time, the joint probability is:

$$P(\mathbf{y}) = \frac{1}{Z} \prod_{i} \psi_n(y_i) \prod_{c} \psi_c(\mathbf{y_c})$$

2.2 Conditional Random Fields

In accordance with the simple Markov Networks, if we would like to express the conditional probability of a full network, given an event X, then we can have a similar formulation, augmented with the new variable. For the pairwise case, we have:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_x} \prod_i \psi_n(y_i, \mathbf{x}) \prod_{i,j} \psi_e(y_i, y_j, \mathbf{x})$$

where $\mathbf{x}$ is a set of features, or observations. We can assume that these features affect only a node or an edge label. Thus, we have the following more specific model:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_x} \prod_i \psi_n(y_i, \mathbf{x}_i) \prod_{i,j} \psi_e(y_i, y_j, \mathbf{x}_{ij})$$

2.3 Log-Linear Markov Models

2.3.1 General Case

The potential functions mentioned before can be of any arbitrary type. However, in the literature, a specific type of functions is preferred in many cases. This type is the log-linear functions. For these functions, we have:

$$\log(\psi(\mathbf{y_c}, \mathbf{x})) = \mathbf{w}^T \mathbf{f}(\mathbf{y_c}, \mathbf{x})$$

It is obvious that any product of these functions will result to a log-linear function and therefore, the conditional probability function will be log linear. This is a convenient property that eases maximization problems.

2.3.2 Simplified Case

A simple but efficient type of potential functions is the **log-linear combination of the features**. We will describe this potential function for edge and node cliques.

Let $\mathbf{y}_i$ be the 1-of-K representation of the label of a node and y_i^k the k-th element of the vector:

$$\mathbf{y}_i = [000...010...000]$$
$$y_i^k \in \{0, 1\}$$

The log-linear combination functions are given by:

$$\log \psi_i(k) = \mathbf{w}_n^k \mathbf{x}_i$$
$$\log \psi_{i,j}(k, l) = \mathbf{w}_e^{k,l} \mathbf{x}_{i,j}$$

We can see that the effect of the $\mathbf{y}$ on the function is in the selection of w. We have a w vector for each type of clique $(\mathbf{w}_n, \mathbf{w}_e)$, and for each label or combination of labels $(\mathbf{w}^i, \mathbf{w}^{i,j})$. Obviously, the length of the vectors is equal to

the feature vectors. To get the big picture, we can combine all these potentials and write down the log of the probability function:

$$\begin{aligned} \log P(\mathbf{y}|\mathbf{x};\mathbf{w}) &= \sum_i \sum_k (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k \\ &+ \sum_{i,j\in E} \sum_k (\mathbf{w}_e^{k,l} \cdot \mathbf{x}_{i,j}) y_i^k y_j^l \\ &- logZ_w(\mathbf{x}) \end{aligned} \tag{2.1}$$

where: i are the nodes, (i,j) the edges, k,l the labels.

2.4 Associative Markov Networks

This special case, proposed by Taskar et al. in [2] is a significant subset of the Markov Networks used widely in clustering and 2D-3D image segmentation problems. The basic idea is that the potential functions capture the essence of "guilt by association", where connected nodes are forced to have similar labels by means of a bigger probability. They are similar to the log-linear combination models, but **the edge potentials are equal to 1 if the edge labels are not the same**:

$$\log \psi_i(k) = \mathbf{w}_n^k \mathbf{x}_i$$

$$\log \psi_{i,j}(k,l) = \begin{cases} \mathbf{w}_e^{k,l} \mathbf{x}_{i,j}, \text{ if } k = l \\ 0, \text{ if } k \neq l \end{cases} \tag{2.2}$$

$$\begin{aligned} \log P(\mathbf{y}|\mathbf{x};\mathbf{w}) &= \sum_i \sum_k (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k \\ &+ \sum_{i,j\in E} \sum_k (\mathbf{w}_e^k \cdot \mathbf{x}_{i,j}) y_{i,j}^k \\ &- logZ_w(\mathbf{x}) \end{aligned} \tag{2.3}$$

where $y_{i,j}^k = 1$ when $y_i^k = 1 \wedge y_j^k = 1$. An extension of this model for larger cliques, would result to:

$$\begin{aligned} \log P(\mathbf{y}|\mathbf{x};\mathbf{w}) &= \sum_i \sum_k (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k \\ &+ \sum_{c\in C} \sum_k (\mathbf{w}_e^k \cdot \mathbf{x}_c) y_c^k \\ &- logZ_w(\mathbf{x}) \end{aligned} \tag{2.4}$$

Simplified Notation For the log-linear combination case, and consequently for the AMNs case, we can simplify the notation of the conditional log likelihood function. We define:

- $\mathbf{w} = (\mathbf{w}_n, \mathbf{w}_e)$, where:

  - $\mathbf{w}_n = (\mathbf{w}_n^1, ..., \mathbf{w}_n^K)$
  - $\mathbf{w}_e = (\mathbf{w}_e^1, ..., \mathbf{w}_n^K)$
- $\mathbf{y} = (\mathbf{y}_n, \mathbf{y}_e)^T$, where:
  - $\mathbf{y}_n = (..., y_i^1, ..., y_i^K, ...)^T$
  - $\mathbf{y}_e = (..., y_{i,j}^1, ..., y_{i,j}^K, ...)^T$

So, we have gathered all the weights and all the binary label assignments in long vectors. If we define an appropriate matrix $\mathbf{X}$, the equation (2.3) becomes:

$$\log P(\mathbf{y}|\mathbf{x}; \mathbf{w}) = \mathbf{w}\mathbf{X}\mathbf{y} - logZ_w(\mathbf{x}) \tag{2.5}$$

MAP solution Supposing we have already trained our model, the next step is to find the optimal solution for $\mathbf{y}$. This is an Integer Optimization Problem, as $\mathbf{y}$ has integer values. A popular method for minimizing this type of energy functions is Min-Cut inference as mentioned in [5]. Also, in [2], there is another approach using a relaxed LP instead, that is followed by rounding of the variables. In the next chapter, a way of learning the weights of AMNs is described.

Chapter 3

Max Margin Learning for Markov Networks

The task Max-Margin learning has to accomplish is the optimal estimation of the parameter vector $\mathbf{w}$. The next sections describe this technique.

3.1 Support Vector Machines: Max-Margin Learning

To introduce the idea of Max-Margin Learning, we demonstrate the case of SVMs. Here, our goal is to define discriminant functions for binary classification. So, we have to define hyperplanes that linearly separate the data. The max-margin rule says that this hyperplane should be placed in a position so that the distance between the closest data points from either sides is maximum.

So, we define two more hyperplanes, each one of which includes these support points (support vectors, see figure 3.1). We can formulate the equation of the decision hyperplane as:

$$\mathbf{w}^T\mathbf{x} + b = 0$$

And the equations of the two support vector hyperplanes as:

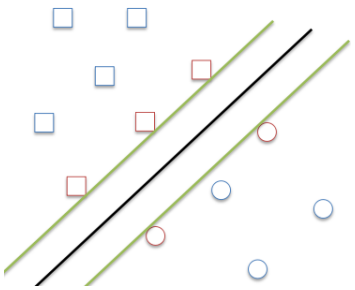

Figure 3.1: SVMs idea

$$\begin{aligned} \mathbf{w}^T\mathbf{x} + b &= 1 \\ \mathbf{w}^T\mathbf{x} + b &= -1 \end{aligned}$$

It can be easily proven that the distance between the two support hyperplanes is:

$$\gamma = \frac{2}{\|\mathbf{w}\|}$$

We would like to maximize the margin γ. So, we have the equivalent problem:

$$\text{minimize: } \|\mathbf{w}\|$$
$$\text{subject to: } t_i(\mathbf{w}^T\mathbf{x} + b) > 1$$

where t_i is the indicator of the cluster of the point. It is $t_i = 1$ for cluster 0 or $t_i = -1$ for cluster 1. Finally, using the dual formulation, we can calculate the $\mathbf{w}$.

3.2 Markov Networks: Max-Margin Learning

Similarly to the SVMs problem, in order to find the optimal parameter vector for log-linear Markov Networks, we can set up a Max-Margin optimization problem. Here, we don't have a discriminant function to learn. But we have a discriminative model, that is the probability function $P(\mathbf{y}|\mathbf{x}; \mathbf{w})$. Our goal is to maximize the margin between the probability of the correct assignment of labels $\hat{\mathbf{y}}$ and the probability of each incorrect label assignment.

3.2.1 MM Framework

To construct this problem, some quantities have to be defined first:

- The log-probability function: $\mathbf{w}^T\mathbf{f}(\mathbf{x}, \mathbf{y}) - logZ_w(\mathbf{x})$
- The difference between the kernel function of a correct and an incorrect assignment: $\Delta\mathbf{f}_x(\mathbf{y}) = \mathbf{f}(\mathbf{x}, \hat{\mathbf{y}}) - \mathbf{f}(\mathbf{x}, \mathbf{y})$
- The number of labeling errors for each incorrect assignment: $\Delta(\mathbf{y}, \hat{\mathbf{y}})$
- The margin variable: $\gamma > 0$

The optimization problem is defined in [1] and is the following:

$$\text{maximize: } \gamma$$
$$\text{s.t. } \begin{cases} \|\mathbf{w}\| \leq 1 \\ \mathbf{w}^T\Delta\mathbf{f}_x(\mathbf{y}) \geq \gamma\Delta(\mathbf{y}, \hat{\mathbf{y}}), \forall\mathbf{y} \end{cases} \tag{3.1}$$

We include the number of wrong labels in order to have a larger margin for less correct assignments, making the system more robust. Similarly to the SVMs

problem, we can absorb the γ factor. Now we have an equivalent Quadratic Problem:

$$\text{minimize: } \frac{1}{2}\|\mathbf{w}\|^2 \text{ s.t. } \mathbf{w}^T \Delta \mathbf{f}_x(\mathbf{y}) \geq \Delta(\mathbf{y}, \hat{\mathbf{y}}), \forall \mathbf{y} \tag{3.2}$$

An important thing of this problem is the number of constraints. If the length of $\mathbf{y}$ is L and the number of possible label values is K, the total number of constraints is K^L. It is exponential with respect to the number of the labels. Consequently, the number of Lagrange multipliers in the dual problem will be exponential as well.

3.2.2 Reduction of Constraints (outline)

In [1] it is mentioned that if we use a Markov Chain only with Edge cliques (no nodes in the model), we can apply a transformation to these Lagrange Multipliers (α) and get new variables (μ). These variables correspond to edges and nodes ($\mu_i, \mu_{i,j}$). If the cliques of the Markov Net form a forest, then the authors prove that we can go to the dual problem and replace the exponential number of constraints with a set of only three Polynomial constraints. These constraints demand that the μ variables represent expected values arisen from a valid probability density function and they are presented here:

$$\begin{aligned} \sum_k \mu_{i,j}(k,l) &= \mu_j(l) \\ \sum_k \mu_i(k) &= C \\ \mu_{i,j}(k,l) &\geq 0 \end{aligned} \tag{3.3}$$

After solving the dual problem, with these new constraints, we can then trivially calculate the optimal parameters $\mathbf{w}$. Now, the MAP problem can be solved using any inference algorithm available in the literature. This framework was experimentally used for Optical Character Recognition in [1] and had superior performance compared to older methods.

3.3 AMNs: Max-Margin Learning

In the case of the AMNs, the problem is more simple because instead of a dot product of $\mathbf{w}$ and functions of the features, we have just a dot product of $\mathbf{w}$ and the features themselves.

3.3.1 MM Framework

The quantities of the Max-Margin Problem are going to be simplified:

- The log-probability function: $\mathbf{wXy} - logZ_w(\mathbf{x})$
- The difference between correct and incorrect: $\mathbf{wX}(\hat{\mathbf{y}} - \mathbf{y})$
- The number of labeling errors for each incorrect assignment: $\Delta(\mathbf{y}, \hat{\mathbf{y}}) = N - \mathbf{y}_n^T \mathbf{y}_n$, where N the number of nodes-labels.

Now, by substituting and introducing slack variables for non-separable data, the minimization problem (3.2) becomes:

$$\begin{aligned} \text{minimize: } & \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \\ \text{s.t. } & \mathbf{wX}(\hat{\mathbf{y}} - \mathbf{y}) \geq N - \mathbf{y}_n^T\mathbf{y}_n, \forall \mathbf{y} - \xi \end{aligned} \tag{3.4}$$

3.3.2 Reduction of Constraints (outline)

Again, we have a Quadratic Program with exponential number of constraints. In [2], a solution adapted to the AMNs problem is proposed instead of the general solution mentioned in the previous section. The above problem is equivalent to:

$$\begin{aligned} \text{minimize: } & \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \\ \text{s.t. } & \mathbf{wX}\hat{\mathbf{y}} - N + \xi \geq \max_{y \in \Upsilon}\{\mathbf{wXy} - \mathbf{y}_n^T\mathbf{y}_n\} \end{aligned} \tag{3.5}$$

where Υ is the set of binary vectors created by the 1-of-K representation for $\mathbf{y}$ mentioned before. We can see that now, we have only one non linear constraint. This constraint contains the MAP problem:

$$\max_{y \in \Upsilon}\{\mathbf{wXy}\}$$

This is an integer problem. The solution given by the authors of [2] is to replace this integer problem with a Linear Problem whose results will be rounded in the end. Optimization theory allows us to convert this LP to its dual and finally we can obtain Polynomial constraints for the initial problem. The conversion of the Integer Program to a LP does not guarantee optimality of the solution, but is claimed to work well in practice.

The idea of this procedure is the following: If we have the following LP:

$$\max_{\mathbf{y}} \quad \mathbf{wBy} \quad \text{s.t.} \quad \mathbf{y} \geq 0, \quad \mathbf{Ay} \leq \mathbf{b}$$

Then, the dual would be:

$$\min_{\mathbf{z}} \quad \mathbf{b}^\top\mathbf{z} \quad \text{s.t. } \mathbf{z} \geq 0, \mathbf{A}^\top\mathbf{z} \geq (\mathbf{wB})^\top$$

So, in [2] it is suggested that the Max-Margin Problem should be:

$$\begin{aligned} \min \quad & \frac{1}{2}||\mathbf{w}||^2 + C\xi \\ \text{s.t.} \quad & \mathbf{wX}\hat{\mathbf{y}} - N + \xi \geq \mathbf{b}^\top\mathbf{z}; \\ & \mathbf{z} \geq 0, \quad \mathbf{A}^\top\mathbf{z} \geq (\mathbf{wB})^\top \end{aligned}$$

Use of AMNs Max-Margin Learning for AMNs is a popular framework for segmentation in 2D and 3D images used widely in the literature. Later works concentrate in finding more efficient and accurate ways of minimizing the Max-Margin functions. For example, in [4] the authors use the AMN framework but they calculate the Max-Margin weights in a completely different way, using Functional Gradient Methods, a technique applied in Convex Optimization Problems. They also use Boosting, in order to have more accurate results.

3.4 Comments on M^3Ns

Kernel Expansion An important advantage of this Max-Margin framework described in this chapter, is that the dual formulations of the problems contain dot products of the node feature vectors (not for the higher cliques). This allows the use of Kernels ([2]) in order to create a larger feature space and make the system more accurate.

Generalization Bound In [1] there is a formal proof that the average per-label loss for new data sets (generalization error) is bounded and dependent on the margin γ of the classifier trained with a training data set. Therefore, we have a priori knowledge of how well can this classifier behave in the presence of new data.

Chapter 4

Examples of AMNs in Segmentation

An effective approach for segmenting 2D and 3D images in contextual clusters is the AMN framework. This technique leads to semantic clusters, where each point is assigned an class label such as "vertical", "ground", "tree" etc. This can be done to both 3D and 2D images as well, if the appropriate features are used. **AMNs are useful in this labeling procedure because they capture the interactions between neighboring points**. The edge or high-order cliques are forced to have the same label during training, so that inside points of an object result to higher probability values in opposition to the boundary points.

4.1 Segmentation of 3D Point Clouds

In [3] we can see an example of the use of AMNs in 3D point segmentation. We have a number of 3D points and the problem is to give each point a label according the object class it belongs to. A Pairwise Markov Network is constructed using closest points and node features were extracted by gathering various statistics of a neighborhood of each point, such as height indicators, amount of points lying in different areas of the neighborhood's principal plane, etc. It is impressive that the segmentation worked with constant features in the place of the edge features.

Manually labeled 3D point set were used to train the model using the AMN Max-Margin Problem with replaced polynomial constraints. The MAP problem was solved with min-cuts methods as in [5]. Figure 4.1 demonstrates a sample segmentation of a 3D image divided in "buildings", "trees" and "shrubbery".

In [4], functional gradient methods together with boosting are used to learn the parameters of the model, resulting in more accurate labeling for a much bigger number of classes. Here, objective function and constraints are combined in a functional gradient formulation. The reader can refer to Convex Optimization Theory and to [4] for details. The features extracted from the neighborhood of each point are more sophisticated including spectral features describing linearity planarity, scatter, there are direction indicators etc. Also, instead of nodes and edges, high order cliques are used too.

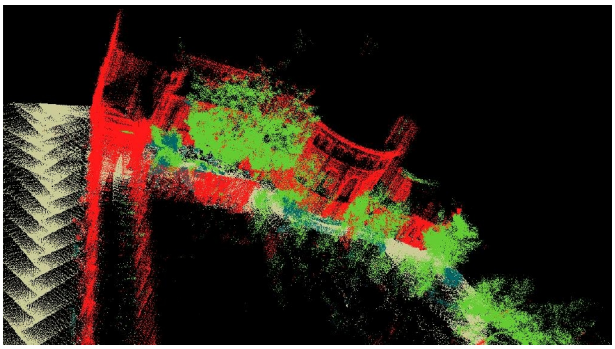

Figure 4.1: Labeling of 3D points as "Ground", "Tree", "Building", "Shrubbery"

Figure 4.2: Labeling of pixels as "Ground", "Vertical", "Sky"

4.2 Geometrical Segmentation of 2D Images

An interesting result of [4] is that the same technique can be used to classify pixels of 2D images in contextual geometric classes, thus extracting 3D geometry from a 2D image. This means that the classes can have names such as: "street", "sky", "house". This kind of categorization differs from the usual image segmentation approaches because it deals with contextual and geometric labels. In this work, the nodes are represented not by pixels, but by *superpixels*. These entities are groups of pixels with similar color values and filtered color values. Also, similar superpixels were considered as high order cliques.

A key factor for the success of this approach is the feature vector, that has to include geometrical features that capture location and perspective as well as generic features capturing shape and color. In figure 4.2 we can see the result of a segmentation of a 2D image into: "ground", "vertical" and "sky". An intuition about the sky segment is that the "sky" superpixels of the training set contributed to the model so that, roughly, the probability $P(y = "sky"|x_{color} = "blue")$ is high. So, this model will be biased towards "sky" labels if it is given blue superpixels.

Bibliography

[1] B. Taskar, C. Guestrin and D. Koller. Max-Margin Markov Networks. Neural Information Processing Systems Conference (NIPS03), Vancouver, Canada, December 2003.

[2] B. Taskar, V. Chatalbashev and D. Koller. Learning Associative Markov Networks. Twenty First International Conference on Machine Learning (ICML04), Banff, Canada, July 2004.

[3] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, A. Ng. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. International Conference on Computer Vision and Pattern Recognition (CVPR05), San Diego, CA, June 2005.

[4] D. Munoz, J. A. Bagnell, N. Vandapel, M. Hebert, Contextual Classification with Functional Max-Margin Markov Networks , IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[5] Yuri Boykov, Olga Veksler, Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. In IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 23, no. 11, pp. 1222-1239, 2001.