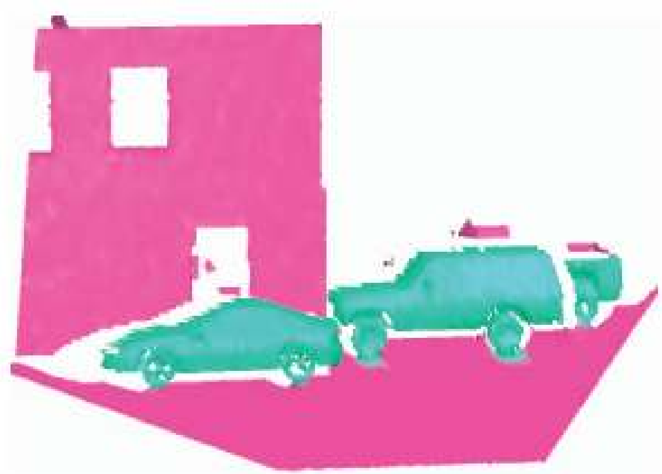
# Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data

By: D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng

Presenter: Yunpeng Li

# Introduction

- Segmentation of 3D Scan
  - Assign object category labels to scan points, e.g. this point is scanned from building/tree/people etc.



# Graphical Model

- Markov random fields (MRF = Markov network)
  - Analogous to HMM, but is undirected and allows higher connectivity and loops
  - Generative model
- Max-margin estimation
  - Discriminative learning

# Associative Markov Networks (AMN)

- Pairwise model, defined by vertex and edge potentials $\phi_i(y_i)$ , $\phi_{ij}(y_i, y_j)$
  - By Hammersley-Clifford Theorem, MRF can be factored:

$$P_\phi(\mathbf{y}) = \frac{1}{Z} \prod_{i=1}^{N} \phi_i(y_i) \prod_{ij \in \mathcal{E}} \phi_{ij}(y_i, y_j)$$

  - Additional restriction for AMN:

$$\phi_{ij}(k,k) = \lambda_{ij}^k, \text{ where } \lambda_{ij}^k \geq 1, \text{ and } \phi_{ij}(k,l) = 1, \ \forall k \neq l$$

    - Intuitively: reward continuity instead of penalized discontinuity

# Log-linear Parameters

- Potentials are formulated in terms of node and edge features $x_i$ and $x_{ij}$.
- The logarithm of node and edge potentials are expressed as weighted feature sums.

$$\begin{aligned}\log \phi_i(k) &= \mathbf{w}_n^k \cdot \mathbf{x}_i \\ \log \phi_{ij}(k,k) &= \mathbf{w}_e^k \cdot \mathbf{x}_{ij}\end{aligned}$$

- $\mathbf{w}_n^k$ and $\mathbf{w}_e^k$ are the parameters to be determined.
- AMN requires that $\mathbf{w}_e^k \cdot \mathbf{x}_{ij} \geq 0$, which is satisfied by constraining $\mathbf{x}_{ij} \geq 0$ and $\mathbf{w}_e^k \geq 0$.

# Optimization of AMN

- Can be exactly solved for binary labels (K = 2) using min-cut.
- NP-hard for K > 2, but can be approximated using *alpha-expansion* (Boykov, Veksler & Zabih) within a factor of 2.
  - AMN guarantees $-\log \phi_{ij}(k, k)$ is *regular*.
- Other optimization methods:
  - Loopy belief propagation (LBP)
  - Tree re-weighted message passing (TRW)
  - Linear program (LP) relaxation

# Integer Program Formulation

- Represent an assignment $\mathbf{y}$ as a set of *K*N* indicators $\{y_i^k\}$, where $y_i^k = I(y_i = k)$.
- Thus the log of conditional probability $\log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x})$ is:

$$\sum_{i=1}^{N} \sum_{k=1}^{K} (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k + \sum_{ij \in \mathcal{E}} \sum_{k=1}^{K} (\mathbf{w}_e^k \cdot \mathbf{x}_{ij}) y_i^k y_j^k - \log Z_{\mathbf{w}}(\mathbf{x}).$$

- In compact notation (see page 4 for abbreviation details):

$$\log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) = \mathbf{w}\mathbf{X}\mathbf{y} - \log Z_{\mathbf{w}}(\mathbf{x})$$

  – $\mathbf{w}$, $\mathbf{y}$ are concatenated weight, assignment vectors respectively.
  – $\mathbf{X}$ contains node and edge feature vectors with padded zeros.

# LP Relaxation of the MAP Problem

$$\max \sum_{i=1}^{N}\sum_{k=1}^{K}(\mathbf{w}_n^k \cdot \mathbf{x}_i)y_i^k + \sum_{ij\in\mathcal{E}}\sum_{k=1}^{K}(\mathbf{w}_e^k \cdot \mathbf{x}_{ij})y_{ij}^k$$

$$\text{s.t.} \quad y_i^k \geq 0, \quad \forall i,k; \qquad \sum_k y_i^k = 1, \quad \forall i;$$

$$y_{ij}^k \leq y_i^k, \qquad y_{ij}^k \leq y_j^k, \quad \forall ij \in \mathcal{E}, k.$$

- Quadratic term $y_i^k y_j^k$ replaced by variable $y_{ij}^k$ .
  - Bound tight at optimal, hence $y_{ij}^k = \min(y_i^k, y_j^k)$ .
  - Therefore $y_{ij}^k = y_i^k y_j^k$ if $y_i^k, y_j^k \in \{0,1\}$

# Maximum Margin Estimation

- The gain of true labeling $\hat{\mathbf{y}}$ over another labeling $\mathbf{y}$ is:

$$\log P_{\mathbf{w}}(\hat{\mathbf{y}} \mid \mathbf{x}) - \log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) = \mathbf{w}\mathbf{X}(\hat{\mathbf{y}} - \mathbf{y}).$$

- Hence the max margin formulation is:

$$\max \quad \gamma \quad \text{s.t.} \quad \mathbf{w}\mathbf{X}(\hat{\mathbf{y}} - \mathbf{y}) \geq \gamma \ell(\hat{\mathbf{y}}, \mathbf{y}); \quad ||\mathbf{w}||^2 \leq 1.$$

- The uniform per-label loss function

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = N - \hat{\mathbf{y}}_n^\top \mathbf{y}_n$$

- Therefore have quadratic program (QP):

$$\begin{aligned} \min \quad & \frac{1}{2}||\mathbf{w}||^2 + C\xi \\ \text{s.t.} \quad & \mathbf{w}\mathbf{X}(\hat{\mathbf{y}} - \mathbf{y}) \geq N - \hat{\mathbf{y}}_n^\top \mathbf{y}_n - \xi, \;\; \forall \mathbf{y} \in \mathcal{Y}. \end{aligned}$$

  – Problem: exponentially many constraint

# Maximum Margin Estimation (cont.)

- Replace exponential-size set of linear constraint

$$\mathbf{w}\mathbf{X}(\hat{\mathbf{y}} - \mathbf{y}) \geq N - \hat{\mathbf{y}}_n^\top \mathbf{y}_n - \xi, \quad \forall \mathbf{y} \in \mathcal{Y}$$

  with an equivalent single non-linear constraint

$$\mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \max_{\mathbf{y} \in \mathcal{Y}} \quad \mathbf{w}\mathbf{X}\mathbf{y} - \hat{\mathbf{y}}_n^\top \mathbf{y}_n.$$

- Thus need to find $\mathbf{y}$ with highest potential relative to parameterization $\mathbf{w}\mathbf{X} - \hat{\mathbf{y}}_n^\top$.
  – The same form as the LP formulation of the MAP problem.
  – Can be solved approximately, either by solving LP or using graph-cut based alpha-expansion (faster in practice).

# QP Solution

- Substituting the (dual of) MAP LP into the QP, and after some (possibly hairy) algebraic manipulation:

$$\begin{aligned}
\min \quad & \frac{1}{2}||\mathbf{w}||^2 + C\xi && (4)\\
\text{s.t.} \quad & \mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \sum_{i=1}^{N} \alpha_i; \qquad \mathbf{w}_e \geq 0; \\
& \alpha_i - \sum_{ij,ji \in \mathcal{E}} \alpha_{ij}^k \geq \mathbf{w}_n^k \cdot \mathbf{x}_i - \hat{y}_i^k, \quad \forall i, k; \\
& \alpha_{ij}^k + \alpha_{ji}^k \geq \mathbf{w}_e^k \cdot \mathbf{x}_{ij}, \quad \alpha_{ij}^k, \alpha_{ji}^k \geq 0, \quad \forall ij \in \mathcal{E}, k.
\end{aligned}$$

# QP Solution (cont.)

- …and the dual:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} (1 - \hat{y}_i^k)\mu_i^k - \frac{1}{2} \sum_{k=1}^{K} \left\| \sum_{i=1}^{N} \mathbf{x}_i (C\hat{y}_i^k - \mu_i^k) \right\|^2 \\
& -\frac{1}{2} \sum_{k=1}^{K} \left\| \lambda^k + \sum_{ij \in \mathcal{E}} \mathbf{x}_{ij} (C\hat{y}_{ij}^k - \mu_{ij}^k) \right\|^2 \\
\text{s.t.} \quad & \mu_i^k \geq 0, \quad \forall i, k; \qquad \sum_{k} \mu_i^k = C, \quad \forall i; \\
& \mu_{ij}^k \geq 0, \quad \mu_{ij}^k \leq \mu_i^k, \quad \mu_{ij}^k \leq \mu_j^k, \quad \forall ij \in \mathcal{E}, k; \\
& \lambda^k \geq 0, \forall k.
\end{aligned}
$$

# QP Solution (cont.)

- After solving the QP, the primal and dual solutions are related by:

$$\mathbf{w}_n^k = \sum_{i=1}^{N} \mathbf{x}_i (C\hat{y}_i^k - \mu_i^k),$$
$$\mathbf{w}_e^k = \lambda^k + \sum_{ij \in \mathcal{E}} \mathbf{x}_{ij} (C\hat{y}_{ij}^k - \mu_{ij}^k).$$

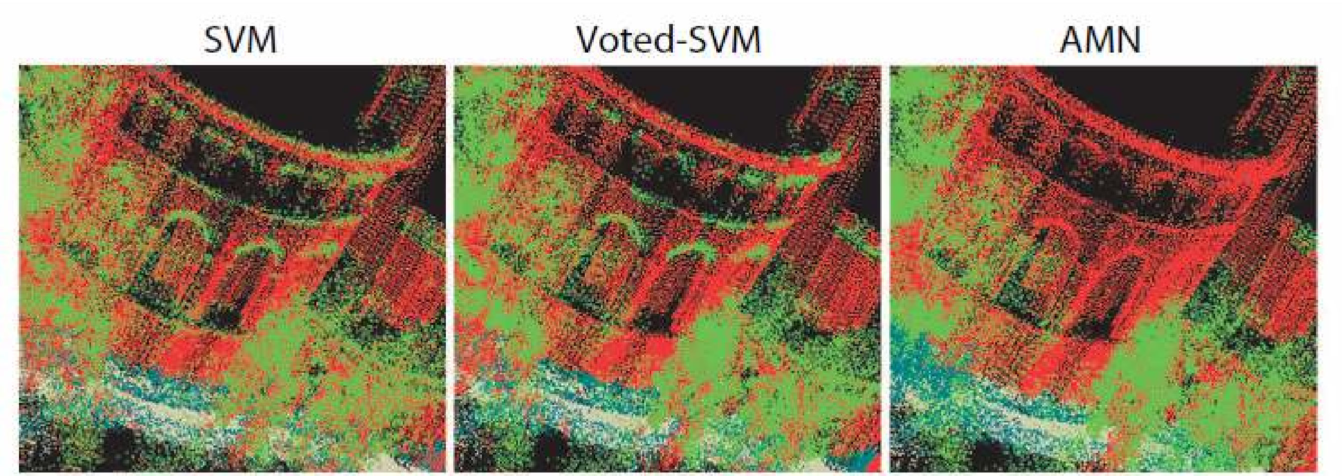
- Kernels can be used on node parameters. However, the extra $\lambda^k$ term prevents edge parameters from being kernelized.

# Experimental Results

- Two real-world and one synthetic datasets
  - Terrain classification
  - Segmentation of articulated objects
  - Princeton benchmark
- Compare against multi-class SVM
  - On each dataset, AMN and SVM use the same set of features.

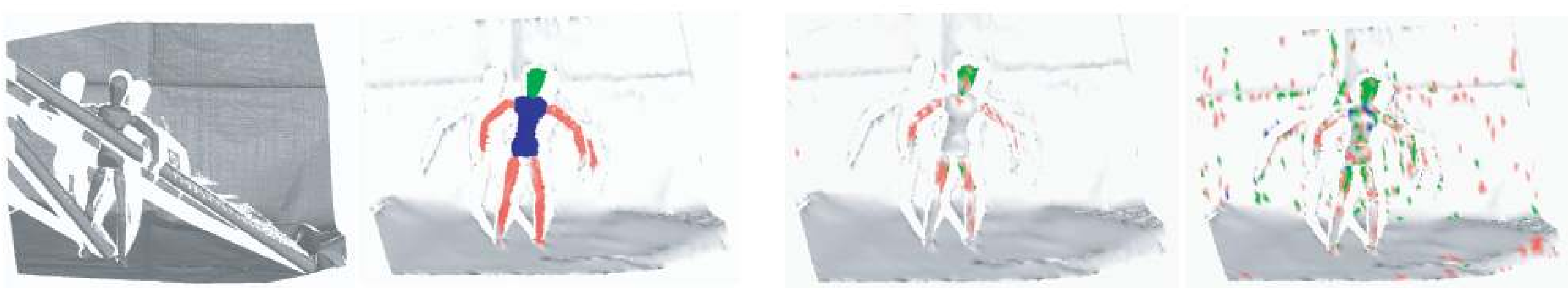
# Terrain Classification

- Campus map built by mobile robot with scanner
  - Four types of terrains: ground, tree, building, and shrubbery
  - Use quadratic kernel
  - Locally sampled edges for AMN
- Accuracy:
  - SVM: 68%, Voted SVM: 73%, and AMN: 93%



# Segmentation of Articulated Objects

- Puppet dataset
  - Four object classes: puppet head, limb, torso, and background
  - Uses surface links output by the scanner as MRF edges.
- Results
  - AMN: accuracy 94.4%, precision 83.9%, recall 86.8%
  - SVM: accuracy 87.16%, precision 93%, recall 18.6%



a) Training instance b) AMN c) SVM d) AMN with edges ignored

# Princeton Benchmark

- Artificially generated scenes
  - Two classes: vehicles and background
  - Readings of “virtual sensor” corrupted by additive white noise
  - Use the same set of features as in the puppet dataset
- Accuracy
  - AMN: 93.76%, SVM: 82.23%

# Conclusion

- MRF-based method for segmentation
  - MAP estimate using graph-cut
  - Max-margin training using QP
- Future work
  - More appropriate kernels
  - Spatial model of objects