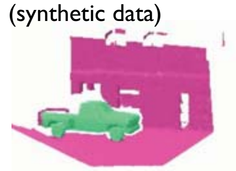
Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data

Anguelov et. al., (CVPR), 2005

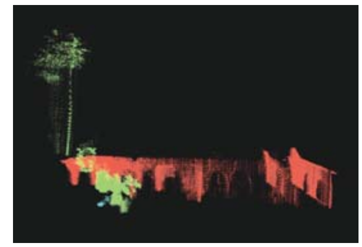
Presentation for CS 6784
Sarah Iams, 18 Feb 2010

Intuition for the problem

(1) ID vehicles vs background (synthetic data)



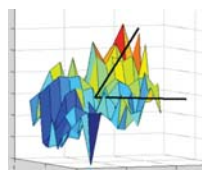
(2) Find buildings, trees, shrubs, ground

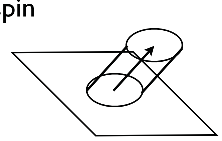


(3) Find head, limbs, torso, background

Features

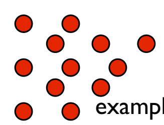
- How planar is the neighborhood of the point?
- Is a point close to the ground?
- Are there many points nearby?
- What are the principal components of the spin images?





Capture problem structure

- Markov network captures geometry of the problem
- Scan points are represented by nodes in a graph
- Edges connect nearby scan points
- Each node will eventually have a label, $Y_i \in \{1, \ldots, K\}$
- The entire network is associated with a set of labels, $\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$
- They are interested in a distribution over $\{1, \ldots, K\}^N$ specified by the geometry of the graph

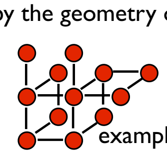


example:
spatial structure embedded in network

example:
one possible labeling
$\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$

Capture problem structure

- Markov network captures geometry of the problem
- Scan points are represented by nodes in a graph
- Edges connect nearby scan points
- Each node will eventually have a label, $Y_i \in \{1, \ldots, K\}$
- The entire network is associated with a set of labels, $\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$
- They are interested in a distribution over $\{1, \ldots, K\}^N$ specified by the geometry of the graph

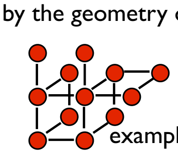


example:
spatial structure embedded in network

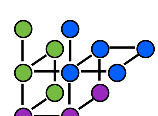
example:
one possible labeling
$\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$

Capture problem structure

- Markov network captures geometry of the problem
- Scan points are represented by nodes in a graph
- Edges connect nearby scan points
- Each node will eventually have a label, $Y_i \in \{1, \ldots, K\}$
- The entire network is associated with a set of labels, $\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$
- They are interested in a distribution over $\{1, \ldots, K\}^N$ specified by the geometry of the graph



example:
spatial structure embedded in network



example:
one possible labeling
$\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$

Pairwise MRF assumption

- pairwise Markov network: nodes and edges are associated with potentials, $\phi_i(Y_i)$ and $\phi_{ij}(Y_i,Y_j)$
- all potentials are then multiplied (and normalized) to produce P(Y|X)

$$P(Y|X) = \frac{1}{Z}\prod_i \phi_i(Y_i)\prod_{ij}\phi_{ij}(Y_i,Y_j)$$

- This is identical to saying the logs of the potentials are added to produce log P(Y|X)

$$\log P(Y|X) = \sum_i \log \phi_i(Y_i) + \sum_{ij}\log\phi_{ij}(Y_i,Y_j) - \log(Z)$$

- the feature values, ψ_i, at each node dictate the values of $\phi_i(Y_i)$

$$\log \phi_i(k) = \mathbf{w}_n^k \cdot \psi_i$$

- the similarity of the prospective labels, ψ_{ij}, along an edge dictates $\phi_{ij}(Y_i,Y_j)$

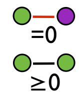
$$\log \phi_{ij}(k,l) = \mathbf{w}_e^{kl} \cdot \psi_{ij}$$

AMN assumption

- want to find the **Y** that maximized P(Y|X). Note maximizing P(Y|X) is identical to maximizing log P(Y|X)

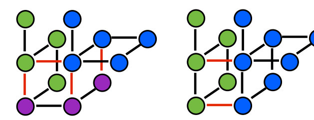
$$\arg\max_{\mathbf{Y}} \log P(Y|X) = \arg\max_{\mathbf{Y}}\left(\sum_i \log\phi_i(Y_i) + \sum_{ij}\log\phi_{ij}(Y_i,Y_j) - \log(Z)\right)$$

- they make one more assumption to simplify the optimization problem: edge weight is 0 when an edge connects nodes with different labels. Otherwise, the weight is non-negative.



$$\log \phi_{ij}(k,l) = 0 \text{ for } (k \neq l)$$
$$\log \phi_{ij}(k,k) = \mathbf{w}_e^k \cdot \psi_{ij} \geq 0$$

- This is the associative Markov network assumption.



Optimization problem

$$\arg\max_{\mathbf{Y}} \log P(Y|X) = \arg\max_{\mathbf{Y}}\left(\sum_i \log\phi_i(Y_i) + \sum_{ij}\log\phi_{ij}(Y_i,Y_j) - \log(Z)\right)$$

$$\log \phi_i(k) = \mathbf{w}_n^k \cdot \psi_i$$
$$\log \phi_{ij}(k,l) = 0 \text{ for } (k \neq l)$$
$$\log \phi_{ij}(k,k) = \mathbf{w}_e^k \cdot \psi_{ij} \geq 0$$

$$\arg\max_{\mathbf{Y}} \log P(Y|X) = \arg\max_{\mathbf{Y}}\left(\sum_i\sum_k(\mathbf{w}_n^k\cdot\psi_i)y_i^k + \sum_{ij}\sum_k(\mathbf{w}_e^k\cdot\psi_{ij})y_{ij}^k\right)$$

- Given weights, we can solve this (min-cut algorithm)
- Or (evidently), we can reformulate as integer program & relax to linear program: they choose this route because this arg max will reappear in the course of their learning method!

Learning method

$$\arg\max_{\mathbf{Y}} \log P(Y|X) = \arg\max_{\mathbf{Y}}\left(\sum_i\sum_k(\mathbf{w}_n^k\cdot\psi_i)y_i^k + \sum_{ij}\sum_k(\mathbf{w}_e^k\cdot\psi_{ij})y_{ij}^k\right)$$
$$= \arg\max_{\mathbf{y}} \mathbf{wXy}$$

- Switch to vector notation (all those subscripted w's, ψ's & y's become vectors in a natural way, with ψ→**X**)
- They take a single training scene.
- Could train weights to maximize $P(Y_{correct}|X)$
- Instead, maximize confidence in correct answer: $P(Y_{correct}|X)$-P(Y|X) (where $Y_{correct}$ is the true label, and Y is any other labeling - this is maximum margin for the Markov network)
- Advantages: allows some kernelization later on
- Evidently pretty accurate

M^{3}N problem

$$\max_{\gamma}\ s.t.\quad \mathbf{wX}(\mathbf{y}_{\mathbf{correct}} - \mathbf{y}) \geq \gamma\Delta(\mathbf{y}_{\mathbf{correct}},\mathbf{y});\ \|\mathbf{w}\|^2 \leq 1$$
$$\Delta(\mathbf{y}_{\mathbf{correct}},\mathbf{y}) = N - \mathbf{y}^{\mathbf{T}}_{\mathbf{correct,nodes}}\mathbf{y}_{\mathbf{nodes}}$$

- Note that **y** is an indicator vector, so when $\mathbf{y}_{correct}$ and **y** agree on a node label, that contributes to their dot product. When they disagree, it contributes 0 to the dot product.
- They define the loss function to count how many times **y** is wrong on labeling the nodes. (Note M^{3}N was approached without a loss function restriction on Tuesday).
- As usual, next they'll divide through by the margin (γ) and add a slack variable (in case the data isn't separable)

Primal formulation

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \ \text{ s.t. }\ \mathbf{wX}(\mathbf{y}_{\mathbf{correct}} - \mathbf{y}) \geq N - \mathbf{y}^{\mathbf{T}}_{\mathbf{correct,nodes}}\mathbf{y}_{\mathbf{nodes}} - \xi \quad \forall \mathbf{y}$$

- this is a quadratic program
- exponentially many constraints
- we can replace the constraints with a single constraint over a quadratic program!

$$\mathbf{wX}(\mathbf{y}_{\mathbf{correct}} - \mathbf{y}) \geq N - \mathbf{y}^{\mathbf{T}}_{\mathbf{correct,nodes}}\mathbf{y}_{\mathbf{nodes}} - \xi \ \forall \mathbf{y}$$
$$\Rightarrow \mathbf{wXy}_{\mathbf{correct}} - N + \xi \geq \mathbf{wXy} - \mathbf{y}^{\mathbf{T}}_{\mathbf{correct,nodes}}\mathbf{y}_{\mathbf{nodes}} \ \forall \mathbf{y}$$
$$\Rightarrow \mathbf{wXy}_{\mathbf{correct}} - N + \xi \geq \max_{\mathbf{y}} \mathbf{wXy} - \mathbf{y}^{\mathbf{T}}_{\mathbf{correct,nodes}}\mathbf{y}_{\mathbf{nodes}}$$

- we recognize this quadratic program from before
- Recall: $\arg\max_{\mathbf{Y}} \log P(Y|X) = \arg\max_{\mathbf{y}} \mathbf{wXy}$

Switch to dual (twice)

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \ \text{ s.t. } \ \mathbf{wXy_{correct}} - N + \xi \geq \max_{\mathbf{y}} \mathbf{wXy} - \mathbf{y^T_{correct,nodes}y_{nodes}}$$

- They switch to the dual problem in the constraint.

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \ \text{ s.t. } \ \mathbf{wXy_{correct}} - N - \xi \geq \sum_{i=1}^{N} \alpha_i; \mathbf{w_e} \geq 0; \alpha_i - \sum_{ij} \alpha^k_{ij} \geq w^k_n \cdot \psi_i - y^k_{correct,i}$$

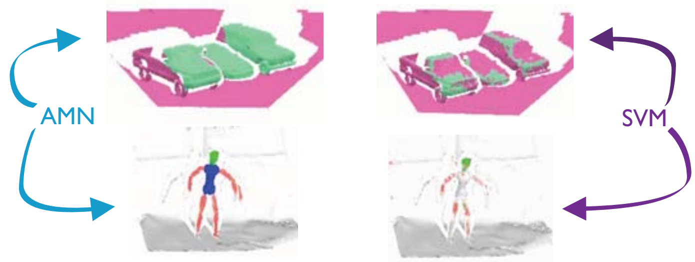
$$\alpha^k_{ij} + \alpha^k_{ji} \geq w^k_e \cdot \psi_{ij}; \alpha^k_{ij}, \alpha^k_{ji} \geq 0$$

- Then they switch to the dual in the overall problem. (I am not including the dual here.) The primal and dual are related as follows: $w^k_n = \sum_{i=1}^{N} \psi_i (C y^k_{correct,i} - \mu^k_i)$ $\quad w^k_e = f(\phi^k_{ij}) + \sum_{i,j} \psi_{ij} (C y^k_{correct,ij} - \mu^k_{ij})$

- Since w_n^k is a sum over ψ_i multiplied by constants, $w_n^k\psi$ can be kernelized. The edge potentials cannot be, however, because of the constant term added to the sum.

Testing the AMN

- The associative Markov network ensures nearby points have the same label (SVM does not do this)
- After five training scenes:



Testing the AMN