Lecture 11. Hierarchical Models.

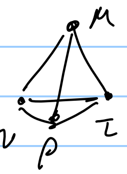
Note Title 2/14/2010

Basic Idea — build a hierarchical model of an object by composition of elementary components. This structure enables efficient inference and learning.

Simple Case:

$G = (\mathcal{V}, \mathcal{E})$ $\quad \mathcal{V} = (\mu, \nu, \rho, \tau)$

$\mathcal{E} = (\mu\nu, \mu\rho, \mu\tau, \nu\rho, \rho\tau, \nu\tau)$



parent node μ

child nodes $\{\nu, \rho, \tau\} = ch(\mu)$.

State variables:

$\omega_\mu = (\underline{x}_\mu, \theta_\mu, s_\mu)$ — position, orientation & size.

$\omega_{ch(\mu)} = (\omega_\nu, \omega_\rho, \omega_\tau)$

Potentials: $\underline{\lambda}^G \cdot \underline{\varphi}^G(\omega_\mu, \underline{\omega}_{ch(\mu)})$

spatial relations between the nodes.

Basic Assumption: the object is present in the image, but we don't know where.

data terms $\underline{\lambda}^D_\tau \cdot \underline{\varphi}^D(\omega_\tau : I)$, $\tau \in \mathcal{V}$. (image)

$$P[\{\omega\} | I] = \frac{1}{Z[I]} e^{-E[\{\omega\}, I]}$$

where
$$E[\{\omega\}, I] = \sum_{\tau \in \mathcal{V}} \underline{\lambda}^D_\tau \cdot \underline{\varphi}^D(\omega_\tau : I) + \underline{\lambda}^G \cdot \underline{\varphi}^G(\omega_\mu, \underline{\omega}_{ch(\mu)})$$

Example:

Leaf nodes ν, ρ, τ represent edgelets.

Higher node μ represents a composition of edgelets.

$\omega_\mu = f(\omega_{ch(\mu)})$ deterministic function.

Gaussian function relating $\omega_\nu, \omega_\rho, \omega_\tau$

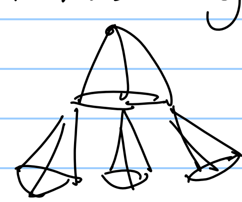
Representation — Executive Summary.

ω_μ describe the position, orientation, and scale at coarse level.

The child nodes $\omega_\nu, \omega_\rho, \omega_\tau$ give more precise description of the object.

Composition of compositions.

We can make a bigger model by combining elementary parts.



(2) Recursive formulation of the energy:

Full energy: $E(\omega|I) = \sum_{\mu \in \mathcal{V}} \underline{\lambda}^D_\mu \cdot \underline{\phi}^D(\omega_\mu, I)$

$\mathcal{V}_{LEAF}$ – leaf nodes. Those without children.

$+ \sum_{\mu \in \mathcal{V}/\mathcal{V}_{LEAF}} \underline{\lambda}^G_\mu \cdot \underline{\phi}(\omega_\mu, \omega_{ch(\mu)})$

The energy can be computed recursively.

Subtree with root v, $\mathcal{V}_v$, $\omega_{des(v)} = \{\omega_\mu : \mu \in \mathcal{V}_v\}$

$\mathcal{V}_v = \{v, ch(v), \{ch(\rho) : \rho \in ch(v)\}, etc..\}$

$$E_v(\omega_{des(v)}|I) = \sum_{\rho \in ch(v)} E_\rho(\omega_{des(\rho)}|I)$$
$$+ \underline{\lambda}^G_v \cdot \underline{\phi}^G(\omega_v, \omega_{ch(v)}) + \underline{\lambda}^D_v \cdot \underline{\phi}^D(\omega_v, I).$$

This recursive formulation enables dynamic programming. But pruning is required because the state space ω_μ of each node $\mu \in \mathcal{V}$ is very large.

Pruning by:
(i) energy – remove configurations whose energy is too large.
(ii) surround suppression – remove state which are too similar to each other.

Initialization:
At leaf nodes $v \in V^{LEAF}$
Calculate states $\omega_{v,b}$ s.t.: b indexes the state.
$E_v(\omega_{v,b}|I) < T$, energy pruning.
$E_v(\omega_{v,b}|I) \leq E_v(\omega_v|I)$, $\forall\ \omega_v \in W(\omega_{v,b})$, surround suppression.
W – window.
list $\{(\omega_{v,b}, E_v(\omega_{v,b}|I))\}$

Recursion: For nodes $v \in \mathcal{V}/\mathcal{V}^{LEAF}$.
form all compositions of the child nodes
$\omega_{\rho,a}, \omega_{\tau,b}, \omega_{\lambda,c}$ $\{\rho, \tau, \lambda\} = ch(v)$.
a, b, c indexes possible states of the children.

note: each $\omega_{\rho,a}$ is formed by compositions of child nodes and so on recursively. Hence corresponds to a specification of node variables for the entire subtree $\rightarrow$ $\omega_{des(\rho,a)}$

(3)

Composition of $\omega_{\rho,a}$, $\omega_{\tau,b}$, $\omega_{\lambda,c}$ and their subtree states

$\omega_{des(\rho,a)}$ with $E_\rho(\omega_{des(\rho,a)})$

$\omega_{des(\tau,b)}$ with $E_\tau(\omega_{des(\tau,b)})$

$\omega_{des(\lambda,c)}$ with $E_\lambda(\omega_{des(\lambda,c)})$

The energy for the composition is

$$E_\nu(\omega_{des(\nu,d)}|I) = E_\rho(\omega_{des(\rho,a)}) + E_\tau(\omega_{des(\tau,b)}) + E_\lambda(\omega_{des(\lambda,c)}) + \underline{\lambda}^G_\nu \cdot \underline{\phi}^G(\omega_{\nu,d};\omega_{\rho,a},\omega_{\tau,b},\omega_{\lambda,c}) + \underline{\lambda}^D_\nu \cdot \underline{\phi}^D(\underline{\omega}_{\nu,d}, I)$$

where $\omega_{des(\nu,d)} = \{\omega_{\nu,d}, \omega_{des(\rho,a)}, \omega_{des(\tau,b)}, \omega_{des(\lambda,c)}\}$.

prune by:

energy $\rightarrow$ keep configurations only if $E_\nu(\omega_{des(\nu,d)}|I) < T$.

surround suppression $\rightarrow$ require $E_\nu(\omega_{des(\nu,d)}|I) \leq E_\nu(\omega_{des(\nu)}|I)$ for $\omega_\nu \in W(\omega_{\nu,d})$

$\rightarrow$ with $\omega_{des(\nu)} = \omega_\nu, \omega_{des(\rho,a)}, \omega_{des(\tau,b)}, \omega_{des(\lambda,c)}$.

This is the bottom-up pass.

The output is a set of configurations for the entire tree $\omega_{des(r,a)}$ where r is the root node. a indexes the configuration.

Top down stage – remove all configurations of the subtrees $\omega_{des(\rho,a)}$ which do not form compositions.

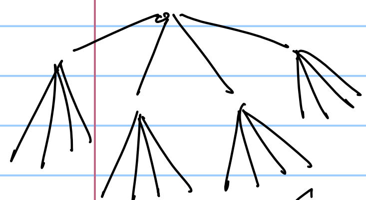
i.e. any $\omega_{\rho,a}$ s.t. $\omega_{\rho,a} \notin \omega_{des(\nu,b)}$ for all b where $\rho \in ch(\nu)$

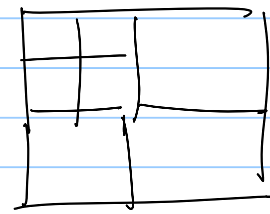
This is like the backward pass of standard DP.

Note: • this process is possibly non-robust, because part of the object may be undetected – e.g., the edge detector may fall below threshold, or part of the object may be occluded.

• 2 out of 3 rule – modify the potential and the inference algorithm so that we can form compositions from only two child nodes ~ use the prior to fill in the 3rd.

(4) Extension to image labelling.





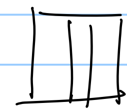
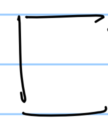
Quadtree representation

Same structure as before — except each node has four children.

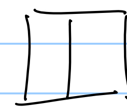
State variables: — represent a partition and labeling of image subregion

$$\omega_\mu = (s_\mu, \underline{\sigma}_\mu)$$

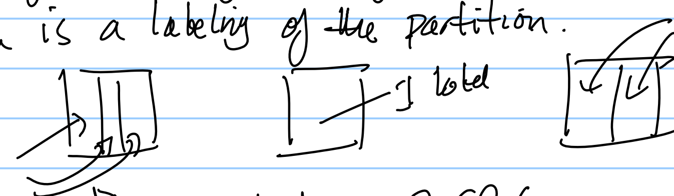
3 region 1 region

s_μ labels a segmentation-template.  or 

there are 40 possible s-t's.

they represent the possible partition of the image subregion. or  ← 2 regions

$\underline{\sigma}_\mu$ is a labeling of the partition.



each label $\sigma \in \{sky, road, etc\}$

potentials $\underline{\lambda} \cdot \underline{\phi}(\omega_\mu, \omega_{ch(\mu)})$

children of μ (four children)

$\underline{\lambda}^D \cdot \underline{\phi}^D(\omega_\mu, I)$ data term.

Energy:
$$E[\omega; I] = \sum_{\mu \in \mathcal{V}} \underline{\lambda}^D \cdot \underline{\phi}^D(\omega_\mu, I) + \sum_{\mu \in \mathcal{V}/\mathcal{V}_{leaf}} \underline{\lambda} \cdot \underline{\phi}(\omega_\mu, \omega_{ch(\mu)})$$

Note: the potentials $\underline{\lambda} \cdot \underline{\phi}(\omega_\mu, \omega_{ch(\mu)})$ encode prior knowledge like the which labels occur next to each other, the consistency between labels at different levels of the hierarchy, etc.

Inference can be performed by DP as before.
The state space is smaller, but we still require pruning on the energy

For both models, we can learn using labelled training data. Structured-SVM and approximations like the structured perceptron are successful.