The previous lecture showed that we could learn a probabilistic grammar Markov model (PGMR) defined on Interest Points (IPs).

But the model is limited because:
(A) It can only perform a subset of visual tasks (e.g., it can not segment or parse).
(B) It uses a restricted set of image cues (e.g., IPs) and so ignore information which might be relevant.

We now show that we can extend the PGMR model to include other cues and to perform other tasks.

This is still an example of structure induction, but the model is more complex and contains disparate cues.
Two types of models

POM-I defined on sparse feature points → form was given in previous lecture. This will be defined on Interest Points (as in previous lecture) or on Edgelets. (POM-I on edgelets with perform parsing)

POM-II defined on a dense object model.

The POM-II is similar to the Grab-Cut model

The POM-II uses statistics inside and outside the model.

Limitation of Grab-Cut: required initialization, no prior on shape, no invariance to scale and rotation.
(3) Strategy for Pursuit:

Learn structure for the Pol-I IP model as before (previous lecture).

This will output a model with several different aspects.

On each image, it will give an estimate for the pose of the object and its silhouette (obtained from the rectangle that contains all the IPs).

The pose and the silhouette can be used to initialize a type of Grab-cut algorithm (recall Grab-cut requires an initialization of the silhouette by the user).

This can be used to segment the object, and to learn a probability model for the shape of the object.
Once the POM-II has been learnt (one for each aspect of the object) we can use it to train POM-I models defined on edgelets. These POM-I models are defined within sub-regions of the object obtained from the silhouette.

They perform a simple form of object parsing.

Knowledge propagation.

The strategy is to grow the model structure by allowing component models (i.e. POM's) to train additional component model by supplying estimates of the missing information (e.g. POM-I with IP's "trains" the POM-II by supplying estimates of the pose & silhouette. The POM-II trains the POM-I edgelets by supplying estimates of their positions.)
More Details:

**Point I's**

\[
P(Z, \theta, A | V, Y, \omega, G) \cdot P(V | l, y) \cdot P(l | s) \cdot p(\theta) \cdot p(\omega) \cdot p(s)
\]

DP \( k \in H \) is used for inference, parameter learning, and model scoring.

**Point II**

\[
P(I, Y, M, q | l, s) = P(I | L, q) \cdot P(l | M, s) \cdot P(L | M, s) \cdot P(M | Y) \cdot P(s)
\]

- \( I \) - intensity (image)
- \( L \) - binary map
- \( L_i = 1 \), pixel \( i \) in object
- \( L_i = 0 \), otherwise
- \( q = (q_{in}, q_{out}) \) - distributions
  - \( q_{in} (I_c) \) - distributions of
  - \( q_{out} (I_c) \) - intensity

\[
P(I | L, q) = \frac{1}{Z} \exp \left( \sum_i \phi_1 (I_i, l_i, q_{in}) + \sum_{i,j} \phi_2 (I_i, I_j, l_i, l_j) \right)
\]

(like a CRF).
\[ P(\mathbf{q}|\mathbf{M}, \mathbf{G}) \] is a prior for the statistics inside and outside the object. It is set to the uniform distribution.

\[ P(\mathbf{L}|\mathbf{M}, \mathbf{G}, \mathbf{S}) \] is the prior probability for the shape of the object conditioned on the aspect \( y \) and the pose \( G \).

\[ P(\mathbf{L}|\mathbf{M}, \mathbf{G}, \mathbf{S}) = \frac{1}{Z} \exp \left\{ \sum_i \Psi_1 (\mathbf{L}_i|\mathbf{M}, \mathbf{G}) + \sum_j \Psi_2 (\mathbf{L}_j|\mathbf{S}, \mathbf{G}) \right\} \]

The potential
\[ \Psi_1 (\mathbf{L}_i|\mathbf{M}, \mathbf{G}) = T(G) (\log \mathbf{M}_i) \]

Here \( T(G) \) transforms the mask to allow for the pose (i.e., translates, rotates, and scales) the mask (\( \mathbf{M}_i \)) is the probability \( \mathbf{M}_i \) that point \( i \) is inside the object. (1-\( \mathbf{M}_i \) is prob that \( i \) is outside)

The potential \( \Psi_2 (\mathbf{L}_j|\mathbf{S}, \mathbf{G}) \) imposes a generic probability that neighboring points in the image tend to have similar labels.
(7) \[ P(M|ly) \] is the probability of the mask depending on the aspect,

\[
P(M|ly) = \frac{1}{5} \sum_{s} P(s) \delta M,M_{s} \quad \text{mask for aspect } s
\]

\[ P(G) \] is the uniform distribution (as before)

We can do inference on Post-II using the max-flow/min-cut algorithm.

We need to simultaneously estimate the \( q_s \)s, which performed directly by estimating the histograms using the current estimate of \( L \) (initialized by Post-I).

Learning the parameters by an approximation to EM. Use max-flow/min-cut to estimate the missing variables.

The results are straightforward - e.g. the probability mask \( M_s \) for aspect \( s \) is learnt by averaging the silhouettes detected by \( L \) over the images with aspect \( s \) (and adjusting for pose).
Combining the Models.

The models are coupled by the pose variable $G$ and the aspect variable $y$.

We have $K$ Port-I models with edgetele.
1. Port-I model with IP's
1. Port-II model

Observables $(z, \theta, A)$ IP's u the image
$$\{(z^e_k, \theta^e_k)\}$$ Edgetele u-the k-subreg
$$\{I\}$$ Intensity.

Variables:
$$y = (s, n)$$ aspect + background IEs
$V$ - corresponding variable for IP model
$V^e_k$ - corresponding variable for edgetele
$G$ - pose variable.
$L$ - silhouette variable

Learned Parameters: probability masks $M_k$. 
\[ (9) \quad P((z, s, a) \mid V, y, G) \]
\[
\prod_{k=1}^{K} P(z_k, s_k \mid V^e, y, G, L) \\
P(I \mid L, q) \\
P(V^1y) \\
P(L \mid H, G) P(M^1y) \\
P(V^e(y)) \\
P(V^e(L))
\]

We perform inference on the entire model. We use approximations to make inference practical:

(i) Port-I with IP to estimate \( y, V \) \& \( G \)

(ii) Port-II to estimate \( L \)

(iii) Port-I edgelets to estimate \( V^e_k \)

The output is the pose \( G \), the silhouette \( L \), the position of the IP feature \( V \), the position of the edgelets \( V^e_k \).
Summary

We learn the full model incrementally.

- When the Port-I is sufficiently complex, we can use it to propose a silhouette structure (bounding rectangle of the IP's) for a Port-II.
- If this proposal is validated – i.e., the Port-II is able to "lock onto" the structure of the object – then we learn a Port-II and couple it to the Port-I.

Then we can use the silhouette detected by the Port-II to activate new Port-I's dubbed over edgelets in subregions of the silhouette.