Felzenswalb

CS 510

Lecture #21

April 29th, 2013

Programming Assignment #4

- Due one week from Friday
  - Any questions?
  - How is it going?

Where are we?

- We have two umbrella schemes for object recognition
  - Bag of Features, Constellations
- We bootstrap these with feature detections
  - Interest points, regions, etc.
- We convert these to feature descriptions
  - SIFT, HoG, LBP, Iconinc
- We group and label them
  - Clustering (K-Means, EM)
  - Classification (SVM, Backprop, Bayes nets, Decision trees, Nearest neighbors)

- *I promised a return to Constellations…. Felzenswalb!*

Felzenswalb's Goals

- Model objects are parts
  - whose positions are inter-related
  - whose appearance is probabilistic
- Solve for positions & appearance simultaneously & efficiently
- Train model from (labeled) examples

Bayesian Basis

- Let I be an image
- Let $\Theta = \{u_1, \ldots, u_n\}$ be an object model
  - U_i is a model of an object part
  - To be described more later
- Let $L=\{l_1,\ldots,l_n\}$ be a configuration
  - A position for each part

$$p(L \mid I,\theta) \propto p(I \mid L,\theta)\, p(L \mid \theta)$$

Bayesian Basis II

- Let's not surf this equation
  - $P(L|I,\theta)$ is the likelihood of an object location, given an image and object model
  - The L that maximizes $P(L|I,\theta)$ is the most likely location for the object
  - The sum of $P(L|I,\theta)$ over all L is the probability that the object is in the image.
- Note that $P(L,\theta) = P(L|\theta)P(\theta)$
- Note its proportional, not equal
  - We dropped $P(\theta)$, the *a-priori* for the object
  - Similarly, we dropped $P(I,\theta)$
- But these don't determine which L is maximal

More Bayesian Basis

- OK, that was only moderately helpful
- But if we assume that parts don't overlap…

$$p(I \mid L, \theta) \propto \prod_i p(I \mid l_i, u_i)$$

- Note that this is only a statement about image formation. *Minus overlap*, its reasonable

Location Modeling

- GHT models locations relative to a reference point
  - But if you model the human body, the hand position depends on the forearm, which depends on the upper arm, which depends on the torso…
- Felzenswalb models objects as a acyclic graph of related parts
  - Nodes are object parts
  - Edges are relative positional constraints

Location II

$$p(L \mid \theta) = p(L \mid E, c) = \prod_{(v_i, v_j) \in E} p(l_i, l_j \mid c_{i,j})$$

- E are the edges in the graph
  - Undirected, must be acyclic
- C describes the relation between i and j
  - In practice, a Gaussian centered at a mean distance, so that $\|l_i - l_j\| = -\log p(l_i, l_j \mid c_{ij})$

Location in Practice

$$p\left(l_i, l_j \mid c_{ij}\right) \propto N\left(T_{ij}\left(l_i\right) - T_{ji}\left(l_j\right), 0, D_{ij}\right)$$

- Where T & D are connection parameters encoded by C
  - T's are translations expected between parts
  - D is a diagonal covariance matrix, giving expected distance variations
- N is a normal distribution

More on Location

- The Graph θ=(E,c) is a restricted form of Bayesian Net
  - Edges capture conditional dependencies
  - Restricted to a tree
  - Accounts for articulation
  - Fails to account for global effects like pose under perspective projection
- "In practice" restricted to simple Gaussians
  - Assumes favored (default) position
  - Distances, not angles
  - Is this a good model?

Stepping Back

- We started with

$$p(L \mid I, \theta) \propto p(I \mid L, \theta)\, p(L \mid \theta)$$

- We reduced p(L|θ) to

$$p(L \mid \theta) = \prod_{(v_i, v_j) \in E} p(l_i, l_j \mid c_{i,j})$$

$$p(L \mid \theta) \propto \prod_{ij} N\left(T_{ij}(l_i) - T_{ji}(l_j), 0, D_{ij}\right)$$

P(I|L,θ)

- This is the probability of generating the image, given object θ at position L
- We have already assumed that object parts don't overlap or occlude each other
- Therefore

$$P(I \mid L,\theta) \propto \prod_i P(I \mid l_i, u_i)$$

$P(I|I_i,u_i)$

- Here, Felzenswalb get wishy-washy
  - Points out many methods could be used
- But what he uses is:
  - Object parts are modeled as points in iconic representation space (~10 dimensions)
  - Every object part has a diagonal coavariance matric in iconic representation space
  - The prob. of an observed point is inversely proportional to its distance in iconic space

$P(I|l_i,u_i)$ – continued

$$P\left(I \mid l_i, u_i\right) \propto N\left(\alpha\left(l_i\right), \mu_i, \Sigma_i\right)$$

- Where
  - $\alpha(l_i)$ is the "iconic index" at l_i.
  - μ_i is the mean "iconic index" for the part
  - Σ_i is the covariance of the "iconic index"

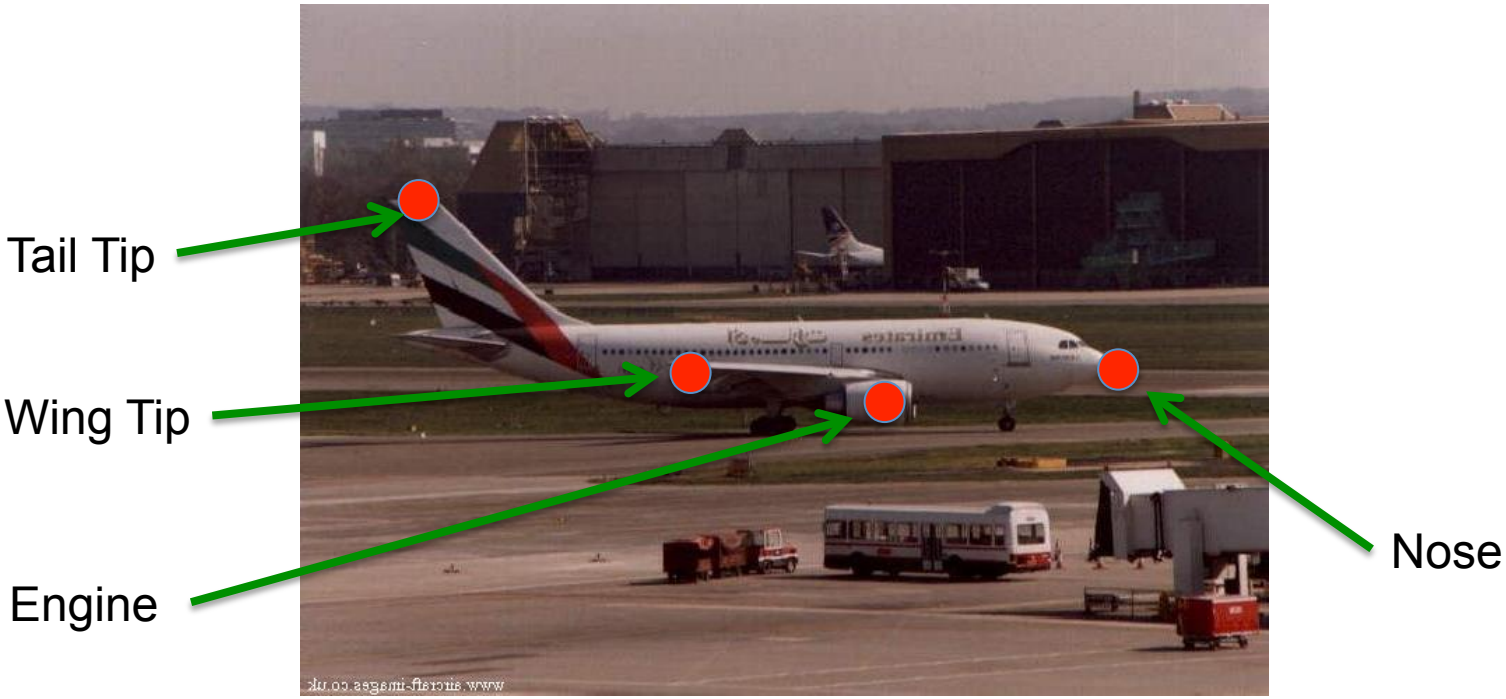
Putting it all together

$$p(I \mid L,\theta) \propto \prod_i N\left(\alpha(l_i),\mu_i,\Sigma_i\right)\prod_{ij} N\left(T_{ij}(l_i)-T_{ji}(l_j),0,D_{ij}\right)$$

- We have an equation!!!
- Two remaining problems:
  1. Where does the model come from?
  2. How to (efficiently) find arg max L?

Training Felzenswalb

- Training samples have labeled points



Training Felzenswalb (II)

- For every part
  - Collect all training samples of that part
  - Compute α for every sample
    - Estimate μ, Σ for the part
- For every pair (i,j) of connected parts in θ
  - Collect samples of pairs
  - Estimate T_{ij} (half the vector from I to j)
  - Estimate D, the covariance matrix of part distances

Training Felzenswalb (III)

- There is even a heuristic for learning the connections between parts
  - Compute the full covariance matrix D
  - Take the largest off-diagonal term, connect those two parts
    - These points depend most strongly on each other
  - Repeatedly take the next largest off-diagonal term
    - As long as it doesn't introduce a cycle
    - Until all parts are connected
- Note that some dependencies will be missed
  - But in general, they will be the smaller ones
  - The conditional independence created by the graph is only approximate, anyway

Intuitions

- The pieces of the model are not sophisticated
  - Appearances modeled as points in iconic representation space
  - Variations in appearance as distances in iconic space
  - Relative positions as vectors + Gaussian noise
- Power comes from the combination of lots of unsophisticated models
- Simple models make training easy

Solving For ArgMax L

$$p(I \mid L,\theta) \propto \prod_i N(\alpha(l_i),\mu_i,\Sigma_i) \prod_{ij} N(T_{ij}(l_i) - T_{ji}(l_j),0,D_{ij})$$

- An inefficient solution is easy
  - Given n parts, and a WxH image
  - There are WxH choose n possible L's.
  - Solve for each, take the max
- Solving for ArgMax L is NP-Hard if all parts are connected (i.e. T is full)

$O(h^2n)$ Solution

- A configuration *L* maps every part *i* to an image location (x,y)
  - Let *H* be the grid of all points (x,y)
  - May be courser than the original image resolution
- P(I|L,θ) is a product of terms of parts and limited part pairs
  - Part pairs only for connected parts in the dependency graph
- The tree-shaped dependency graph has traversal orders such that:
  - Any node, when visited, is connected to at most one other node that hasn't already been visited
  - Traversal begins at a leaf (obviously)
- Find ArgMax L by binding parts to locations in this order
  - Similar to Viterbi

$O(h^2n)$ Solution (continued)

- Create an hxn table L
  - The rows are positions (points in H)
  - The columns are parts, in the traversal order
- $L[i,j] = \text{ArgMax } L_{1,\ldots,j-1} P(I|L_{1,\ldots,j-1}, L_{j=i}, \theta)$
- Intuitively, L[i,j] is the $P(I|L,\theta)$ if part j is bound to location i, and parts 1…j-1 are bound optimally.
- $L[i,j] = \text{Max}_k L[k,j-1] N(\alpha(I_i),\mu_j,\Sigma_j) N(T_{kj}-T_{jk},0,D_k)$
- This can be computed in $O(h^2n)$ time

Efficient Solution (surfed)

- $O(h^2n)$ is still too slow (h is large)
- O(hn) algorithm based on not trying all positions for all points
  - Given a set K points on a grid, it is possible to compute the distance to the nearest point in K for all grid points in O(k) time [Borgefors 1986]
  - This can be modified for probabilities (not distances)
  - This allows L[i,j] to be estimated without considering all previous bindings K
    - Computing only a fixed number reduced the complexity to O(hn)