A Discriminatively Trained, Multiscale, Deformable Part Model

Pedro Felzenszwalb, David McAllester, Deva Ramanan

Presented by Jay Gokhale
The Problem

Object class detection
Key Contributions

• Success with a deformable parts model.
• Formulation of Latent SVM for solving detection problem.
Model
Optimization Problem

\[ f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z), \]

\[ \beta^*(D) = \arg\min_{\beta} \lambda \|\beta\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)) \]

\( x = \) Image
\( \beta = \) non-latent parameters to be optimized (filters, spatial model quadratic functions)
\( Z = \) latent parameters to be optimized (locations of root, and parts models)
\( \Phi = \) Feature function given image and latent parameters
\( D = \) Whole Dataset
\( n = \) Total number of images in dataset
\( Y_i = \) label for \( i^{\text{th}} \) image in \( D \)
Semi-convexity

\[ f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z), \]

\[ \beta^*(D) = \arg\min_\beta \|\beta\|^2 + \sum_{i=\text{pos}}^n \max(0, 1 - \beta \cdot \Phi(x, z)) + \sum_{i=\text{neg}}^n \max(0, 1 + \beta \cdot \Phi(x, z)) \]

If \( f(x) \) is linear in \( \beta \), this is a standard linear SVM (convex)

If \( f(x) \) is arbitrary, this is in general not convex

If \( f(x) \) is convex in \( \beta \), the hinge loss is convex for negative examples (semi-convex)
- hinge loss is convex in \( \beta \) if positive examples are restricted to single choice of \( Z(x) \)

Slide taken from: http://www.cs.cmu.edu/~efros/courses/LBMV09/presentations/latent_presentation.pdf
Problem: Semi-convexity

- Latent (unknown) variables, Z, exist and create a non-convex cost function.
- When latent variables are fixed, problem becomes a regular linear SVM.
- Solution: Solve iteratively holding either regular parameters or latent variables fixed.
Coordinate ascent

• Solve \( \beta \) (non-latent) and \( Z \) (latent) separately.
• 1) Fix \( \beta \) and optimize \( Z \) for positive examples
• 2) Fix \( Z \) and optimize \( \beta \)
• 3) Repeat
Mining “Hard” Negatives

• What is a hard negative?

\[ K(x_i, x_j) = \phi^T(x_i) \phi(x_j) \]

\[ w^T \phi(x) + b = -1 \]

\[ w^T \phi(x) + b = 0 \]

\[ w^T \phi(x) + b = +1 \]

\[ \xi > 1 \]

\[ \xi < 1 \]

\[ \xi = 0 \]

Margin = \( \frac{1}{\sqrt{w^T w}} \)
Mining “Hard” Negatives

• Hard to find a representative set of negatives and also find enough without overloading practical memory constraints.

• Solution: Find the trickiest negatives and solve with those.
Theorems:

• Terms:
  – $D = \text{Dataset of all images (E.G. PASCAL)}$
  – $\beta^*(D) = \text{The globally optimal parameters for } D$
  – $M(\beta, D) = \text{The set of “hard” examples (positive and negative) given a specific set of parameters, } \beta, \text{ and a dataset, } D$
Theorem 1

• Let $C$ be a subset of the examples in $D$. If $M(\beta^*(D), D) \subseteq C$ then $\beta^*(C) = \beta^*(D)$
Theorem 2

If $\beta^*(M(\beta,D)) = \beta^*(D)$
Theorem 3

- $D = \text{Whole Dataset}$
- $C = \text{Initial cache of examples, subset of } D$
- Algorithm:
  - Optimize $\beta$: $\beta = \beta^*(C)$
  - Shrink $C$: $C = M(\beta, C)$
  - Grow $C$ by adding a sample from $M(\beta, D)$ until a memory limit is reached.

- If $|C| < L$ after each iteration, the algorithm will converge to $\beta = \beta^*(D)$ in finite time.
Results

• Results on PASCAL dataset
  – 2006 2X improvement over average precision
  – 2007 (Best in 10/20 categories)
Results
Additional

• Practical system:
  – Root filter initialization
  – Root filter update
  – Part initialization
  – Model update
Root filter initialization

• Filter dimensions:
  – Aspect = mode of object bounding box in PASCAL
  – Size = 0.8*max size in PASCAL

• Train SVM using
  – Positives: bounding boxes resized to size of root filter
  – Negatives: sample from negative images
Root filter update

• For each bounding box:
  – Find best placement of root filter for that bounding box
  – Do this with unscaled images (unlike previous step)

• Find best filter for these new positives given the previous negatives
Part initialization

• 6 parts of size $a$, such that:
  – $6*a = 80\%$ size of root filter
  – Find 6 regions within root filter with highest energy
  – Part filters initialized with values of the root filter in the subregion
  – Initial deformation costs are squared norms
Model update

• Apply filter to every position and scale with 50% overlap with bounding box and find best placement
• Find tricky negatives (as many a memory allows)
• Run SVM
• Rinse and repeat! (This was done 10 times)