Deformable Part Models

References:
Felzenszwalb, Girshick, McAllester and Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” PAMI 2010
Code available at http://www.cs.berkeley.edu/~rbg/latent/

Recall: Dalal and Triggs, 2005
• Detect upright pedestrians
• Histogram of oriented gradient feature vector
• Linear SVM classifier; sliding window detector

Slide from Pedro Felzenswalb

Today: Deformable Part Models

Motivation
• Simple template-based object models lack ability to handle geometric deformations due to articulation and pose.
• Simple bag of words models have no trouble with deformations, but are limited in their ability to finely localize objects.
• Want to make more expressive models based on geometric deformations of a canonical configuration of object parts.

DPM Innovations
• Star-structured graph model to represent body parts and their geometric relations
• Mixtures of models to handle large variations in viewpoint / pose
• Latent-SVM formulation
• Data mining of hard negative examples
• PCA on HOG features for dimension reduction
• Performance on PASCAL challenge dataset

Motivation
• Problem: More expressive object models are difficult to train because they often use latent (unobserved/unlabeled) information.
• Example: learning a part-based model from images where only bounding boxes are labeled.

What we have: bounding box
What we want: torso, head, arms and legs
History: Pictorial Structures

- Introduced by Fischler and Elschlager in 1973
- Part-based models:
  - Each part represents local visual properties
  - “Springs” capture spatial relationships

Matching model to image involves joint optimization of part locations “stretch and fit”

Felzenswalb showed that the best alignment of parts for a tree-structured model can be computed efficiently using dynamic programming.

Pictorial Structures: Matching

- Model is represented by a graph $G = (V, E)$
  - $V = \{v_1, \ldots, v_n\}$ are the parts
  - $(v_i, v_j) \in E$ indicates a connection between parts
- $m_i(l_i)$ is score for placing part $i$ at location $l_i$
- $d(l_i, l_j)$ is a deformation cost
- Optimal configuration for the object is $L = \{l_1, \ldots, l_n\}$ maximizing

$$E(L) = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d(l_i, l_j)$$

From Pictorial Structures to DPM

- Specifies unary costs, Multiscale detector, Pairwise costs.
- Detection, root filter, part filters, deformation models.

Model has a root filter plus deformable parts

Histogram of Gradient (HOG) Features

- Image is partitioned into 8x8 pixel blocks
- In each block we compute a histogram of gradient orientations
  - Invariant to changes in lighting, small deformations, etc.
- We compute features at different resolutions (pyramid)
Filters

- Filters are rectangular templates defining weights for features
- Score is dot product of filter and subwindow of HOG pyramid

Score of $H$ at this location is $H \cdot W$

Parts are represented at twice the resolution of the root filter.

Object Hypothesis

Scoring with Linear Classifiers

- Score of model is sum of filter scores plus deformation scores
  - Bounding box in training data specifies that score should be high for some placement in a range

$\Phi(I, p_0), \ldots, \Phi(I, p_n)$

More Specifically...

$w$ is a model
$x$ is a detection window
$z$ are filter placements

$F_w(x) = \max_z w \cdot \Phi(x, z)$

concatenation of features and part displacements

$F_w(x) = \max_z w \cdot \Phi(x, z)$

concatenation of filters and deformation parameters

$[\Phi(p_0), \ldots, \Phi(p_n), (dx_1, dy_1, dx_2, dy_2, \ldots, dx_n, dy_n)]$

Scoring with Linear Classifiers

$F_w(x) = \max_z w \cdot \Phi(x, z)$

concatenation of features and part displacements

$[\Phi(p_0), \ldots, \Phi(p_n), (dx_1, dy_1, dx_2, dy_2, \ldots, dx_n, dy_n)]$

Detection Overview
Handling Large Variation in Viewpoints

Solution: Use Mixture Model

Solution: Use a Mixture Model

Training

Part 2: DPM parameter learning
Recall: SVM Training

1. Maximize margin $2/||w||$
2. Correctly classify all training data points:
   - Positive ($y_i = 1$): $x_i \cdot w + b \geq 1$
   - Negative ($y_i = -1$): $x_i \cdot w + b \leq -1$

Quadratic optimization problem:

$$\text{Minimize} \quad \frac{1}{2} w^T w$$
$$\text{Subject to} \quad y_i (w \cdot x_i + b) \geq 1$$

Rewrite:

$$w^* = \arg\min_w \lambda ||w||^2 + \sum_{i=1}^n \max(0, 1 - y_i f_w(x_i))$$

Aside: SVM and Hinge Loss

$$w^* = \arg\min_w \lambda ||w||^2 + \sum_{i=1}^n \max(0, 1 - y_i f_w(x_i))$$

To maximize the margin, which is $2/||w||$.

"Hinge" loss

Intuition: the inequality constraints $f_w(x_i) > 1$ and $f_w(x_i) < -1$ contribute a linear penalty when they are not satisfied, but are not penalized when the inequalities hold.

SVM vs LSVM Training

$$w^* = \arg\min_w \lambda ||w||^2 + \sum_{i=1}^n \max(0, 1 - y_i f_w(x_i))$$

SVM

$$f_w(x) = w \cdot \Phi(x)$$

LSVM

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$

Max over latent positions for part $x_i$, $z \in Z(x)$

This introduces some difficulties during training.
SVM vs LSVM Training

\[ w^* = \arg \min_w \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i)) \]

\[ f_w(x_i) = w \cdot \Phi(x_i) \]

SVM

Convex problem, guaranteed optimal solution!

\[ f_w(x_i) = \max_{z \in Z(x_i)} w \cdot \Phi(x_i, z) \]

LSVM

Only semi-convex, no guaranteed optimal solution

Positive examples (y = +1)

\[ x \] specifies an image and bounding box

We want

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

to score \( \geq +1 \)

\( Z(x) \) includes all \( z \) with more than 70% overlap with ground truth

Negative examples (y = -1)

\[ x \] specifies an image and a HOOG pyramid location \( p_i, \) \( \theta_i \)

We want

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

to score \( \leq -1 \)

\( Z(x) \) restricts the root to \( p_i \) and allows any placement of the other filters

How we learn parameters: latent SVM

\[ E(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i)) \]

\[ E(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - \max_{z \in Z(x_i)} w \cdot \Phi(x_i, z)) \]

\[ + C \sum_{i=1}^{n} \max(0, 1 + \max_{z \in Z(x_i)} w \cdot \Phi(x_i, z)) \]
How we learn parameters: latent SVM

\[ E(w) = \frac{1}{2}||w||^2 + C \sum_{n=1}^{N} \max(0, 1 - y_n \phi_n(x_n)) \]

Observations

Latent SVM objective is convex in the negatives but not in the positives. Since whole objective function is no longer convex, there is no longer a single global optimum, so no easy solution method.

Since whole objective function is no longer convex, there is no longer a single global optimum, so no easy solution method.

ApproximaDon by upper bound

Current estimates of latent configurations that yield upper bound for each positive example.

Auxiliary objective

Let \( Z = \{Z_1, Z_2, \dots \} \)

\[ E(w, Z_i) = \frac{1}{2}||w||^2 + C \sum_{n=1}^{N} \max(0, 1 - w \cdot \phi_n(x_n)) \]

Yields upper bound for whole objective function.

Note that \( E(w, Z_i) \geq \min\ E(w) = E(w) \)

Yields upper bound for whole objective function.

Note that \( E(w, Z_i) \geq \min\ E(w) = E(w) \)

And it is a tight upper bound

\[ w^* = \min_{w, Z_i} \ E(w) = E(w) \]
Unfortunately, this isn’t easy to optimize either, not to mention it is based on estimates of upper bound latent configurations, which could be wrong.

Iterative approach to solving this. NOT guaranteed to find optimal solution unless you start with good initial estimates.

Find stationary point by coordinate descent on $E(w, Z)$

Initialization: either by picking a $w_0$ (or $Z_0$)

\[
\begin{align*}
Z_0 &= \arg\max_{z \in \mathcal{Z}} w_0 \cdot \Phi(a_i, z) \quad \forall i \in P \\
\end{align*}
\]

Easy to compute

Step 1

\[
Z_{w_0} = \arg\max_{z \in \mathcal{Z}} w_{(w_0)} \cdot \Phi(a_i, z) \quad \forall i \in P
\]

This is just detection
Step 2

$$\min \frac{1}{2} ||w||^2 + C \sum_{c \in C} \max(0, 1 - w \cdot \Phi(c, z)) + C \sum_{z \in W} \max(0, 1 - \max_{w \in \mathbb{R}} w \cdot \Phi(c, z))$$

Convex  Easy to solve

What about the model structure?
Can we learn that as well?

Model structure
- # components
- # parts per component
- root and part filter shapes
- part anchor locations

IniDalize a mixture model to handle different viewpoints
IniDalizaDon:

Learning model structure

Initialize a mixture model to handle different viewpoints

Initialization:
Split positives by aspect ratio

Warp to common size
Train Dalal & Triggs model for each aspect ratio on its own

Learning model structure

(a) Car component 1 (Phase 1)
(b) Car component 2 (Phase 1)
(c) Car component 1 (Phase 2)
(d) Car component 2 (Phase 2)

Use D&T filters as initial $w$ for LSVM training
Merge components
Root filter placement and component choice are latent

Summary: Model Learning

- Learn mixture model (component root filters)
- Learn part model for each mixture component
- LSVM used during both phases
Mining Hard Negatives

Typical dataset

- 300 – 8,000 positive examples
- 500 million to 1 billion negative examples (not including latent configurations)

Detections problems are highly unbalanced (many more negatives than positives). We REALLY don’t want to consider all negative examples while training!

Mining “Hard Negative” Examples

- Set of negative training examples is HUGE (a single image can yield $10^9$ examples for a scanning window classifier).
- Construct training data from the positive instances and a selection of the “hard negative” instances, via bootstrapping
  - Train using subset of negative examples
  - Apply resulting classifier to all negative examples, and take those incorrectly classified as a new set of hard negatives
  - Retrain, and perhaps repeat

PCA Analysis of HOG

- Collected 36-dimensional HOG features at a variety of scales over large number of images
- PCA analysis to identify top eigenvectors

Recall: HOG Features

- Tile window into 8 x 8 pixel cells
- Each cell represented by HOG

HoG Feature Extraction: Blocks

Independent contrast normalization over 2x2 blocks of cells.

Each cell contains a histogram of gradient orientations, weighted by gradient magnitude

Independent contrast normalization over 2x2 blocks of cells.

Each histogram gets normalized 4 different times.

9 orientations x 4 orientations = 36 feature values for each cell
PCA Analysis of HOG

- Collected 36-dimensional HOG features at a variety of scales over large number of images
- PCA analysis to identify top eigenvectors

Top 11 account for nearly all of the variation

Bounding Box Prediction

Problem: Location and size of root filter (red rectangle) may not correlate perfectly with bounding boxes in human-labeled data

Solution: Learn four linear regression functions to predict \(x_1, y_1, x_2, y_2\) of bounding box from a vector containing locations of each part (including root) and the width (scale) of the root. \((2n+3)\times 1\) (or \(y_1, x_2, y_2\))

Non-Maximum Suppression

- Sliding window detectors tend to produce many overlapping detections on same object

Greedy solution: run through list of detections in descending order of confidence, removing ones that overlap more than 50% with a higher-scoring detection.
Part III: PASCAL Challenge

- ~10,000 images, with ~25,000 target objects
  - Objects from 20 categories (person, car, bicycle, cow, table...)
  - Objects are annotated with labeled bounding boxes

Example Annotations from the Dataset

Bounding box overlap of 50% needed, to detect

Formalizing the object detection task

Many possible ways, this one is popular:

Input
Desired output

Performance summary:
Average Precision (AP)
0 is worst 1 is perfect

Learned Models

Example Results
Summary: DPM Innovations

- Star-structured graph model to represent body parts and their geometric relations
- Mixtures of models to handle large variations in viewpoint / pose
- Latent-SVM formulation
- Data mining of hard negative examples
- PCA on HOG features for dimension reduction
- Performance on PASCAL challenge dataset

Overall Results

- 9 systems competed in the 2007 challenge
- Out of 20 classes we get:
  - First place in 10 classes
  - Second place in 6 classes
- Some statistics:
  - It takes ~2 seconds to evaluate a model in one image
  - It takes ~3 hours to train a model
  - MUCH faster than most systems