Deformable part models

Ross Girshick
UC Berkeley

CS231B Stanford University
Guest Lecture April 16, 2013
Image understanding

Snack time in the lab

photo by “thomas pix” http://www.flickr.com/photos/thomaspix/2591427106
What objects are where?

robot: “I see a table with twinkies, pretzels, fruit, and some mysterious chocolate things...”
DPM lecture overview

Part 1: modeling

Part 2: learning
Formalizing the object detection task

Many possible ways
Formalizing the object detection task

Many possible ways, this one is popular:

- cat,
- dog,
- chair,
- cow,
- person,
- motorbike,
- car,
- ...
Formalizing the object detection task

Many possible ways, this one is popular:

cat,
dog,
chair,
cow,
person,
motorbike,
car,
...

Performance summary:

Average Precision (AP)
0 is worst    1 is perfect
Benchmark datasets

PASCAL VOC 2005 – 2012
- 54k objects in 22k images
- 20 object classes
- annual competition
Benchmark datasets

PASCAL VOC 2005 – 2012
- 54k objects in 22k images
- 20 object classes
- annual competition
Reduction to binary classification

\[
\text{pos} = \{ \ldots \} \\
\text{neg} = \{ \ldots \quad \text{background patches} \quad \ldots \} \\
\]

HOG \rightarrow SVM \rightarrow “Sliding window” detector

Dalal & Triggs (CVPR’05)
Sliding window detection

- Compute HOG of the whole image at multiple resolutions
- Score every subwindow of the feature pyramid
- Apply non-maxima suppression

\[ \text{score}(l, p) = \mathbf{w} \cdot \phi(l, p) \]
Detection

number of locations $p \sim 250,000$ per image
Detection

number of locations $p \sim 250,000$ per image

test set has $\sim 5000$ images

$\gg 1.3 \times 10^9$ windows to classify
Detection

number of locations $p \sim 250,000$ per image

test set has $\sim 5000$ images

$>> 1.3 \times 10^9$ windows to classify

typically only $\sim 1,000$ true positive locations
Detection

number of locations $p \sim 250,000$ per image

test set has $\sim 5000$ images

$\gg 1.3 \times 10^9$ windows to classify

typically only $\sim 1000$ true positive locations

*Extremely unbalanced binary classification*
Dalal & Triggs detector on INRIA

- AP = 75%
- (79% in my implementation)
- Very good
- Declare victory and go home?
Dalal & Triggs on PASCAL VOC 2007

AP = 12%
(using my implementation)
How can we do better?

Revisit an old idea: part-based models ("pictorial structures")
- Fischler & Elschlager '73, Felzenszwalb & Huttenlocher '00

Combine with modern features and machine learning
Part-based models

- Parts — local appearance templates
- “Springs” — spatial connections between parts (geom. prior)

Image: [Felzenszwalb and Huttenlocher 05]
Part-based models

• Local appearance is easier to model than the global appearance
  - Training data shared across deformations
  - “part” can be local or global depending on resolution
• Generalizes to previously unseen configurations
General formulation

$$G = (V, E)$$

$$V = (v_1, \ldots, v_n) \quad E \subseteq V \times V$$

$$(p_1, \ldots, p_n) \in P^n$$

part locations in the image
(or feature pyramid)
Part configuration score function

\[
\text{score}(p_1, \ldots, p_n) = \sum_{i=1}^{n} m_i(p_i) - \sum_{(i,j) \in E} d_{ij}(p_i, p_j)
\]

Part match scores

Highest scoring configurations
Part configuration score function

\[
\text{score}(p_1, \ldots, p_n) = \sum_{i=1}^{n} m_i(p_i) - \sum_{(i,j) \in E} d_{ij}(p_i, p_j)
\]

- **Objective:** maximize score over \( p_1, \ldots, p_n \)
- \( h^n \) configurations! (\( h = |P| \), about 250,000)
- **Dynamic programming**
  - If \( G = (V,E) \) is a tree, \( O(nh^2) \) general algorithm
    - \( O(nh) \) with some restrictions on \( d_{ij} \)
Star-structured deformable part models

test image

“star” model

root part
detection
Recall the Dalal & Triggs detector

\[
score(l, p) = w \cdot \phi(l, p)
\]

- HOG feature pyramid
- Linear filter / sliding-window detector
- SVM training to learn parameters \( w \)
- Add parts to the Dalal & Triggs detector
  - HOG features
  - Linear filters / sliding-window detector
  - Discriminative training

D&T + parts

Image pyramid

HOG feature pyramid

$p_0$

Z

[FMR CVPR’08]
[FGMR PAMI’10]
Abstract

This paper describes a discriminatively trained, multiscale, deformable part model for object detection. Our system achieves a two-fold improvement in average precision over the best performance in the 2006 PASCAL person detection challenge. It also outperforms the best results in the 2007 challenge in ten out of twenty categories. The system relies heavily on deformable parts, while deformable part models have become quite popular, their value had not been demonstrated on difficult benchmarks such as the PASCAL challenge. Our system also relies heavily on new methods for discriminative training. We combine a margin-sensitive approach for data mining hard negative examples with a formalism we call latent SVM. A latent SVM, like a hidden CRF, leads to a non-convex training problem. However, a latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. We believe that our training methods will eventually make possible the effective use of more latent information such as hierarchical grammar models and models involving latent three dimensional pose.

1. Introduction

We consider the problem of detecting and localizing object categories such as people or cars in static images. We have developed a new multiscale deformable part model for solving this problem. The models are trained using a discriminative procedure that only requires bounding box labels for the positive examples. Using these models, we implemented a detection system that is both highly efficient and accurate, processing an image in about 4 seconds and achieving recognition rates that are significantly better than previous systems.

Our system achieves a two-fold improvement in average precision over the winning system [7] in the 2006 PASCAL person detection challenge. The system also outperforms the best results in the 2007 challenge in ten out of twenty categories.

This material is based upon work supported by the National Science Foundation under Grant No. 05-2756042 and 05-2757396.

Figure 3: Example detection obtained with the person model. The model is defined by a coarse template, several higher resolution part templates and a spatial model for the location of each part.

The notion that objects can be modeled by parts in a deformable configuration provides an elegant framework for representing object categories [3–5, 8, 32, 34, 35, 37, 38, 44]. While these models are appealing from a conceptual point of view, it has been difficult to establish their value in practice. On difficult datasets, deformable models are often outperformed by “conceptually weaker” models such as rigid templates [7] or bag-of-features [45]. One of our main goals is to address this performance gap.

Our models include both a coarse global template covering an entire object and higher resolution part templates. The templates represent histogram of gradient features [7]. As in [36, 37, 43], we train models discriminatively. However, our system is semi-supervised, trained with a max-margin framework, and does not rely on feature detection.

We also describe a simple and effective strategy for learning parts from weakly-labeled data. In contrast to computationally demanding approaches such as [6], we can learn a model in 5 hours on a single CPU.

Another contribution of our work is a new methodology for discriminative training. We generalize SVMs for handling latent variables such as part positions, and introduce a new method for data mining “hard negative” examples during training. We believe that handling partially labeled data is a significant issue in machine learning for computer vision. For example, the PASCAL dataset only specifies a part category collection.

Sliding window DPM score function

\[ z = (p_1, \ldots, p_n) \]

\[ \text{score}(l, p_0) = \max_{p_1, \ldots, p_n} \sum_{i=0}^{n} m_i(l, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i) \]

Filter scores \quad \text{Spring costs}
Detection in a slide

- Test image
- Feature map
- Feature map at 2x resolution
- Model

- Root filter
- Responses of part filters
- Transformed responses
- Color encoding of filter response values

- Detection scores for each root location
What are the parts?
General philosophy: enrich models to better represent the data
Mixture models

Data driven: aspect, occlusion modes, subclasses

FMR CVPR ’08: AP = 0.27 (person)
FGMR PAMI ’10: AP = 0.36 (person)
Pushmi–pullyu?

Good generalization properties on Doctor Dolittle’s farm

\[
\left(\begin{array}{c}
\text{horse} \\
\end{array}\right) + \left(\begin{array}{c}
\text{horse} \\
\end{array}\right) / 2 = \left(\begin{array}{c}
\text{Pushmi–pullyu} \\
\end{array}\right)
\]

This was supposed to detect horses.
Latent orientation

Unsupervised left/right orientation discovery

FGMR PAMI ’10: AP = 0.36 (person)
voc-release5: AP = 0.45 (person)

Publicly available code for the whole system: current voc-release5
Summary of results

[DT'05] AP 0.12

[FGMR'10] AP 0.36

[FGM voc-release5] AP 0.45

[GFM'11] AP 0.49
Part 2: DPM parameter learning

fixed model *structure*

component 1

component 2
Part 2: DPM parameter learning

fixed model structure

component 1

component 2

training images

$y$

$+1$
Part 2: DPM parameter learning

fixed model structure

component 1

component 2

training images

\( y \)

\( +1 \)

\( -1 \)
Part 2: DPM parameter learning

Parameters to learn:
- biases (per component)
- deformation costs (per part)
- filter weights
Linear parameterization

\[ z = (p_1, \ldots, p_n) \]

\[
\text{score}(l, p_0) = \max_{p_1, \ldots, p_n} \sum_{i=0}^{n} m_i(l, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i)
\]

Filter scores \[ m_i(l, p_i) = w_i \cdot \phi(l, p_i) \]

Spring costs \[ d_i(p_0, p_i) = d_i \cdot (dx^2, dy^2, dx, dy) \]

\[
\text{score}(l, p_0) = \max_{z} w \cdot \Phi(l, (p_0, z))
\]
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 HOG pyramid location $p_0$

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 $p_0$ and allows any placement of the other filters
Typical dataset

300 – 8,000 positive examples

500 million to 1 billion negative examples
(not including latent configurations!)

Large-scale*

*unless someone from google is here
How we learn parameters: latent SVM

\[ E(w) = \frac{1}{2} \|w\|^2 + C \sum_i \max\{0, 1 - y_i f_w(x_i)\} \]
How we learn parameters: latent SVM

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

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

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

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

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

\[ + C \sum_{i \in N} \max \{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\} \]

+ score

convex


How we learn parameters: latent SVM

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

\[ E(w) = \frac{1}{2}\|w\|^2 + C \sum_{i \in P} \max\{0, 1 - \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\} + C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\} \]
Latent SVM objective is convex in the negatives but not in the positives

>> “semi-convex”
Convex upper bound on loss

\[
\max\{0, 1 - \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
\]

convex

\[
\max\{0, 1 - w \cdot \Phi(x_i, Z_{Pi})\}
\]
Auxiliary objective

Let $Z_P = \{Z_{P1}, Z_{P2}, \ldots \}$

$$E(w, Z_P) = \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{Pi})\}$$

$$+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}$$
Auxiliary objective

Let $Z_P = \{Z_{P1}, Z_{P2}, \ldots \}$

$$
E(w, Z_P) = \frac{1}{2}\|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{Pi})\} \\
+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
$$

Note that $E(w, Z_p) \geq \min_{Z_P} E(w, Z_P) = E(w)$
Auxiliary objective

Let $Z_P = \{Z_{P1}, Z_{P2}, \ldots \}$

$$E(w, Z_P) = \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{Pi})\}$$

$$+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}$$

Note that $E(w, Z_P) \geq \min_{Z_P} E(w, Z_P) = E(w)$

and $w^* = \min_{w, Z_P} E(w, Z_P) = \min_{w} E(w)$
Auxiliary objective

\[ w^* = \min_{w, Z_P} E(w, Z_P) = \min_{w} E(w) \]

This isn’t any easier to optimize
Auxiliary objective

\[ w^* = \min_{w, Z_P} E(w, Z_P) = \min_w E(w) \]

This isn’t any easier to optimize

Find stationary point by coordinate descent on \( E(w, Z_P) \)
Auxiliary objective

\[ w^* = \min_{w, Z_P} E(w, Z_P) = \min_w E(w) \]

This isn’t any easier to optimize

Find stationary point by coordinate descent on \( E(w, Z_P) \)

Initialization: either by picking a \( w(0) \) (or \( Z_P \))
Auxiliary objective

\[ \mathbf{w}^* = \min_{\mathbf{w}, Z_P} E(\mathbf{w}, Z_P) = \min_{\mathbf{w}} E(\mathbf{w}) \]

This isn’t any easier to optimize

Find stationary point by coordinate descent on \( E(\mathbf{w}, Z_P) \)

Initialization: either by picking a \( \mathbf{w}(0) \) (or \( Z_P \))

Step 1:

\[ Z_{Pi} = \arg\max_{\mathbf{w}(t)} \mathbf{w}(t) \cdot \Phi(x_i, z) \quad \forall i \in P \]
Auxiliary objective

\[ w^* = \min_{w, Z_P} E(w, Z_P) = \min_{w} E(w) \]

This isn’t any easier to optimize

Find stationary point by coordinate descent on \( E(w, Z_P) \)

Initialization: either by picking a \( w(0) \) (or \( Z_P \))

Step 1:
\[ Z_{Pi} = \arg\max_{z \in Z(x_i)} w(t) \cdot \Phi(x_i, z) \quad \forall i \in P \]

Step 2:
\[ w(t+1) = \arg\min_{w} E(w, Z_P) \]
Step 1

\[ Z_{Pi} = \arg\max_w w(t) \cdot \Phi(x_i, z) \quad \forall i \in P \]

This is just detection:
Step 2

$$
\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i \in P} \max \{0, 1 - \mathbf{w} \cdot \Phi(x_i, Z_{pi})\}
$$

$$
+ C \sum_{i \in N} \max \{0, 1 + \max_{z \in Z(x)} \mathbf{w} \cdot \Phi(x_i, z)\}
$$

Convex
Step 2

\[
\min_w \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{P_i})\} \\
+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
\]

Convex

Similar to a structural SVM
Step 2

\[
\min_w \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{P_i})\} \\
+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
\]

Convex

Similar to a structural SVM

But, recall 500 million to 1 billion negative examples!
Step 2

\[
\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in P} \max \left\{ 0, 1 - \mathbf{w} \cdot \Phi(x_i, Z_{P_i}) \right\} \\
+ C \sum_{i \in N} \max \left\{ 0, 1 + \max_{z \in Z(x)} \mathbf{w} \cdot \Phi(x_i, z) \right\}
\]

Convex

Similar to a structural SVM

But, recall 500 million to 1 billion negative examples!

Can be solved by a working set method

- “bootstrapping”
- “data mining”
- “constraint generation”
- requires a bit of engineering to make this fast
Latent SVM is mathematically equivalent to MI-SVM (Andrews et al. NIPS 2003)

Latent SVM can be written as a latent structural SVM (Yu and Joachims ICML 2009)
– natural optimization algorithm is concave-convex procedure
– similar to, but not exactly the same as, coordinate descent
What about the model structure?

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

Split positives by aspect ratio

Warp to common size

Train Dalal & Triggs model for each aspect ratio on its own
Learning model structure

Use D&T filters as initial $\mathbf{w}$ for LSVM training

Merge components

Root filter placement and component choice are latent
Learning model structure

Add parts to cover high-energy areas of root filters

Continue training model with LSVM
Learning model structure

without orientation clustering

(a) Car component 1 (Phase 1)  (b) Car component 2 (Phase 1)  (c) Car comp. 3 (Phase 1)

with orientation clustering

(a) Car component 1  (b) Car component 2  (c) Car component 3
Learning model structure

In summary
- repeated application of LSVM training to models of increasing complexity
- structure learning involves many heuristics (and vision insight!)