Recognition continued: discriminative classifiers

Tues Nov 17
Kristen Grauman
UT Austin

Announcements

• A5 out today, due Dec 2
Previously

• Supervised classification
• Window-based generic object detection
  – basic pipeline
  – boosting classifiers
  – face detection as case study
• Hidden Markov Models

Review questions

• Why is it more efficient to extract Viola-Jones-style rectangular filter responses at multiple scales, vs. extract typical convolution filter responses at multiple scales?
• What does it mean to be a “weak” classifier?
• For a classifier cascade used for object detection, what properties do we require the early vs. later classifiers (stages) in the cascade to have?
Today

- Sliding window object detection wrap-up
  - Attentional cascade
  - Pros and cons
  - Object proposals for detection
- Supervised classification continued
  - Nearest neighbors
  - HMM example
  - Support vector machines

Considering all possible filter parameters: position, scale, and type:

180,000+ possible features associated with each 24 x 24 window

Which subset of these features should we use to determine if a window has a face?

Use AdaBoost both to select the informative features and to form the classifier
Viola-Jones detector: AdaBoost

- Want to select the single rectangle feature and threshold that best separates **positive** (faces) and **negative** (non-faces) training examples, in terms of **weighted error**.

Resulting weak classifier:

\[
\begin{align*}
    h_t(x) &= \begin{cases} 
        +1 & \text{if } f_t(x) > \theta_t \\
        -1 & \text{otherwise}
    \end{cases}
\end{align*}
\]

For next round, reweight the examples according to errors, choose another filter/threshold combo.

Cascading classifiers for detection

- Form a *cascade* with low false negative rates early on
- Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative
Training the cascade

- Set target detection and false positive rates for each stage
- Keep adding features to the current stage until its target rates have been met
  - Need to lower AdaBoost threshold to maximize detection (as opposed to minimizing total classification error)
  - Test on a validation set
- If the overall false positive rate is not low enough, then add another stage
- Use false positives from current stage as the negative training examples for the next stage

Viola-Jones detector: summary

Train with 5K positives, 350M negatives
Real-time detector using 38 layer cascade
6061 features in all layers

[Implementation available in OpenCV]
Viola-Jones detector: summary

- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - **Integral images** for fast feature evaluation
  - **Boosting** for feature selection
  - **Attentional cascade** of classifiers for fast rejection of non-face windows


---

Boosting: pros and cons

- **Advantages of boosting**
  - Integrates classification with feature selection
  - Complexity of training is linear in the number of training examples
  - Flexibility in the choice of weak learners, boosting scheme
  - Testing is fast
  - Easy to implement

- **Disadvantages**
  - Needs many training examples
  - Other discriminative models may outperform in practice (SVMs, CNNs, ...)
    - especially for many-class problems
Viola-Jones Face Detector: Results

Window-based detection: strengths

- Sliding window detection and global appearance descriptors:
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes
Window-based detection: Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

Limitations (continued)

- Not all objects are “box” shaped
Limitations (continued)

- Non-rigid, deformable objects not captured well with representations assuming a fixed 2D structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions

Limitations (continued)

- If considering windows in isolation, context is lost

Fig: Credit: Derek Hoiem
Limitations (continued)

- In practice, often entails large, cropped training set (expensive)
- Requiring good match to a global appearance description can lead to sensitivity to partial occlusions

When will sliding window face detection work best?

Class photos
What other categories are amenable to window-based representation?

Pedestrian detection

- Detecting upright, walking humans also possible using sliding window’s appearance/texture; e.g.,

  - SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]
  - Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]
  - SVM with HoGs [Dalal & Triggs, CVPR 2005]
Recap so far

- Basic pipeline for window-based detection
  - Model/representation/classifier choice
  - Sliding window and classifier scoring
- Boosting classifiers: general idea
- Viola-Jones face detector
  - Exemplar of basic paradigm
  - Plus key ideas: rectangular features, Adaboost for feature selection, cascade
- Pros and cons of window-based detection

Object proposals

**Main idea:**
- Learn to generate category-independent regions/boxes that have object-like properties.
- Let object detector search over “proposals”, not exhaustive sliding windows

Alexe et al. Measuring the objectness of image windows, PAMI 2012
Object proposals

Alexe et al. Measuring the objectness of image windows, PAMI 2012
Object proposals

More proposals

Alexe et al. Measuring the objectness of image windows, PAMI 2012

Region-based object proposals

Object proposals: Several related formulations

- Alexe et al. Measuring the objectness of image windows, PAMI 2012

Today

- Sliding window object detection wrap-up
  - Attentional cascade
  - Pros and cons
  - Object proposals for detection
- Supervised classification continued
  - Nearest neighbors
  - HMM example
  - Support vector machines
Window-based models: Three case studies

- Boosting + face detection
  Viola & Jones

- NN + scene Gist classification
  e.g., Hays & Efros

- SVM + person detection
  e.g., Dalal & Triggs

Nearest Neighbor classification

- Assign label of nearest training data point to each test data point

  Voronoi partitioning of feature space for 2-category 2D data

  Black = negative
  Red = positive

  Novel test example
  Closest to a positive example from the training set, so classify it as positive.
K-Nearest Neighbors classification

- For a new point, find the $k$ closest points from training data
- Labels of the $k$ points “vote” to classify

If query lands here, the $5$ NN consist of $3$ negatives and $2$ positives, so we classify it as negative.

A nearest neighbor recognition example
Where in the World?


Where in the World?
Where in the World?

6+ million geotagged photos by 109,788 photographers

Annotated by Flickr users
6+ million geotagged photos by 109,788 photographers

Annotated by Flickr users

Which scene properties are relevant?
Spatial Envelope Theory of Scene Representation
Oliva & Torralba (2001)

A scene is a single surface that can be represented by global (statistical) descriptors

Global texture: capturing the “Gist” of the scene

Capture global image properties while keeping some spatial information

\[ V = \{ \text{energy at each orientation and scale} \} = 6 \times 4 \text{ dimensions} \]

Gist descriptor

Oliva & Torralba IJCV 2001, Torralba et al. CVPR 2003
Which scene properties are relevant?

- **Gist scene descriptor**
- **Color Histograms** - L*A*B* 4x14x14 histograms
- **Texton Histograms** – 512 entry, filter bank based
- **Line Features** – Histograms of straight line stats

Im2gps: Scene Matches

Im2gps: Scene Matches

Scene Matches

Quantitative Evaluation Test Set
The Importance of Data


Nearest neighbors: pros and cons

- **Pros:**
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data

- **Cons:**
  - Large search problem to find nearest neighbors
  - Storage of data
  - Must know we have a meaningful distance function

Kristen Grauman
HMM example: Photo Geo-location

Where was this picture taken?

Example: Photo Geo-location

Where was this picture taken?
Example: Photo Geo-location

Where was this picture taken?

Example: Photo Geo-location

Where was each picture in this sequence taken?
Idea: Exploit the beaten path

• Learn dynamics model from “training” tourist photos
• Exploit timestamps and sequences for novel “test” photos

[Chen & Grauman CVPR 2011]
Hidden Markov Model

\[ P(\text{Observation} \mid \text{State}) \]

\[ P(\text{State}) \]

\[ P(S_2 \mid S_1) \quad P(S_1 \mid S_2) \quad P(S_1 \mid S_1) \quad P(S_2 \mid S_2) \quad P(S_3 \mid S_2) \quad P(S_2 \mid S_3) \quad P(S_3 \mid S_3) \quad P(S_1 \mid S_3) \quad P(S_3 \mid S_1) \]

Discovering a city’s locations

Define states with data-driven approach:

mean shift clustering on the GPS coordinates of the training images
Observation model

\[
P(\text{Observation} \mid \text{State}) = P(\text{Observation} \mid \text{Liberty Island})
\]

A burst \( B \) in test image sequence

\( M \) retrieved training images
Location estimation accuracy

<table>
<thead>
<tr>
<th></th>
<th>NN</th>
<th>Img-HMM</th>
<th>Burst Only</th>
<th>Burst-HMM (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg/seq</td>
<td>0.1502</td>
<td>0.1608</td>
<td>0.1764</td>
<td><strong>0.2036</strong></td>
</tr>
<tr>
<td>Overall</td>
<td>0.1592</td>
<td>0.1660</td>
<td>0.2617</td>
<td><strong>0.2782</strong></td>
</tr>
</tbody>
</table>

(a) Rome dataset

<table>
<thead>
<tr>
<th></th>
<th>NN</th>
<th>Img-HMM</th>
<th>Burst Only</th>
<th>Burst-HMM (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg/seq</td>
<td>0.2323</td>
<td>0.2124</td>
<td>0.2099</td>
<td><strong>0.3021</strong></td>
</tr>
<tr>
<td>Overall</td>
<td>0.2302</td>
<td>0.2070</td>
<td>0.2055</td>
<td><strong>0.3143</strong></td>
</tr>
</tbody>
</table>

(b) New York City dataset

Qualitative Result – New York
Discovering travel guides’ beaten paths
Routes from travel guide book for New York vs.
Random walks in learned HMM

<table>
<thead>
<tr>
<th></th>
<th>Rand. Walk</th>
<th>Rand. Walk(TS)</th>
<th>Guidebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Prob.</td>
<td>$6.3 \cdot 10^{-12}$</td>
<td>$4.2 \cdot 10^{-11}$</td>
<td>$2.0 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

Video textures

- Schodl, Szeliski, Salesin, Essa; Siggraph 2000.
- [http://www.cc.gatech.edu/cpl/projects/videotexture/](http://www.cc.gatech.edu/cpl/projects/videotexture/)
Today

- Sliding window object detection wrap-up
  - Attentional cascade
  - Pros and cons
  - Object proposals for detection
- Supervised classification continued
  - Nearest neighbors
  - HMM example
    - Support vector machines

Window-based models: Three case studies

- Boosting + face detection
  - Viola & Jones
- NN + scene Gist classification
  - e.g., Hays & Efros
- SVM + person detection
  - e.g., Dalal & Triggs
Linear classifiers

- Find linear function to separate positive and negative examples

\[ x_i \text{ positive: } x_i \cdot w + b \geq 0 \]
\[ x_i \text{ negative: } x_i \cdot w + b < 0 \]

Which line is best?
Support Vector Machines (SVMs)

- Discriminative classifier based on optimal separating line (for 2d case)
- Maximize the \textit{margin} between the positive and negative training examples

Support vector machines

- Want line that maximizes the margin.

\begin{align*}
    x_i \text{ positive } (y_i = 1): & \quad x_i \cdot w + b \geq 1 \\
    x_i \text{ negative } (y_i = -1): & \quad x_i \cdot w + b \leq -1 \\

    \text{For support vectors,} \quad & \quad x_i \cdot w + b = \pm 1
\end{align*}

C. Burges, \textit{A Tutorial on Support Vector Machines for Pattern Recognition}, Data Mining and Knowledge Discovery, 1998
Support vector machines

- Want line that maximizes the margin.

For support vectors, \( x_i \cdot w + b = \pm 1 \)

Distance between point and line:
\[
\frac{|x_i \cdot w + b|}{\|w\|}
\]

For support vectors:
\[
\frac{w^T x + b}{\|w\|} = \frac{\pm 1}{\|w\|} \quad M = \frac{1}{\|w\|} \cdot \frac{-1}{\|w\|} = \frac{2}{\|w\|}
\]

Therefore, the margin is \( 2 / \|w\| \)
Finding the maximum margin line

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

Quadratic optimization problem:

\[
\begin{align*}
\text{Minimize} & \quad \frac{1}{2} w^T w \\
\text{Subject to} & \quad y_i(w \cdot x_i + b) \geq 1
\end{align*}
\]

Finding the maximum margin line

- Solution: $w = \sum_i \alpha_i y_i x_i$

learned weight Support vector
Finding the maximum margin line

- Solution: \( w = \sum_i \alpha_i y_i x_i \)
  \[ b = y_i - w \cdot x_i \quad \text{(for any support vector)} \]
  \[ w \cdot x + b = \sum_i \alpha_i y_i x_i \cdot x + b \]

- Classification function:

\[
\begin{align*}
    f(x) &= \text{sign} \left( w \cdot x + b \right) \\
    &= \text{sign} \left( \sum_i \alpha_i y_i x_i \cdot x + b \right)
\end{align*}
\]

If \( f(x) < 0 \), classify as negative, if \( f(x) > 0 \), classify as positive

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery

---

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alpes, 655 avenue de l’Europe, Montbonnot 38334, France

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1000 annotated human images with a large range of pose variations and backgrounds.

1 Introduction

We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5-6. The main conclusions are summarized in §7.

2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18, 17, 22, 16, 20]. See [6] for a survey. Papageorgiou et al. [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Despertiere et al. give an optimized version of this [2]. Gavrila & Philomin [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [7]. Viola et al. [22] build an efficient

• CVPR 2005
HoG descriptor

• Map each grid cell in the input window to a histogram counting the gradients per orientation.
• Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Person detection with HoG’s & linear SVM’s

Dalal & Triggs, CVPR 2005
Person detection with HoGs & linear SVMs

• Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005

Questions

• What if the data is not linearly separable?
Non-linear SVMs

- Datasets that are linearly separable with some noise work out great:

- But what are we going to do if the dataset is just too hard?

- How about… mapping data to a higher-dimensional space:

Non-linear SVMs: feature spaces

- General idea: the original input space can be mapped to some higher-dimensional feature space where the training set is separable:

Slide from Andrew Moore’s tutorial: http://www.autonlab.org/tutorials/svm.html
Nonlinear SVMs

- The kernel trick: instead of explicitly computing the lifting transformation $\varphi(x)$, define a kernel function $K$ such that
  \[ K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) \]

- This gives a nonlinear decision boundary in the original feature space:
  \[ \sum_i \alpha_i y_i K(x_i, x) + b \]

"Kernel trick": Example

2-dimensional vectors $x = [x_1 \ x_2]$;
let $K(x_i, x_j) = (1 + x_i^T x_j)^2$

Need to show that $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$:

\[
K(x_i, x_j) = (1 + x_i^T x_j)^2,
= 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2}
= \begin{bmatrix} 1 & x_{i1}^2 & \sqrt{2} x_{i1} x_{i2} & x_{i2}^2 & \sqrt{2} x_{i1} \sqrt{2} x_{i2} \\ 1 & x_{j1}^2 & \sqrt{2} x_{j1} x_{j2} & x_{j2}^2 & \sqrt{2} x_{j1} \sqrt{2} x_{j2} \end{bmatrix}^T
\]

where $\varphi(x) = [1 \ x_1^2 \ \sqrt{2} x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2]$
Examples of kernel functions

- Linear: \[ K(x_i, x_j) = x_i^T x_j \]

- Gaussian RBF: \[ K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \]

- Histogram intersection:

\[
K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))
\]

SVMs for recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples.
4. Use this “kernel matrix” to solve for SVM support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.
Questions

• What if the data is not linearly separable?
• What if we have more than just two categories?

Multi-class SVMs

• Achieve multi-class classifier by combining a number of binary classifiers

• One vs. all
  – Training: learn an SVM for each class vs. the rest
  – Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value

• One vs. one
  – Training: learn an SVM for each pair of classes
  – Testing: each learned SVM “votes” for a class to assign to the test example

Kristen Grauman
SVMs: Pros and cons

• Pros
  • Kernel-based framework is very powerful, flexible
  • Often a sparse set of support vectors – compact at test time
  • Work very well in practice, even with small training sample sizes

• Cons
  • No “direct” multi-class SVM, must combine two-class SVMs
  • Can be tricky to select best kernel function for a problem
  • Computation, memory
    – During training time, must compute matrix of kernel values for every pair of examples
    – Learning can take a very long time for large-scale problems

Summary

• Object recognition as classification task
  • Boosting (face detection ex)
  • Support vector machines and HOG (person detection ex)
  • Nearest neighbors and global descriptors (scene rec ex)

• Sliding window search paradigm
  • Pros and cons
  • Speed up with attentional cascade
  • Object proposals as alternative to exhaustive search

• HMM examples