Recognition continued: discriminative classifiers

Tues April 18
Kristen Grauman
UT Austin

Last time

• Introduction to object categorization
• Window-based generic object detection
  – basic pipeline
  – boosting classifiers
  – face detection as case study

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
  - Applications / examples
  - Pros and cons
- Supervised classification continued
  - Nearest neighbors
  - Support vector machines

Recall: Viola-Jones face detector

Main ideas:
- Represent local texture with efficiently computable “rectangular” features within window of interest
- Select discriminative features to be weak classifiers
- Use boosted combination of them as final classifier
- Form a cascade of such classifiers, rejecting clear negatives quickly

Viola-Jones detector: features

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.

  ![Diagram of Viola-Jones detector](image)

  **Resulting weak classifier:**
  
  \[ h(x) = \begin{cases} 
  +1 & \text{if } f(x) > 0 \\
  -1 & \text{otherwise} 
  \end{cases} \]

  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

![Diagram of cascading classifiers](image)

**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

- A seminal approach to real-time object detection
  - 15,000 citations and counting
- 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

P. Viola and M. Jones. Robust real-time face detection. IJCV 57(2), 2004.

Viola-Jones Face Detector: Results
Viola-Jones Face Detector: Results

Detecting profile faces?

Can we use the same detector?
Viola-Jones Face Detector: Results

Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006. http://www.robots.ox.ac.uk/~vgg/research/face/index.html
Google street view blurs face of cow to protect its identity

Consumer application: iPhoto

http://www.apple.com/ilife/iphoto/

Consumer application: iPhoto

Things iPhoto thinks are faces
Consumer application: iPhoto
Can be trained to recognize pets!

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 [Nistér, Jones & Stew, ICCV 2003]
  - SVM with Haar [Dalal & Triggs, CVPR 2005]
Use rectangular features, select good features to distinguish the chest from non-chests with Adaboost.

Perform identification by matching the pattern of spots to a database of known penguins.
Privacy Gift Shop – CV Dazzle

http://www.wired.com/2015/06/facebook-can-recognize-even-dont-show-face/
Wired, June 15, 2015
Slide credit: Kristen Grauman

Privacy Visor

Slide credit: Kristen Grauman

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

Slide credit: Lana Lazebnik
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

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
Summary 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

Today

- Sliding window object detection wrap-up
  - Attentional cascade
  - Applications / examples
  - Pros and cons
- Supervised classification continued
  - Nearest neighbors
  - Support vector machines
Window-based models: Three case studies

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

Nearest Neighbor classification

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

K-Nearest Neighbors classification

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

Novel test example

- Closest to a positive example from the training set, so classify it as positive.

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

K = 5

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?
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

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

<table>
<thead>
<tr>
<th>Database size (thousands of images, log scale)</th>
<th>Percentage of Geolocations within 200km</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09</td>
<td>2.1</td>
</tr>
<tr>
<td>0.38</td>
<td>3.8</td>
</tr>
<tr>
<td>1.54</td>
<td>7.6</td>
</tr>
<tr>
<td>6.16</td>
<td>12.9</td>
</tr>
<tr>
<td>24.6</td>
<td>18.2</td>
</tr>
<tr>
<td>98.5</td>
<td>23.0</td>
</tr>
<tr>
<td>394</td>
<td>31.5</td>
</tr>
<tr>
<td>1,576</td>
<td>36.7</td>
</tr>
<tr>
<td>6,304</td>
<td>40.6</td>
</tr>
</tbody>
</table>

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

Today

• Sliding window object detection wrap-up
  • Attentional cascade
  • Applications / examples
  • Pros and cons

• Supervised classification continued
  • Nearest neighbors
  • Support vector machines

Window-based models: Three case studies

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

- Find linear function to separate positive and negative examples

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

Which line is best?

Support Vector Machines (SVMs)

- Discriminative classifier based on optimal separating line (for 2d case)
- Maximize the margin between the positive and negative training examples
Support vector machines

- Want line that maximizes the margin.

\[ x_+, y_+ = 1 \]
\[ x_-, y_+ = -1 \]

For support vectors, \[ x \cdot w + b = \pm 1 \]

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

For support vectors:
\[ \frac{1}{||w||} \]

Margin \( M \)

Therefore, the margin is \( \frac{2}{||w||} \)

Finding the maximum margin line

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

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: $w = \sum_{i=1}^{m} \alpha_i y_i x_i$

    learned weight Support vector

- $b = y_i - w \cdot x_i$ (for any support vector)

- $w \cdot x + b = \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x + b$

- Classification function:
  $$f(x) = \text{sign} (w \cdot x + b)$$
  $$= \text{sign} \left( \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x + b \right)$$

  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

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

Person detection with HoG's & linear SVM's

- 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 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:

$$\Phi: x \rightarrow \phi(x)$$

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 $\phi(x)$, define a kernel function $K$ such that

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum \alpha_i y_i K(x_i, x) + b$$

“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 $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$:

$$K(x_i, x_j) = (1 + x_i^T x_j)^2$$

$$= 1 + x_i^2 x_j^2 + 2 x_i^T x_j + x_i^2 x_j^2 + 2 x_i x_j + 2 x_i x_j$$

$$= \begin{bmatrix} 1 & x_i^2 \sqrt{2} & x_i x_j \sqrt{2} & x_j^2 \sqrt{2} x_j \end{bmatrix}^T$$

$$= \Phi(x_i)^T \Phi(x_j),$$

where $\Phi(x) = \begin{bmatrix} 1 & x_1^2 \sqrt{2} x_1 x_2 & x_2^2 \sqrt{2} x_2 \end{bmatrix}$
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

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