Appearance-based recognition & detection II
Kristen Grauman
UT-Austin
Tuesday, Nov 11

Last time
• **Appearance-based recognition**: using global appearance descriptions within a window to characterize a class.
  – Classification: basic idea of supervised learning
    • Skin color detection example
  – Sliding windows: detection via classification
    • Make a yes/no decision at every window
    • Face detection example using boosting and rectangular features [Viola-Jones 2001]

Misc notes
• Extra disk space
• SIFT extraction
  – http://www.cs.ubc.ca/~lowe/keypoints/

Today
• Additional classes well-suited by global appearance representations
• Discriminative classifiers
  – Boosting (last time)
  – Nearest neighbors
  – Support vector machines
    • Application to pedestrian detection
    • Application to gender classification

Viola-Jones Face Detector: Summary
- Train cascade of classifiers with AdaBoost
  - Train with 5K positives, 350M negatives
  - Real-time detector using 38 layer cascade
  - 6061 features in final layer
  - [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Viola-Jones Face Detector: Results
Example application: faces in photos

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

- Other classes that might work with global appearance in a window?

Penguin detection & identification

Penguin detection & identification

This project uses the Viola-Jones AdaBoost face detection algorithm to detect penguin chests, and then matches the pattern of spots to identify a particular penguin.

Given a detected chest, try to extract the whole chest for this particular penguin.

Perform identification by matching the pattern of spots to a database of known penguins.

**Penguin detection & identification**

Discriminative classifiers

- Nearest neighbor
  - 10 examples
  - Shaferovich, Viola, Darrell 2003
  - Borg, Berg, Malik 2005...

- Neural networks
  - LeCun, Bottou, Bengio, Haffner 1998
  - Rowley, Baluja, Kanade 1998...

- Support Vector Machines
  - Guyon, Vapnik
  - Heisele, Serre, Poggio, 2001, ...

- Boosting
  - Viola, Jones 2001
  - Torralba et al. 2004
  - Opelt et al. 2006, ...

- Conditional Random Fields
  - McCallum, Freitag, Pereira 2000
  - Kumar, Hebert 2003...

Slide adapted from Antonio Torralba
Today

- Additional classes well-suited by global appearance representations
- Discriminative classifiers
  - Boosting (last time)
  - Nearest neighbors
- Support vector machines
  - Application to pedestrian detection
  - Application to gender classification

Nearest Neighbor classification

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

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

K-Nearest Neighbors classification

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

Example: nearest neighbor classification

Similarly, if the video frames we were indexing in the Video Google database had labels, we could classify the query.

Example: nearest neighbor classification

Labeled database of frames from movie
Today

- Additional classes well-suited by global appearance representations
- Discriminative classifiers
  - Boosting (last time)
  - Nearest neighbors
  - Support vector machines
    - Application to pedestrian detection
    - Application to gender classification

Linear classifiers

```
Let \( w = \begin{bmatrix} a \\ c \end{bmatrix} \), \( x = \begin{bmatrix} x \\ y \end{bmatrix} \)

\[ ax + cy + b = 0 \]
```

Lines in \( \mathbb{R}^2 \)

```
Let \( w = \begin{bmatrix} a \\ c \end{bmatrix} \), \( x = \begin{bmatrix} x \\ y \end{bmatrix} \)

\[ ax + cy + b = 0 \]
```

Linear classifiers

- Find linear function to separate positive and negative examples

- 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

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

Distance from point to line:
\[
D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}}
\]

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

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

Support vector machines

- Want line that maximizes the margin.

\[
x_1 \text{ positive } (y_1 = 1): \quad x_1 \cdot w + b \geq 1 \\
-x_1 \text{ negative } (y_1 = -1): \quad x_1 \cdot w + b \leq -1
\]

Margin M

\[
M = \frac{1}{\|w\|} = \frac{1}{2}
\]
Finding the maximum margin line

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

Quadratic optimization problem:

Minimize $\frac{1}{2} w^T w$

Subject to $y_i (w \cdot x_i + b) \geq 1$ for each training point.

Note: sign trick.

Finding the maximum margin line

• Solution: $w = \sum \alpha_i y_i x_i$
• Classification function:
  \[ f(x) = \text{sign}(w \cdot x + b) \]
  \[ = \text{sign}\left(\sum \alpha_i y_i x_i \cdot x + b\right) \]
• Notice that it relies on an inner product between the test point $x$ and the support vectors $x_i$.
• (Solving the optimization problem also involves computing the inner products $x_i \cdot x_j$ between all pairs of training points.)

Questions

• What if the features are not 2d?
• What if the data is not linearly separable?
• What if we have more than just two categories?
Questions

- What if the features are not 2d?
  - Generalizes to d-dimensions – replace line with “hyperplane”
- What if the data is not linearly separable?
- What if we have more than just two categories?

Planes in $\mathbb{R}^3$

Let $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ $x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

$$ax + by + cz + d = 0$$

$$w \cdot x + d = 0$$

$$D = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} = \frac{w^T x + d}{\|w\|} \text{ distance from point to plane}$$

Hyperplanes in $\mathbb{R}^n$

Hyperplane $H$ is set of all vectors $x \in \mathbb{R}^n$ which satisfy:

$$w_1x_1 + w_2x_2 + \ldots + w_nx_n + b = 0$$

$$w^T x + b = 0$$

$$D(H, x) = \frac{w^T x + b}{\|w\|} \text{ distance from point to hyperplane}$$

Questions

- What if the features are not 2d?
- What if the data is not linearly separable?
- What if we have more than just two categories?

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-separable by a hyperplane in 2-d

Source: Bill Freeman

Slide from Andrew Moore’s tutorial: http://www.autonlab.org/tutorials/svm.html
Another example:
Separable by a hyperplane in 3-d

Source: Bill Freeman

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 $\phi(x)$, 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:

$$\sum_{i} a_i y_i K(x_i, x) + b$$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Examples of General Purpose Kernel Functions
- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polynomial of power $p$: $K(x_i, x_j) = (1+ x_i^T x_j)^p$
- Gaussian (radial-basis function network):

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

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

Questions
- What if the features are not 2d?
- 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 for recognition

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

Example: pedestrian detection with HoG’s and 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.

Example: learning gender with SVMs

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.
Moghaddam and Yang, Face & Gesture 2000.

Pedestrian detection

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

Pedestrian detection with HoG’s & SVM’s

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

- Training examples:
  - 1044 males
  - 713 females
- Experiment with various kernels, select Gaussian RBF

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

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Gender perception experiment: How well can humans do?

- Subjects:
  - 30 people (22 male, 8 female)
  - Ages mid-20’s to mid-40’s
- Test data:
  - 254 face images (6 males, 4 females)
  - Low res and high res versions
- Task:
  - Classify as male or female, forced choice
  - No time limit

Moghaddam and Yang, Face & Gesture 2000.

Classifer Performance

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>Male</td>
</tr>
<tr>
<td>SVM with RBF kernel</td>
<td>3.88%</td>
</tr>
<tr>
<td>SVM with cubic polynomial kernel</td>
<td>6.88%</td>
</tr>
<tr>
<td>Large ensemble of RBF</td>
<td>5.54%</td>
</tr>
<tr>
<td>Classical RBF</td>
<td>7.79%</td>
</tr>
<tr>
<td>Quadratic classifier</td>
<td>10.6%</td>
</tr>
<tr>
<td>Fisher linear discriminant</td>
<td>13.03%</td>
</tr>
<tr>
<td>Nearest neighbor</td>
<td>27.16%</td>
</tr>
<tr>
<td>Linear classifier</td>
<td>58.95%</td>
</tr>
</tbody>
</table>

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Support Faces

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Human vs. Machine

- SVMs performed better than any single human test subject, at either resolution

Figure 6. SVM vs. Human performance

Moghaddam and Yang, Face & Gesture 2000.
Hardest examples for humans

Top five human misclassifications

Moghaddam and Yang, Face & Gesture 2000.

SVMs: Pros and cons

• Pros
  • Many publicly available SVM packages:
    http://www.kernel-machines.org/software
    http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  • 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 very 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: today

• Additional classes well-suited by global appearance representations
• Discriminative classifiers
  – Boosting (last time)
  – Nearest neighbors
  – Support vector machines
    • Application to pedestrian detection
    • Application to gender classification