Classifier based methods for Object Recognition

CMP 719– Computer Vision
Pinar Duygulu

(Slide credits:
Kristen Grauman, Fei Fei Li, Antonio Torralba, Hames Hays)
Classifier-based methods
Classifier based methods

Object detection and recognition is formulated as a classification problem. The image is partitioned into a set of overlapping windows ... and a decision is taken at each window about if it contains a target object or not.
Learning Models

Training

Training Images

Image Features

Training Labels

Training

Learned model

Testing

Test Image

Image Features

Learned model

Prediction

Slide credit: Derek Hoiem
Supervised classification

- Given a collection of labeled examples, come up with a function that will predict the labels of new examples.

  **Training examples**  **Novel input**

  "four"  
  "nine"  

- How good is some function we come up with to do the classification?

- Depends on
  - Mistakes made
  - Cost associated with the mistakes
Supervised classification

- Given a collection of labeled examples, come up with a function that will predict the labels of new examples.

- Consider the two-class (binary) decision problem
  - $L(4 \rightarrow 9)$: Loss of classifying a 4 as a 9
  - $L(9 \rightarrow 4)$: Loss of classifying a 9 as a 4

- **Risk** of a classifier $s$ is expected loss:

$$R(s) = \Pr(4 \rightarrow 9 \mid \text{using } s)L(4 \rightarrow 9) + \Pr(9 \rightarrow 4 \mid \text{using } s)L(9 \rightarrow 4)$$

- We want to choose a classifier so as to minimize this total risk
Supervised classification

Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

So, best decision boundary is at point $x$ where

$$P(\text{class is } 9 \mid x) L(9 \rightarrow 4) = P(\text{class is } 4 \mid x) L(4 \rightarrow 9)$$

To classify a new point, choose class with lowest expected loss; i.e., choose “four” if

$$P(4 \mid x) L(4 \rightarrow 9) > P(9 \mid x) L(9 \rightarrow 4)$$
Supervised classification

Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

So, best decision boundary is at point $x$ where

$$P(\text{class is } 9 \mid x) L(9 \rightarrow 4) = P(\text{class is } 4 \mid x) L(4 \rightarrow 9)$$

To classify a new point, choose class with lowest expected loss; i.e., choose “four” if

$$P(4 \mid x) L(4 \rightarrow 9) > P(9 \mid x) L(9 \rightarrow 4)$$

How to evaluate these probabilities?
Probability

- Basic probability
  - X is a random variable
  - \( P(X) \) is the probability that X achieves a certain value

- Conditional probability: \( P(X \mid Y) \)
  - probability of X given that we already know Y

\[
P(X) \geq 0 \quad \text{and} \quad \sum P(X) = 1
\]

- Continuous X: \( \int_{-\infty}^{\infty} P(X) \, dX = 1 \)

- Discrete X: \( \sum P(X) = 1 \)

Source: Steve Seitz
Example: learning skin colors

• We can represent a class-conditional density using a histogram (a “non-parametric” distribution)

\[ P(x|\text{skin}) \]

\[ P(x|\text{not skin}) \]

Percentage of skin pixels in each bin

Feature \( x = \text{Hue} \)

Kristen Grauman
Example: learning skin colors

- We can represent a class-conditional density using a histogram (a “non-parametric” distribution)

Now we get a new image, and want to label each pixel as skin or non-skin.

What’s the probability we care about to do skin detection?

Kristen Grauman
Bayes rule

\[ P(\text{skin} \mid x) = \frac{P(x \mid \text{skin})P(\text{skin})}{P(x)} \]

\[ P(\text{skin} \mid x) \propto P(x \mid \text{skin})P(\text{skin}) \]
Example: classifying skin pixels

Now for every pixel in a new image, we can estimate probability that it is generated by skin.

Classify pixels based on these probabilities

- if $p(\text{skin}|\mathbf{x}) > \theta$, classify as skin
- if $p(\text{skin}|\mathbf{x}) < \theta$, classify as not skin
Example: classifying skin pixels

Figure 6: A video image and its flesh probability image

Figure 7: Orientation of the flesh probability distribution marked on the source video image

Gary Bradski, 1998
Example: classifying skin pixels

Using skin color-based face detection and pose estimation as a video-based interface

Gary Bradski, 1998
Supervised classification

• Want to minimize the expected misclassification

• Two general strategies
  – Use the training data to build representative probability model; separately model class-conditional densities and priors (*generative*)
  – Directly construct a good decision boundary, model the posterior (*discriminative*)
Discriminative classifiers for image recognition
– nearest neighbors (+ scene match app)
– support vector machines (+ gender, person app)
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 from Duda et al.

Novel test example

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

Black = negative
Red = 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.

Source: D. Lowe
A nearest neighbor recognition example
Where in the World?

Where in the World?
Where in the World?

Slides: James Hays
6+ million geotagged photos by 109,788 photographers

Annotated by Flickr users

Slides: James Hays
Which scene properties are relevant?
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.

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

\[ \rightarrow \left| V_t \right| \rightarrow \text{PCA} \rightarrow G \]

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
Scene Matches


Slides: James Hays
Scene Matches

Scene Matches


Slides: James Hays
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
Linear classifiers
Lines in $\mathbb{R}^2$

Let

\[ w = \begin{bmatrix} a \\ c \end{bmatrix}, \quad x = \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ ax + cy + b = 0 \]
Lines in $\mathbb{R}^2$

Let

$$w = \begin{bmatrix} a \\ c \end{bmatrix} \quad x = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$ax + cy + b = 0$$

$$w \cdot x + b = 0$$
Lines in $\mathbb{R}^2$

Let $w = \begin{bmatrix} a \\ c \end{bmatrix}$, $x = \begin{bmatrix} x \\ y \end{bmatrix}$

\[ ax + cy + b = 0 \]

\[ w \cdot x + b = 0 \]
Lines in $\mathbb{R}^2$

Let $w = \begin{bmatrix} a \\ c \end{bmatrix}$ and $x = \begin{bmatrix} x \\ y \end{bmatrix}$

$$aw + cy + b = 0$$

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

distance from point to line

$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}}$$
Let \( w = \begin{bmatrix} a \\ c \end{bmatrix} \) and \( x = \begin{bmatrix} x \\ y \end{bmatrix} \).

Equation of a line in \( \mathbb{R}^2 \):

\[
ax + cy + b = 0
\]

Distance from point to line:

\[
D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}} = \frac{w^T x + b}{\|w\|}
\]

\( (x_0, y_0) \) is the point, \( w \) is the normal vector of the line, and \( b \) is the constant term.
Linear classifiers

• Find linear function to separate positive and negative examples

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

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.

\[ \begin{align*}
\text{x}_i \text{ positive (} y_i = 1 \text{)}: & \quad \text{x}_i \cdot \mathbf{w} + b \geq 1 \\
\text{x}_i \text{ negative (} y_i = -1 \text{)}: & \quad \text{x}_i \cdot \mathbf{w} + b \leq -1 \\
\text{For support vectors,} & \quad \text{x}_i \cdot \mathbf{w} + b = \pm 1
\end{align*} \]

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:

\[
\frac{w^T x + b}{\|w\|} = \pm 1 \\
M = \frac{1}{\|w\|} - \frac{-1}{\|w\|} = \frac{2}{\|w\|}
\]
Support vector machines

• Want line that maximizes the margin.

\[ \mathbf{x}_i \text{ positive (} y_i = 1) : \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1 \]
\[ \mathbf{x}_i \text{ negative (} y_i = -1) : \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \]

For support, vectors, \[ \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1 \]

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

Therefore, the margin is \( \frac{2}{\|\mathbf{w}\|} \)
Finding the maximum margin line

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

• **Quadratic optimization problem:**

\[
\text{Minimize } \frac{1}{2} \mathbf{w}^T \mathbf{w}
\]

Subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$
Finding the maximum margin line

- Solution: \( \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \)
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:

\[
f(x) = \text{sign} \left( w \cdot x + b \right)
\]

\[
= \text{sign} \left( \sum_i \alpha_i x_i \cdot x + b \right)
\]

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

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?
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.

Code available: http://pascal.inrialpes.fr/soft/olt/
Person detection with HoG’s & linear SVM’s

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

SIFT, D. Lowe, ICCV 1999

Shape context
Belongie, Malik, Puzicha, NIPS 2000

Count the number of points inside each bin, e.g.:

- Count = 4
- Count = 10

Compact representation of distribution of points relative to each point
Bin gradients from 8x8 pixel neighborhoods into 9 orientations (Dalal & Triggs CVPR 05)

Histograms of oriented gradients (HOG)

Source: Deva Ramanan
Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.
Histograms of Oriented Gradients for Human Detection

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

Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.
A Support Vector Machine (SVM) learns a classifier with the form:

$$H(x) = \sum_{m=1}^{M} a_m y_m k(x, x_m)$$

Where \( \{x_m, y_m\} \), for \( m = 1 \ldots M \), are the training data with \( x_m \) being the input feature vector and \( y_m = +1,-1 \) the class label. \( k(x, x_m) \) is the kernel and it can be any symmetric function satisfying the Mercer Theorem.

The classification is obtained by thresholding the value of \( H(x) \).

There is a large number of possible kernels, each yielding a different family of decision boundaries:

- Linear kernel: \( k(x, x_m) = x^T x_m \)
- Radial basis function: \( k(x, x_m) = \exp(-|x - x_m|^2/\sigma^2) \).
- Histogram intersection: \( k(x,x_m) = \sum_i (\min(x(i), x_m(i))) \)
Linear SVM

\[ f(x) = (w \cdot x + b) \]
Scanning-window templates

Dalal and Triggs CVPR05 (HOG)

Papageorgiou and Poggio ICIP99 (wavelets)

\(w_{x} > 0\)

\(w = \text{weights for orientation and spatial bins}\)

\(w \cdot x > 0\)

Train with a linear classifier (perceptron, logistic regression, SVMs...)

Source: Deva Ramanan
How to interpret positive and **negative** weights?

\[ w \cdot x > 0 \]

\[ (w_{pos} - w_{neg}) \cdot x > 0 \]

\[ w_{pos} \cdot x > w_{neg} \cdot x \]

Right approach is to **compete** pedestrian, pillar, doorway... models.

Background class is hard to model - easier to penalize particular vertical edges.

\[ w_{pos}, w_{neg} = \text{weighted average of positive, negative support vectors} \]

Source: Deva Ramanan
Histograms of oriented gradients

Dalal & Trigs, 2006

Not a person

person
Figure 3. The performance of selected detectors on (left) MIT and (right) INRIA data sets. See the text for details.
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-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:

\[ \Phi: x \rightarrow \varphi(x) \]

Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html
The “Kernel Trick”

- The linear classifier relies on dot product between vectors $K(x_i, x_j) = x_i^T x_j$

- If every data point is mapped into high-dimensional space via some transformation $\Phi: x \rightarrow \phi(x)$, the dot product becomes:

  $$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- A kernel function is similarity function that corresponds to an inner product in some expanded feature space.

Slide from Andrew Moore’s tutorial: http://www.autonlab.org/tutorials/svm.html
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 $K(x_i,x_j)= \phi(x_i)^T \phi(x_j)$:

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

where $\phi(x) = [1 \ x_1^2 \ \sqrt{2} x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2]$
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:

$$\sum_i \alpha_i y_i K(x_i, x) + b$$
Examples of kernel functions

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

- Linear:

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

- Gaussian RBF:

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

- Histogram intersection:
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.
Example: learning gender with SVMs

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

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

• Training examples:
  – 1044 males
  – 713 females

• Experiment with various kernels, select Gaussian RBF

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

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

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall</td>
</tr>
<tr>
<td>SVM with RBF kernel</td>
<td>3.38%</td>
</tr>
<tr>
<td>SVM with cubic polynomial kernel</td>
<td>4.88%</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>10.63%</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.
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.
Gender perception experiment: How well can humans do?

Stimuli →

Results →

<table>
<thead>
<tr>
<th></th>
<th>High-Res</th>
<th>Low-Res</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>6.54%</td>
<td>30.7%</td>
</tr>
</tbody>
</table>

\( \sigma = 3.7\% \)

Moghaddam and Yang, Face & Gesture 2000.
Human vs. Machine

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

Figure 6. SVM vs. Human performance
Hardest examples for humans

Top five human misclassifications

Moghaddam and Yang, Face & Gesture 2000.
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: 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

• Discriminative classifiers
  – Boosting
  – Nearest neighbors
  – Support vector machines

• Useful for object recognition when combined with “window-based” or holistic appearance descriptors
Global window-based appearance representations

- These examples are truly global; each pixel in the window contributes to the representation.
- Classifier can account for relative relevance...
- *When might this not be ideal?*
Generic category recognition: representation choice

Window-based

Part-based