CS 3710: Visual Recognition
Classification and Detection

Adriana Kovashka
Department of Computer Science
January 13, 2015
Plan for Today

• Visual recognition basics part 2: Classification and detection
• Adriana’s research
• Next time: First student presentation
Classification vs Detection

• Classification
  – Given an image or an image region, determine which of $N$ categories it represents

• Detection
  – Determine where in the image a category is to be found
Classification
Machine Learning Problems

- **Supervised Learning**
  - Discrete: classification or categorization
  - Continuous: regression

- **Unsupervised Learning**
  - Discrete: clustering
  - Continuous: dimensionality reduction
The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

\[
\begin{align*}
f(\text{apple}) &= \text{“apple”} \\ f(\text{tomato}) &= \text{“tomato”} \\ f(\text{cow}) &= \text{“cow”}
\end{align*}
\]
The machine learning framework

\[ y = f(x) \]

- **Training:** given a training set of labeled examples \( \{(x_1, y_1), \ldots, (x_N, y_N)\} \), estimate the prediction function \( f \) by minimizing the prediction error on the training set

- **Testing:** apply \( f \) to a never before seen test example \( x \) and output the predicted value \( y = f(x) \)
**Training**

- Training Images

**Steps**

- Training Labels
- Image Features
- Training
- Learned model

**Testing**

- Test Image
- Image Features
- Learned model
- Prediction

Derek Hoiem, Svetlana Lazebnik
Recognition task and supervision

• Images in the training set must be annotated with the “correct answer” that the model is expected to produce

“Contains a motorbike”
Generalization

• How well does a learned model generalize from the data it was trained on to a new test set?
Classification

- Assign input vector to one of two or more classes
- Any decision rule divides the input space into decision regions separated by decision boundaries
Supervised classification

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

  “four”
  “nine”

Training examples          Novel input

• How good is some function that 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

Feature value \( x \)

Optimal classifier will minimize total risk.

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

If we choose class “four” at boundary, expected loss is:

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

If we choose class “nine” at boundary, expected loss is:

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

So, best decision boundary is at point \( x \) where

\[
P(\text{class is 9} \mid x) \ L(9 \rightarrow 4) = P(\text{class is 4} \mid x) L(4 \rightarrow 9)
\]
Supervised classification

Optimal classifier will minimize total risk.

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

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

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

Loss for choosing “four”  Loss for choosing “nine”
Supervised classification

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

\[
P(9 | x)L(9 \rightarrow 4) < P(4 | x)L(4 \rightarrow 9)
\]

Loss for choosing “four”     Loss for choosing “nine”

Optimal classifier will minimize total risk.

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

How to evaluate these probabilities?
Classifiers: Nearest neighbor

\[ f(x) = \text{label of the training example nearest to } x \]

- All we need is a distance function for our inputs
- No training required!
1-nearest neighbor
5-nearest neighbor
Linear classifiers

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998
Lines in $\mathbb{R}^2$

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

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

Let

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

$$ax + cy + b = 0$$

$$w \cdot x + b = 0$$
Let \( w = \begin{bmatrix} a \\ c \end{bmatrix} \) and \( x = \begin{bmatrix} x \\ y \end{bmatrix} \)

\[
ax + cy + b = 0
\]

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

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

The equation of the line is $ax + cy + b = 0$.

The distance from point $(x_0, y_0)$ to the line is given by:

$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}}$$

Kristen Grauman
Lines in $\mathbb{R}^2$

Let

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

$$ax + cy + b = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

Distance from point to line

$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}} = \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|}$$
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?

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

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

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998
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, \[ \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1 \]

Support vector machines

- Want line that maximizes the margin.

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

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

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

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

Support vector machines

- Want line that maximizes the margin.

\[ w^T x + b = \pm 1 \]

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

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

For support vectors, \( \text{x}_i \cdot w + b = \pm 1 \)

Distance between point and line:

\[ \frac{|\text{x}_i \cdot w + b|}{||w||} \]

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

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

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

Subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$

One constraint for each training point.

Note sign trick.

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998
Finding the maximum margin line

• Solution: \( w = \sum_i \alpha_i y_i x_i \)
Finding the maximum margin line

• Solution: \( w = \sum_i \alpha_i y_i x_i \)
  \[ b = y_i - w \cdot x_i \] (for any support vector)

• Classification function:
  \[ f(x) = \text{sign} \left( w \cdot x + b \right) \]
  \[ = \text{sign} \left( \sum_i \alpha_i y_i x_i \cdot x + b \right) \]

  If \( f(x) < 0 \), classify as negative, otherwise classify as positive.

• 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)

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

• Datasets that are linearly separable work out great:

• But what if the dataset is just too hard?

• We can map it to a higher-dimensional space:
Nonlinear SVMs

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

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

• *The kernel trick*: instead of explicitly computing the lifting transformation $\phi(x)$, define a kernel function $K$ such that

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$
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:**
  \[ K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k)) \]
Summary:
SVMs for image classification

1. Pick an image representation
2. Pick a kernel function for that representation
3. Compute the matrix of kernel values between every pair of training examples
4. Feed the kernel matrix into your favorite SVM solver to obtain support vectors and weights
5. At test time: compute kernel values for your test example and each support vector, and combine them with the learned weights to get the value of the decision function
What about multi-class SVMs?

- Unfortunately, there is no “definitive” multi-class SVM formulation
- In practice, we have to obtain a multi-class SVM by combining multiple two-class SVMs
- One vs. others
  - Training: learn an SVM for each class vs. the others
  - Testing: apply each SVM to the test example, and assign it to 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
Detection
PASCAL Visual Object Challenge

aeroplane bike bird boat bottle bus car cat chair cow table dog horse motorbike person plant sheep sofa train tv

Deva Ramanan
Detection: Scanning-window templates

Dalal and Triggs CVPR05 (HOG)
Papageorgiou and Poggio ICIP99 (wavelets)

\[ w \cdot x > 0 \]

\[ w = \text{weights for orientation and spatial bins} \]
Deformable part models

Model encodes local appearance + spring deformation
Homework for Next Time

• Paper review for Features due at 10pm on 1/14, send to kovashka@cs.pitt.edu