6.819 / 6.869: Advances in Computer Vision

Learning:
Introduction to Machine Learning for Vision

Website:
http://6.869.csail.mit.edu/fa15/

Instructor: Yusuf Aytar

Lecture TR 9:30AM – 11:00AM
(Room 34-101)
Key Concepts

**Concepts**
Pattern, Category/Class, Instance, Generalization, Classification

**Classification**
Feature space, Objective function, Regularization, Loss Function, Optimization

**Support Vector Machines**
Where is the bottle?
Where is Waldo?
Find the same patch

Task: find the most similar patch in a second image
人生易老天难老
岁岁重阳，今又重阳
战地黄花分外香
一抹夕阳一抹霞
春光寥廓
万里江山
Pattern vs Category

Computers are good with patterns and We are good with categories

... but computers are also getting better with categories
Patterns have discriminative representations with less variation.

Categories also have discriminative representations, but with great variations.

Oliva & Torralba (2007) TICS
Instance vs Category

**Instances** Find these two objects

**Categories** Find a bottle:
Generalization: Extracting the essence of a concept based on its analysis of similarities from many discrete objects.

http://en.wikipedia.org/wiki/Generalization
Challenges of Generalization

A successful object category detector should be invariant to changes in illumination, occlusion, background clutter, scale, viewpoint, deformation and intra-class variance.
Object Instance Detection

Find the Object

Which of the invariances below apply for the given object instance detection problem?

illumination, occlusion, background clutter, scale, viewpoint, deformation and intra-class variance.
Classification vs. Detection

Is this a … image?

Where is the …?
Localize the object.

Detection can be performed through a classifier, i.e. sliding window search

Kitchen  Table  Horse

Waldo  Car
Every training sample is represented as a point in the feature space.
Example Feature Spaces

**SIFT**: Scale-Invariant Feature Transform
(Lowe, 1999)

**HOG**: Histograms of oriented gradients
(Dalal & Triggs CVPR 05)

**Gist**: Grid of gabor filters
(Oliva & Torralba, 2001)
Machine Learning Methods

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

- Unsupervised Learning:
  - Discrete: clustering
  - Continuous: dimensionality reduction
Generative vs Discriminative

- Represent both the data and the labels
- Often, makes use of conditional independence and priors
- Examples
  - Naïve Bayes classifier
  - Bayesian network
- Models of data may apply to future prediction problems

- Learn to directly predict the labels from the data
- Often, assume a simple boundary (e.g., linear)
- Examples
  - Logistic regression
  - SVM
  - Boosted decision trees
- Often easier to predict a label from the data than to model the data
Discriminative Models

Nearest neighbor

Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

10^6 examples

Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
...

Support Vector Machines

Guyon, Vapnik, Heisele, Serre, Poggio...

Latent SVM

Structural SVM

Felzenszwalb 00
Ramanan 03...

Boosting

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

Source: Vittorio Ferrari, Kristen Grauman, Antonio Torralba, Fei-Fei Li
Classification

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

\[
\begin{align*}
\text{f(🍎)} &= \text{“apple”} \\
\text{f(🍅)} &= \text{“tomato”} \\
\text{f(🐮)} &= \text{“cow”}
\end{align*}
\]
Classification Formulation

\[ y = f(x) \]

- **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 \( y = f(x) \)

Slide credit: L. Lazebnik
Learning Framework

Training

Training Images

Image Features

Training Labels

Training

Learned model

Testing

Test Image

Image Features

Learned model

Prediction

Slide credit: D. Hoiem and L. Lazebnik
f(x) = label of the training example nearest to x in the feature space

- All we need is a distance function for our inputs
- No training required!
K-Nearest neighbor classification

Test example
3-Nearest neighbor classification
5-Near neighbor classification

Simple, a good one to try first
Binary classification can be viewed as the task of separating classes in feature space:

\[ w^T x + b = 0 \]

\[ w^T x + b > 0 \]

\[ w^T x + b < 0 \]

\[ f(x) = \text{sign}(w^T x + b) \]
Linear Classifiers

\[ f(x_i) = \text{sign}(w^T x_i + b) \]

Which one is a better classifier?

A

B

C
Linear Classifiers

\[ f(x_i) = \text{sign}(w^T x_i + b) \]

Which one is a better classifier?
Support Vector Machines (Intuition)
The linear discriminant function (classifier) with the maximum margin is the best.

Margin is defined as the width that the boundary could be increased by before hitting a data point.

Why it is the best?

- Robust to outliers and thus strong generalization ability.
Support Vector Machines

- Given a set of data points:
  \[ \{(x_i, y_i)\}, \ i = 1, 2, \ldots, n, \text{ where} \]
  
  For \( y_i = +1 \), \( w^T x_i + b > 0 \)
  
  For \( y_i = -1 \), \( w^T x_i + b < 0 \)

- With a scale transformation on both \( w \) and \( b \), the above is equivalent to
  
  For \( y_i = +1 \), \( w^T x_i + b \geq 1 \)
  
  For \( y_i = -1 \), \( w^T x_i + b \leq -1 \)
Support Vector Machines

\[ \min_{w,b} \quad \|w\|^2 + C \sum_i \max(0, 1 - y_i (w^T x_i + b)) \]

- **Regularizer**
- **Loss Function (Hinge Loss)**
Support Vector Machines

Objective Function

\[
\min_{w,b} \quad \|w\|^2 + C \sum_i \max(0,1 - y_i (w^T x_i + b))
\]

Prediction Function

\[
y = \text{sign}(w^T x_i + b)
\]

Learning: Convex Optimization
Non-Linear 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:
Non-Linear SVMs

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

\[ \Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x}) \]
Non-Linear Kernel: Example

- Consider the mapping \( \varphi(x) = (x, x^2) \)

\[
\varphi(x) \cdot \varphi(y) = (x, x^2) \cdot (y, y^2) = xy + x^2 y^2
\]

\[
K(x, y) = xy + x^2 y^2
\]
Non-Linear 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:

\[
\sum_i \alpha_i y_i \phi(x_i) \cdot \phi(x) + b = \sum_i \alpha_i y_i K(x_i, x) + b
\]
Common Kernel Functions

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

- Polynomial kernel: \[ K(x_i, x_j) = (1 + x_i^T x_j)^p \]

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

- Sigmoid:
  \[ K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1) \]

- In general, functions that satisfy *Mercer’s condition* can be kernel functions.
Summary: SVMs for image classification

1. Pick an image representation (HoG, SIFT+BOW, etc.)
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
MATLAB SVM Example

```matlab
load fisheriris
xdata = meas(51:end,3:4);
group = species(51:end);
svmStruct = svmtrain(xdata,group,'ShowPlot',true);
```
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 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 (LibSVM, Liblinear, etc): [http://www.kernel-machines.org/software](http://www.kernel-machines.org/software)
  • Kernel-based framework is very powerful, flexible
  • SVMs work very well in practice, even with very small training sample sizes

• Cons
  • No “direct” multi-class SVM, must combine two-class SVMs
  • 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
How well does a learned model generalize from the data it was trained on to a new test set?
Overfitting vs Underfitting

“Everything should be made as simple as possible, but not simpler.”

Albert Einstein

**Underfitting**: model is too “simple” to represent all the relevant class characteristics

**Overfitting**: model is too “complex” and fits irrelevant characteristics (noise) in the data
Use Case: Linear SVMs over HoG

Traditional Detector Training (Motorbike)

Training Samples
- Positive Samples
- Negative Samples

Feature Extraction
(Histogram of Oriented Gradients)

HOG Features
- Positive HOGs
- Negative HOGs

Linear SVM

Motorbike Detector

[Dalal et al. CVPR’05]
[Felzenszwalb et al. CVPR’08]
Use Case: Exemplar SVMs

Training an SVM with a **single positive** and **many negative** samples

Linear SVMs over HoG features

Exemplar SVM

[Dalal & Triggs'05],
[Felzenszwalb'08],
[Malisiewicz'11]
Another Classifier: Randomized Decision Forests

Shotton et al., Real-Time Human Pose Recognition in Parts from Single Depth Images, CVPR, 2011
Body Part Classification with Randomized Decision Forests

Shotton et.al., Real-Time Human Pose Recognition in Parts from Single Depth Images, CVPR, 2011
Another Classifier: Deep Learning