Lecture 22: Implementing Recognition
Last Time

• We talked about the bag-of-words representation for images
feature detection & representation

image representation

codewords dictionary

category models (and/or) classifiers

category decision
Today

- We will talk about how to implement the classification step

- The key problem is, how do we use histograms to match categories?

Figure from Li’s tutorial
Basic Classifier

• We’ll start with a basic nearest neighbor classifier
• To classify an image, look at its neighbors
• Let them vote
Key problem for Recognition

- How do we compute distance between histograms?
Best Survey of Methods

• Every vision student should read this paper

The Earth Mover’s Distance as a Metric for Image Retrieval

YOSSI RUBNER, CARLO TOMASI AND LEONIDAS J. GUIBAS
Computer Science Department, Stanford University, Stanford, CA 94305, USA
rubner@cs.stanford.edu
tomasi@cs.stanford.edu
guibas@cs.stanford.edu

Abstract. We investigate the properties of a metric between two distributions, the Earth Mover’s Distance (EMD), for content-based image retrieval. The EMD is based on the minimal cost that must be paid to transform one distribution into the other, in a precise sense, and was first proposed for certain vision problems by Peleg, Werman,
For the next few slides

- We’ll assume we have histogram \( H \) and \( K \). Specific bins will be \( h_i \) and \( k_i \)
- Most basic comparison - \textit{Minkowski Form Distance}

\[
d_{L_r}(H, K) = \left( \sum_i |h_i - k_i|^r \right)^{1/r}
\]
Histogram Intersection

• The histogram intersection can match histograms when the number of pixels is different
• We’ll come back to this one

\[ d \cap (H, K) = 1 - \frac{\sum_i \min(h_i, k_i)}{\sum_i k_i} \]
Chi-Squared

\( \chi^2 \) Statistics:

\[
d_{\chi^2}(H, K) = \sum_i \frac{(h_i - m_i)^2}{m_i}
\]
Jeffrey Divergence

\[ d_J(H, K) = \sum_i \left( h_i \log \frac{h_i}{m_i} + k_i \log \frac{k_i}{m_i} \right) \]
How do we evaluate a system?

- Several popular databases are available
- For BoW-style recognition, most popular are CalTech-101, CalTech-256
Caltech 101 & 256

Fei-Fei, Fergus, Perona 2004

Griffin, Holub, Perona 2007

(Slide from Torralba Tutorial)
Visual Object Classes Challenge 2009 (VOC2009)

3rd October 2009, ICCV 2009, Kyoto, Japan

(Slide from Torralba Tutorial)
Lotus Hill Dataset

(Slide from Torralba Tutorial)
Yao, Liang, Zhu, EMMCVPR, 2007
14,847 categories, 9,349,136 images

- Animals
  - Fish
  - Bird
  - Mammal
  - Invertebrate

- Sport Activities
- Fabric Materials
- Instrumentation
  - Tool
  - Appliances
  - ...

- Scenes
  - Indoors
  - Geological formations

- Plants
  - ...

(Slide from Torralba Tutorial)

Deng, Wei, Socher, Li, Li, Fei-Fei, CVPR 2009
“Cycling”

The sport of traveling on a bicycle or motorcycle

(Slide from Torralba Tutorial)

Deng, Wei, Socher, Li, Li, Fei-Fei, CVPR 2009
How Many Papers are Compared

![Graph showing the comparison of accuracy with the number of training examples. Each line represents a different paper or method, with colors indicating different sources such as CVPR06, ICCV05, and TR06. The legend lists the following papers:

- Zhang, Berg, Maire, and Malik (CVPR06)
- Lazebnik, Schmid, and Ponce (CVPR06)
- Wang, Zhang, and Fei-Fei (CVPR06)
- Grauman and Darrell (ICCV05)
- Mutch and Lowe (CVPR06)
- Pinto, Cox, and DiCarlo (PLOS08)
- Griffin, Holub, and Perona (TR06)
- LP-β (this paper)
Most Popular Method: SVMs

• Support Vector Machines are probably the most popular method for implementing recognition systems
The Support Vector Machine
The Support Vector Machine

• Last time, we talked about different criteria for a good classifier

• Now, we will consider a criterion called the margin
The Support Vector Machine

- Margin – minimum distance from a data point to the decision boundary
The Support Vector Machine

- The SVM finds the boundary that maximizes the margin
The Support Vector Machine

- Data points that are along the margin are called support vectors.
Non-Linear Classification in SVMs

- We will do the same trick as before

This is the decision boundary from
\[ x^2 + 8xy + y^2 > 0 \]

This is the same as making a new set of features, then doing linear classification.
Non-Linear Classification in SVMs

• The decision function can be expressed in terms of dot-products

\[ f(x) = b + \sum_{i=1}^{N} \alpha_i y_i < x, x_i > \]

• Each \( \alpha \) will be zero unless the vector is a support vector
Non-Linear Classification in SVMs

• What if we wanted to do non-linear classification?
• We could transform the features and compute the dot product of the transformed features.
• But there may be an easier way!
The Kernel Trick

• Let \( \Phi(x) \) be a function that transforms \( x \) into a different space

\[
\Phi : \mathbb{R}^d \rightarrow \mathcal{H}.
\]

• A kernel \( K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \) such that
Kernels for BoW Recognition

• It turns out that many of the histogram measures that we talked about previously are kernels
Histogram Intersection Kernel

\[ K(x, y) = \sum_k \min(x_k, y_k) \]
Chi-Squared Kernel used in [Gehler 2009]

\[ K(x, y) = \exp \left( \frac{d(x, y)}{\gamma} \right) \]
How to build your own object recognitions system

1. Download a lot of images

Fei-Fei, Fergus, Perona
2004

Griffin, Holub, Perona
2007
How to build your own object recognitions system

- 2. Extract a lot of image patches
- 3. Represent with some feature representation
Vector quantization

4. Make Dictionary

Slide credit: Josef Sivic
5. Build Histograms for all images

Figure from Li’s tutorial
6. Compute Kernel Matrix

\[ K(x, y) = \sum_{k} \min(x_k, y_k) \]

\[
\begin{bmatrix}
K(x_1, x_1) & K(x_1, x_2) & K(x_1, x_3) & \ldots \\
K(x_2, x_1) & K(x_2, x_2) & K(x_2, x_3) & \ldots \\
\vdots & \vdots & \vdots & \ddots \\
\end{bmatrix}
\]
7. Call libSVM

LIBSVM -- A Library for Support Vector Machines

Chih-Chung Chang and Chih-Jen Lin

*NEW* Version 2.9 released on November 1, 2009. Scripts for parameter selection now works under python 3.x as well.
*NEW* We now have a nice page LIBSVM data sets providing problems in LIBSVM format.
*NEW* A practical guide to SVM classification is available now! (mainly written for beginners)
*NEW* libsvm tools available now!

We now have an easy script (easy.py) for users who know NOTHING about svm. It makes everything automatic--from data scaling to parameter selection.

The parameter selection tool grid.py generates the following contour of cross-validation accuracy. To use this tool, you also need to install python and gnuplot.