Mining, and Intro to Categorization

Tues April 11
Kristen Grauman
UT Austin

Recognition and learning

Recognizing categories (objects, scenes, activities, attributes...), learning techniques

Last time

- Instance recognition wrap up:
  - Spatial verification
  - Sky mapping example
  - Query expansion
Review questions

• Does an inverted file index sacrifice accuracy in bag-of-words image retrieval? Why or why not?
• Name a pro and con of query expansion.
• Why does a single SIFT match cast a 4D vote for the Generalized Hough spatial verification model?
• What does a perfect precision recall curve look like?

Today

• Discovering visual patterns
  • Randomized hashing algorithms
  • Mining large-scale image collections
• Introduction to visual categorization

Locality Sensitive Hashing (LSH)

[indyk and motwani '98, gionis et al. '99, charikar '02, andoni et al. '04]

 Guarantees approximate near neighbors in sub-linear time, given appropriate hash functions.
The probability that a random hyperplane separates two unit vectors depends on the angle between them:

\[
\Pr[\text{sign}(x_i^T r) = \text{sign}(x_j^T r)] = 1 - \frac{1}{\pi} \cos^{-1}(x_i^T x_j)
\]

Corresponding hash function:

\[
h_r(x) = \begin{cases} 
1, & \text{if } r^T x \geq 0 \\
0, & \text{otherwise}
\end{cases}
\]

for \( r_i \sim N(\mu = 0, \sigma^2 = 1) \)


---

**LSH function example:**

**inner product similarity**

The probability that a random hyperplane separates two unit vectors depends on the angle between them:

\[
\Pr[\text{sign}(x_i^T r) = \text{sign}(x_j^T r)] = 1 - \frac{1}{\pi} \cos^{-1}(x_i^T x_j)
\]

Corresponding hash function:

\[
h_r(x) = \begin{cases} 
1, & \text{if } r^T x \geq 0 \\
0, & \text{otherwise}
\end{cases}
\]

for \( r_i \sim N(\mu = 0, \sigma^2 = 1) \)


---

**LSH function example:**

**Min-hash for set overlap similarity**

[Broder, 1999]

\[
\Pr_{h \in \mathcal{H}}[h(x) = h(y)] = \text{sim}(x, y)
\]

\[
\text{sim}(A_1, A_2) = \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|} \in (0, 1)
\]

[Broder, 1999]
LSH function example:
Min-hash for set overlap similarity

\[ A : \{A, E, Q, R, V\} \quad B : \{A, C, Q, V, Z\} \]

Ordering by \( f_1 \)
\[ A \cup B : \{A, C, E, J, Q, R, V, Z\} \]

\[ h_1(A) \quad h_1(B) \]
\[ h_2(A) \quad h_2(B) \]

\[ \Pr(h(A) = h(B)) = \frac{|A \cap B|}{|A \cup B|} \]

Multiple hash functions and tables

- Generate \( k \) such hash functions, concatenate outputs into hash key:
  \[ p(h_{\ell_{-1}}(x) = h_{\ell_{-1}}(y)) = \]

- To increase recall, search multiple independently generated hash tables
  - Search/rank the union of collisions in each table, or
  - Require that two examples in at least \( T \) of the tables to consider them similar.

Mining for common visual patterns

In addition to visual search, want to be able to summarize, mine, and rank the large collection as a whole.

- What is common?
- What is unusual?
- What co-occurs?
- Which exemplars are most representative?
Mining for common visual patterns

In addition to visual search, want to be able to **summarize, mine, and rank** the large collection as a whole.

We’ll look at a few examples:
- **Connected component clustering** via hashing
  - [Geometric Min-hash, Chum et al., 2009]
- **Visual Rank** to choose “image authorities”
  - [Jing and Baluja, 2008]
- **Frequent item-set mining** with spatial patterns
  - [Quack et al., 2007]

### Connected component clustering with hashing

1. Detect seed pairs via hash collisions
2. Hash to related images
3. Compute connected components of the graph

---

### Geometric Min-hash

[Chum, Perdoch, Matas, CVPR 2009]

- Main idea: build spatial relationships into the hash key construction:
  - Select first hash output according to min hash (“central word”)
  - Then append subsequent hash outputs from within its neighborhood
Results:
Geometric Min-hash clustering
[Chum, Perdoch, Matas, CVPR 2009]

- 100,000 Images downloaded from FLICKR
- Includes 11 Oxford Landmarks with manually labeled ground truth

Slide credit: Ondrej Chum

Results:
Geometric Min-hash clustering
[Chum, Perdoch, Matas, CVPR 2009]

Discovering small objects

Slide credit: Ondrej Chum
Mining for common visual patterns

In addition to visual search, want to be able to **summarize, mine, and rank** the large collection as a whole.

We'll look briefly at a few recent examples:
- **Connected component clustering** via hashing
  [Geometric Min-hash, Chum et al. 2009]
- **Visual Rank** to choose “image authorities” [Jing and Baluja, 2008]
- **Frequent item-set mining** with spatial patterns
  [Quack et al., 2007]

Visual Rank: motivation

- **Goal**: select small set of “best” images to display among millions of candidates

Visual Rank

[Geometric Min-hash, Chum et al. 2009]

- **Goal**: select small set of “best” images to display among millions of candidates

- **Main ideas**:
  - Graph weights = number of matched local features between two images
  - Exploit text search to narrow scope of each graph
  - Use LSH to make similarity computations efficient
Results: Visual Rank

[Jing and Baluja, PAMI 2008]

Kristen Grauman

Results: Visual Rank

Similarity graph generated from top 1,000 text search results of "Lincoln Memorial".

Note the diversity of the high-ranked images.

Mining for common visual patterns

In addition to visual search, want to be able to summarize, mine, and rank the large collection as a whole.

We’ll look briefly at a few recent examples:

- **Connected component clustering** via hashing
  [Geometric Min-hash, Chum et al. 2009]
- **Visual Rank** to choose "image authorities" [Jing and Baluja, 2008]
- **Frequent item-set mining** with spatial patterns
  [Quack et al., 2007]
Frequent item-sets

Frequent item-set mining for spatial visual patterns
[Quack, Ferrari, Leibe, Van Gool, CIVR 2006, ICCV 2007]

- What configurations of local features frequently occur in large collection?
- **Main idea:** Identify item-sets (visual word layouts) that often occur in transactions (images)
- Efficient algorithms from data mining (e.g., Apriori algorithm, Agrawal 1993)
Frequent item-set mining for spatial visual patterns
[Quack, Ferrari, Leibe, Van Gool, CIVR 2006, ICCV 2007]

Two example itemset clusters

Discovering favorite views
Discovering Favorite Views of Popular Places with Iconoid

Today

• Discovering visual patterns
  • Randomized hashing algorithms
  • Mining large-scale image collections

• Introduction to visual categorization
What does recognition involve?

Detection: are there people?

Activity: What are they doing?
Object categorization

- mountain
- tree
- building
- banner
- street lamp
- people
- vendor

Instance recognition

- Potala Palace
- A particular sign

Scene and context categorization

- outdoor
- city
- ...
Object Categorization

- Task Description
  “Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label.”

- Which categories are feasible visually?

Visual Object Categories

- Basic Level Categories in human categorization [Rosch 76, Lakoff 87]
  - The highest level at which category members have similar perceived shape
  - The highest level at which a single mental image reflects the entire category
  - The level at which human subjects are usually fastest at identifying category members
  - The first level named and understood by children
  - The highest level at which a person uses similar motor actions for interaction with category members
Visual Object Categories

- Basic-level categories in humans seem to be defined predominantly visually.
- There is evidence that humans (usually) start with basic-level categorization before doing identification.

⇒ Basic-level categorization is easier and faster for humans than object identification!

- How does this transfer to automatic classification algorithms?

Basic level

Abstract levels

Individual level

How many object categories are there?

~10,000 to 30,000

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.
Other Types of Categories

- Functional Categories
  - e.g. chairs = “something you can sit on”

Why recognition?

- Recognition a fundamental part of perception
  - e.g., robots, autonomous agents

- Organize and give access to visual content
  - Connect to information
  - Detect trends and themes

Autonomous agents able to detect objects
Discovering visual patterns

Auto-annotation

Challenges: robustness
Challenges: context and human experience

Context cues

Function

Dynamics

Challenges: complexity

- Millions of pixels in an image
- 30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Billions of images online
- 82 years to watch all videos uploaded to YouTube per day!
- About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]
Challenges: learning with minimal supervision

Less | More

- Unlabeled, multiple objects
- Classes labeled, images cluster
- Classes to objects, pixels not classes

Slide credit: Kristen Grauman

---

This is a pottopod

Slide from Pietro Perona, 2004 Object Recognition workshop

---

Find the pottopod

Slide from Pietro Perona, 2004 Object Recognition workshop
Recognizing flat, textured objects (like books, CD covers, posters)

Frontal face detection

Fingerprint recognition

What kinds of things work best today?

Progress charted by datasets

Slide credit: Kristen Grauman
Evolution of methods

- Hand-crafted models
- 3D geometry
- Hypothesize and align

- Hand-crafted features
- Learned models
- Data-driven

- “End-to-end” learning of features and models*, **

* Labeled data availability
** Architecture design decisions, parameters.

Next

- Sliding window object detection (Faces!)

Supervised classification

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

  "four" 1 1 1 1 1 1 1
  "nine" 1 1 1 1 1 1 1

  Training examples  Novel input

- 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:
  \[
  R(s) = \Pr(4 \rightarrow 9 \mid s) L(4 \rightarrow 9) + \Pr(9 \rightarrow 4 \mid s) L(9 \rightarrow 4)
  \]

- 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.

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

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

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

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

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

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

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

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

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

\[
P(4 \mid x) \cdot L(4 \rightarrow 9) > P(9 \mid x) \cdot L(9 \rightarrow 4)
\]

Probability

Basic probability

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

\[
P(X)
\]

- \( 0 \leq P(X) \leq 1 \)
- \( \int P(X) \, dX = 1 \) or \( \sum P(X) = 1 \)

Continuous \( X \) Discrete \( X \)

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

Example: learning skin colors

- We can represent a class-conditional density using a histogram (a “non-parametric” distribution)
Example: learning skin colors

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

\[
P(x | \text{skin}) \quad \text{Feature } x = \text{Hue}
\]

\[
P(x | \text{not skin}) \quad \text{Feature } x = \text{Hue}
\]

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?

Bayes rule

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

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

Where does the prior come from?

Why use a prior?

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

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)
General classification
This same procedure applies in more general circumstances
- More than two classes
- More than one dimension

Example: face detection
- Here, X is a image region
  - dimension = # pixels
  - each face can be thought of as a point in a high dimensional space


Next
- Sliding window object detection (Faces!)