Machine Learning Problems

- Supervised Learning:
  - Classification or categorization
  - Regression

- Unsupervised Learning:
  - Clustering
  - Dimensionality reduction
The machine learning framework

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

\[
\begin{align*}
f(\text{apple}) &= \text{“apple”} \\
f(\text{tomato}) &= \text{“tomato”} \\
f(\text{cow}) &= \text{“cow”}
\end{align*}
\]
Learning a classifier

Given some set of features with corresponding labels, learn a function to predict the labels from the features
Generalization

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

• Components of generalization error
  – **Bias**: how much the average model over all training sets differ from the true model?
    • Error due to inaccurate assumptions/simplifications made by the model.
  – **Variance**: how much models estimated from different training sets differ from each other.

• **Underfitting**: model is too “simple” to represent all the relevant class characteristics
  – High bias (few degrees of freedom) and low variance
  – High training error and high test error

• **Overfitting**: model is too “complex” and fits irrelevant characteristics (noise) in the data
  – Low bias (many degrees of freedom) and high variance
  – Low training error and high test error
Bias-Variance Trade-off

- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).
Bias-variance tradeoff

- **Underfitting**: Low Bias, Low Variance
- **Overfitting**: High Bias, High Variance

Error vs. Complexity

Slide credit: D. Hoiem
Bias-variance tradeoff

- Many training examples:
  - Low Bias
  - Low Variance
- Few training examples:
  - High Bias
  - High Variance

Test Error

Complexity

High Bias
Low Variance

Low Bias
High Variance

Slide credit: D. Hoiem
Effect of Training Size

Fixed prediction model

Error

Generalization Error

Number of Training Examples
Remember...

• No classifier is inherently better than any other: you need to make assumptions to generalize

• Three kinds of error
  – Inherent: unavoidable
  – Bias: due to over-simplifications
  – Variance: due to inability to perfectly estimate parameters from limited data
How to reduce variance?

• Choose a simpler classifier

• Regularize the parameters

• Get more training data
Very brief tour of some classifiers

- K-nearest neighbor
- SVM
- Boosted Decision Trees
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- RBMs
- Etc.
Generative vs. Discriminative Classifiers

Generative Models
• 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

Discriminative Models
• 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

Slide credit: D. Hoiem
Classification

• Assign input vector to one of two or more classes

• Any decision rule divides input space into decision regions separated by decision boundaries
Nearest Neighbor Classifier

- Assign label of nearest training data point to each test data point

Voronoi partitioning of feature space for two-category 2D and 3D data

Source: D. Lowe
K-nearest neighbor
1-nearest neighbor
3-nearest neighbor
5-nearest neighbor
Using K-NN

• Simple, a good one to try first

• With infinite examples, 1-NN provably has error that is at most twice Bayes optimal error
Classifiers: Linear SVM

- Find a *linear function* to separate the classes:

\[ f(x) = \text{sgn}(w \cdot x + b) \]
Classifiers: Linear SVM

• Find a linear function to separate the classes:

\[ f(x) = \text{sgn}(w \cdot x + b) \]
Classifiers: Linear SVM

• Find a *linear function* to separate the classes:

\[ f(x) = \text{sgn}(w \cdot x + b) \]
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: 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
What to remember about classifiers

• No free lunch: machine learning algorithms are tools, not dogmas

• Try simple classifiers first

• Better to have smart features and simple classifiers than simple features and smart classifiers

• Use increasingly powerful classifiers with more training data (bias-variance tradeoff)

Slide credit: D. Hoiem
Machine Learning Considerations

• 3 important design decisions:
  1) What data do I use?
  2) How do I represent my data (what feature)?
  3) What classifier / regressor / machine learning tool do I use?

• These are in decreasing order of importance

• Deep learning addresses 2 and 3 simultaneously (and blurs the boundary between them).

• You can take the representation from deep learning and use it with any classifier.
Recognition: Overview and History

Slides from Lana Lazebnik, Fei-Fei Li, Rob Fergus, Antonio Torralba, and Jean Ponce
How many visual object categories are there?

~10,000 to 30,000

Biederman 1987
~10,000 to 30,000
OBJECTS

ANIMALS

.....

VERTEBRATE

MAMMALS

TAPIR

BOAR

BIRDS

GROUSE

INANIMATE

NATURAL

MAN-MADE

PLANTS

MAN-MADE

NATURAL

CAMERA
Specific recognition tasks
Scene categorization or classification

• outdoor/indoor
• city/forest/factory/etc.
Image annotation / tagging / attributes

- street
- people
- building
- mountain
- tourism
- cloudy
- brick
- ...

Svetlana Lazebnik
Object detection

- find pedestrians

Svetlana Lazebnik
Image parsing / semantic segmentation

- sky
- mountain
- building
- tree
- banner
- street lamp
- market
- people

Svetlana Lazebnik
Scene understanding?
Recognition is all about modeling variability

Variability: Camera position
Illumination
Shape parameters

Within-class variations?
Within-class variations
History of ideas in recognition

• 1960s – early 1990s: the geometric era
Variability:

- Camera position
- Illumination

Alignment

Shape: assumed known

Roberts (1965); Lowe (1987); Faugeras & Hebert (1986); Grimson & Lozano-Perez (1986); Huttenlocher & Ullman (1987)
Recall: Alignment

- Alignment: fitting a model to a transformation between pairs of features (matches) in two images

\[ \sum_{i} \text{residual}(T(x_i), x'_i) \]
Recognition as an alignment problem: Block world


**Fig. 1.** A system for recognizing 3-d polyhedral scenes. a) L.G. Roberts. b) A blocks world scene. c) Detected edges using a $2\times2$ gradient operator. d) A 3-d polyhedral description of the scene, formed automatically from the single image. e) The 3-d scene displayed with a viewpoint different from the original image to demonstrate its accuracy and completeness. (b) - e) are taken from [64] with permission MIT Press.)

Representing and recognizing object categories is harder...

ACRONYM (Brooks and Binford, 1981)
Binford (1971), Nevatia & Binford (1972), Marr & Nishihara (1978)
Recognition by components

Biederman (1987)

Primitives (geons)

- Cube
  - Straight Edge
  - Straight Axis
  - Constant

- Wedge
  - Straight Edge
  - Straight Axis
  - Expanded

- Pyramid
  - Straight Edge
  - Straight Axis
  - Expanded

- Cylinder
  - Curved Edge
  - Straight Axis
  - Constant

- Barrel
  - Curved Edge
  - Straight Axis
  - Expanded
  - Cont

- Arch
  - Straight Edge
  - Curved Axis
  - Constant

- Cone
  - Curved Edge
  - Straight Axis
  - Expanded

- Expanded Cylinder
  - Curved Edge
  - Straight Axis
  - Expanded

- Handle
  - Curved Edge
  - Curved Axis
  - Constant

- Expanded Handle
  - Curved Edge
  - Curved Axis
  - Expanded

Objects


Svetlana Lazebnik
Generalized cylinders
Ponce et al. (1989)

Zisserman et al. (1995)

Forsyth (2000)

General shape primitives?
History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
Empirical models of image variability

Appearance-based techniques

Turk & Pentland (1991); Murase & Nayar (1995); etc.
Eigenfaces (Turk & Pentland, 1991)

<table>
<thead>
<tr>
<th>Experimental Condition</th>
<th>Correct/Unknown Recognition Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lighting</td>
<td>Orientation</td>
</tr>
<tr>
<td>Forced classification</td>
<td>96/0</td>
</tr>
<tr>
<td>Forced 100% accuracy</td>
<td>100/19</td>
</tr>
<tr>
<td>Forced 20% unknown rate</td>
<td>100/20</td>
</tr>
</tbody>
</table>
Color Histograms

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- 1990s – present: sliding window approaches
Sliding window approaches
Sliding window approaches

- Turk and Pentland, 1991
- Belhumeur, Hespanha, & Kriegman, 1997
- Schneiderman & Kanade, 2004
- Viola and Jones, 2000

- Schneiderman & Kanade, 2004
- Argawal and Roth, 2002
- Poggio et al. 1993
History of ideas in recognition

• 1960s – early 1990s: the geometric era
• 1990s: appearance-based models
• Mid-1990s: sliding window approaches
• Late 1990s: local features
Local features for object instance recognition

Large-scale image search
Combining local features, indexing, and spatial constraints
Large-scale image search
Combining local features, indexing, and spatial constraints

Philbin et al. ‘07
Large-scale image search
Combining local features, indexing, and spatial constraints

Google Goggles in Action
Click the icons below to see the different ways Google Goggles can be used.

Available on phones that run Android 1.6+ (i.e. Donut or Eclair)

Svetlana Lazebnik
History of ideas in recognition

• 1960s – early 1990s: the geometric era
• 1990s: appearance-based models
• Mid-1990s: sliding window approaches
• Late 1990s: local features
• Early 2000s: parts-and-shape models
Parts-and-shape models

• Model:
  – Object as a set of parts
  – Relative locations between parts
  – Appearance of part

Figure from [Fischler & Elschlager 73]
Constellation models

Pictorial structure model

Fischler and Elschlager(73), Felzenszwalb and Huttenlocher(00)

\[ \Pr(P_{\text{tor}}, P_{\text{arm}}, \ldots | \text{Im}) \propto \prod_{i,j} \Pr(P_i | P_j) \prod_i \Pr(\text{Im}(P_i)) \]

part geometry

part appearance
Discriminatively trained part-based models

History of ideas in recognition

• 1960s – early 1990s: the geometric era
• 1990s: appearance-based models
• Mid-1990s: sliding window approaches
• Late 1990s: local features
• Early 2000s: parts-and-shape models
• Mid-2000s: bags of features
Bag-of-features models

Svetlana Lazebnik
Bag-of-features models

Object → Bag of ‘words’
Objects as texture

- All of these are treated as being the same

- No distinction between foreground and background: scene recognition?
Origin 1: Texture recognition

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters

Origin 1: Texture recognition

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models


US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/
Bag-of-features steps

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”
1. Feature extraction

- Regular grid or interest regions
1. Feature extraction

Compute descriptor

Normalize patch

Detect patches

Slide credit: Josef Sivic
1. Feature extraction

Slide credit: Josef Sivic
2. Learning the visual vocabulary

Slide credit: Josef Sivic
2. Learning the visual vocabulary

Clustering

Slide credit: Josef Sivic
2. Learning the visual vocabulary

Visual vocabulary

Clustering

Slide credit: Josef Sivic
Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be “universal”

- The codebook is used for quantizing features
  - A vector quantizer takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word
Example codebook

Appearance codebook

Source: B. Leibe
Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting

- Computational efficiency
  - Vocabulary trees
    (Nister & Stewenius, 2006)
But what about layout?

All of these images have the same color histogram
Spatial pyramid

Compute histogram in each spatial bin
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)
Scene category dataset

Multi-class classification results
(100 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features</th>
<th>Strong features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(vocabulary size: 16)</td>
<td>(vocabulary size: 200)</td>
</tr>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0 ((1 \times 1))</td>
<td>45.3 ±0.5</td>
<td>72.2 ±0.6</td>
</tr>
<tr>
<td>1 ((2 \times 2))</td>
<td>53.6 ±0.3</td>
<td>56.2 ±0.6</td>
</tr>
<tr>
<td>2 ((4 \times 4))</td>
<td>61.7 ±0.6</td>
<td>64.7 ±0.7</td>
</tr>
<tr>
<td>3 ((8 \times 8))</td>
<td>63.3 ±0.8</td>
<td>\textbf{66.8} ±0.6</td>
</tr>
</tbody>
</table>
Caltech101 dataset


Multi-class classification results (30 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (16)</th>
<th>Strong features (200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0</td>
<td>15.5 ±0.9</td>
<td>26.6 ±1.3</td>
</tr>
<tr>
<td>1</td>
<td>31.4 ±1.2</td>
<td>32.8 ±1.3</td>
</tr>
<tr>
<td>2</td>
<td>47.2 ±1.1</td>
<td>49.3 ±1.4</td>
</tr>
<tr>
<td>3</td>
<td>52.2 ±0.8</td>
<td>54.0 ±1.1</td>
</tr>
</tbody>
</table>
Bags of features for action recognition

Space-time interest points

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models
- Mid-2000s: bags of features
- Present trends: combination of local and global methods, context, *deep learning*