Computer Vision
CSC5930/CSC9010

Instructor: Dr. Edward Kim
Probability
Independence

• The probability of independent events $A$, $B$ and $C$ is given by:

$$P(A,B,C) = P(A)P(B)P(C)$$

A and $B$ are independent, if knowing that $A$ has happened does not say anything about $B$ happening.
Dependent

• The probability of dependent events $A$, $B$ is given by:

$$P(A,B) = P(A \mid B)P(B)$$
• What is the probability that two female students will be selected at random to participate in a certain research project, from a class of seven males and three female students?
• – Define the events:

• A – the first student selected is a female

• B – the second student selected is a female

• \[ P(A \text{ and } B) = P(A)P(B|A) = \frac{3}{10}\frac{2}{9} = \frac{6}{90} = 0.067 \]
Bayes Theorem

- Provides a way to convert \textit{a-priori} probabilities to \textit{a-posteriori} probabilities:

\[ P(A|B)P(B) = P(B|A)P(A) \]
Bayes

\[ P(A|B) = \frac{P(B|A) P(A)}{P(B)} \]

- **Posterior**
- **Likelihood**
- **Prior**
- **Evidence**

**Special case**
- Binary partition

**General case**
- J way partition

\[ P(A|B) = \frac{P(B|A) P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)} \]

\[ \Rightarrow P(A_i|B) = \frac{P(B|A_i) P(A_i)}{\sum_j P(B|A_j) P(A_j)} \]
Bayes Example

• 1% of women have breast cancer (and therefore 99% do not).

• 80% of mammograms detect breast cancer when it is there (and therefore 20% miss it).

• 9.6% of mammograms detect breast cancer when it’s not there (and therefore 90.4% correctly return a negative result).
A test comes back positive... what are the chances that you have cancer? 80%? 99%? 1%?

<table>
<thead>
<tr>
<th></th>
<th>Cancer (1%)</th>
<th>No Cancer (99%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Pos</td>
<td>80%</td>
<td>9.6%</td>
</tr>
<tr>
<td>Test Neg</td>
<td>20%</td>
<td>90.4%</td>
</tr>
</tbody>
</table>
What differentiates the two?
Tying to bayes

• The sea bass/salmon example

  – State of nature, prior

    • State of nature is a random variable

    • The catch of salmon and sea bass is equiprobable

      – \( P(\omega_1) = P(\omega_2) \) (uniform priors)

      – \( P(\omega_1) + P(\omega_2) = 1 \) (exclusivity and exhaustivity)
• Decision rule with only the prior information
  – Decide $\omega_1$ if $P(\omega_1) > P(\omega_2)$ otherwise decide $\omega_2$

• Use of the class – conditional information

• $P(x \mid \omega_1)$ and $P(x \mid \omega_2)$ describe the difference in lightness between populations of sea and salmon
• Decision given the posterior probabilities

X is an observation for which:

if \( P(\omega_1 \mid x) > P(\omega_2 \mid x) \) \( \Rightarrow \) True state of nature = \( \omega_1 \)
if \( P(\omega_1 \mid x) < P(\omega_2 \mid x) \) \( \Rightarrow \) True state of nature = \( \omega_2 \)

e.g.

\[
P(salmon \mid length) > P(seabass \mid length) \quad \text{Classify as salmon}
\]
\[
P(salmon \mid length) < P(seabass \mid length) \quad \text{Classify as seabass}
\]
• KNN makes a prediction based upon the class membership of the closest sample in the training set
KNN

- Simple + works well
- Empirical error of 1-NN is 0!
- Lazy learning – nothing at train time
Distance functions

**Euclidean**

\[ \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \]

**Manhattan**

\[ \sum_{i=1}^{k} |x_i - y_i| \]

**Minkowski**

\[ \left( \sum_{i=1}^{k} (|x_i - y_i|^q)^{1/q} \right) \]
SVM (Support Vector Machines)

• Pattern Recognition

• Object classification / detection
Human Detection

Navneet Dalal and Bill Triggs “Histograms of Oriented Gradients for Human Detection” CVPR05
Blocks, Cells

- 16x16 blocks of 50% overlap.
  - 7x15 = 105 blocks in total

- Each block should consist of 2x2 cells with size 8x8.
Final Feature Vector

- Concatenate histograms
  - Make it a 1D vector of length 3780.

- Visualization
Application: Pedestrian detection
Averaged positive examples
Usage

• Train using positive and negative examples

• SVM “learns” the data

• Use the classifier on unknown examples
**Algorithm**

**Training (Learning)**

- Represent each example window by a HOG feature vector

\[ x_i \in \mathbb{R}^d, \]

- Train a SVM classifier

**Testing (Detection)**

- Sliding window classifier

\[ f(x) = w^T x + b \]
Goal: Detect all instances of objects
Sliding window detection
What the Detector Sees
Linear Classifiers

Label $y$:
- denotes $+1$
- denotes $-1$

$$f(x,w,b) = \text{sign}(wx + b)$$

How would you classify this data?
Linear Classifiers

\[ f(x, w, b) = \text{sign}(wx + b) \]

How would you classify this data?

Slide Credits: Guo-Jun Qi
Linear Classifiers

\[ f(x, w, b) = \text{sign}(wx + b) \]

How would you classify this data?

\* denotes +1
\circ denotes -1

Slide Credits: Guo-Jun Qi
Linear Classifiers

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

- \( \bullet \) denotes +1
- \( \circ \) denotes -1

Any of these would be fine..

..but which is the best?

Slide Credits: Guo-Jun Q1
Classifier Margin

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a data point.

Slide Credits: Guo-Jun QI
Maximum Margin

1. Maximizing the margin makes sense according to intuition
2. Implies that only support vectors are important; other training examples can be discarded without affecting the training result.

• denotes +1  
• denotes -1

Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

Parameters

\[ y_{est} \]

\[ + b \]

Slide Credits: Guo-Jun Qi
Maximum Margin

\[ f(x, w, b) = \text{sign}(w \cdot x + b) \]

- **denotes +1**
- **denotes -1**

Keeping only support vectors will not change the maximum margin classifier.

- Robust to the small changes (noise) in non-support vectors

Linear SVM

Slide Credits: Guo-Jun Qi
Dataset with noise

- Hard Margin: So far, all data points are classified correctly
- No training error
- What if the training set is noisy?
What should our quadratic optimization criterion be? We expect $\zeta_i$ to be small.

$$\Phi(w) = \frac{1}{2} w^T w + C \sum \zeta_i$$
Bag of Visual Words
Image classification

- Image classification: assigning a class label to the image

```none
Car: present
Cow: present
Bike: not present
Horse: not present
...```

![Car and cow in a field](image)
• Image classification: assigning a class label to the image
  
  Car: present
  Cow: present
  Bike: not present
  Horse: not present
  ...

• Object localization: define the location and the category
  
  Location
  Category
Difficulties: within class variations
Image classification

• Given
  Positive training images containing an object class

  Negative training images that don’t

• Classify
  A test image as to whether it contains the object class or not
• Texture is characterized by the repetition of basic elements or *textons*
Bag-of-features – Origin: texture recognition

- Texture is characterized by the repetition of basic elements or textons
Bag-of-features – Origin: texture recognition
Bag of Words Model

Documents | Doc-1 | Doc-2 | Doc-3 | Doc-4
--- | --- | --- | --- | ---
Words
Bank | 1 | 1 | 0 | 0
Loan | 1 | 0 | 0 | 3
Water | 0 | 2 | 2 | 1
Farmer | 0 | 0 | 0 | 1

Documents | Doc-1 | Doc-2 | Doc-3 | Doc-4
--- | --- | --- | --- | ---
Words
Bank | 0.5 | 0.33 | 0 | 0
Loan | 0.5 | 0 | 0 | 0.6
Water | 0 | 0.66 | 1 | 0.2
Farmer | 0 | 0 | 0 | 0.2
Bag of Visual Words model
Bag of Visual Words model

Local Patches

Descriptors

Clustering

Generate the visual vocabulary

Words
Bag of Visual Words model

Represent an image as a histogram or bag of words
Bag-of-features – Origin: bag-of-words (text)

- Orderless document representation: frequencies of words from a dictionary
- Classification to determine document categories

Bag-of-words

<table>
<thead>
<tr>
<th></th>
<th>Common</th>
<th>People</th>
<th>Sculpture</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d4</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

...
Bag-of-features for image classification

Step 1
- Extract regions
- Compute descriptors

Step 2
- Find clusters and frequencies

Step 3
- Compute distance matrix
- Classification

SVM
Step 1: feature extraction

- Detect Interest Points
  - SIFT
  - Harris
  - Dense (take every nth pixel as interest point)
- Compute Descriptor around each interest point
  - SIFT
  - HOG
Dense features
Bag-of-features for image classification

Step 1:
- Extract regions
- Compute descriptors

Step 2:
- Find clusters and frequencies
- Compute distance matrix

Step 3:
- Classification
- SVM
Step 2: Quantization

Visual vocabulary

Clustering
Step 2: Quantization

- Cluster descriptors
  - K-means

- Assign each visual word to a cluster

- Build frequency histogram
Choose $k$ data points to act as cluster centers

Until the cluster centers are unchanged

Allocate each data point to cluster whose center is nearest

Replace the cluster centers with the mean of the elements in their clusters.

end

Algorithm 16.5: Clustering by K-Means
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance
K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance
## Examples for visual words

<table>
<thead>
<tr>
<th>Category</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplanes</td>
<td><img src="image1.png" alt="Airplanes Images" /></td>
</tr>
<tr>
<td>Motorbikes</td>
<td><img src="image2.png" alt="Motorbikes Images" /></td>
</tr>
<tr>
<td>Faces</td>
<td><img src="image3.png" alt="Faces Images" /></td>
</tr>
<tr>
<td>Wild Cats</td>
<td><img src="image4.png" alt="Wild Cats Images" /></td>
</tr>
<tr>
<td>Leaves</td>
<td><img src="image5.png" alt="Leaves Images" /></td>
</tr>
<tr>
<td>People</td>
<td><img src="image6.png" alt="People Images" /></td>
</tr>
<tr>
<td>Bikes</td>
<td><img src="image7.png" alt="Bikes Images" /></td>
</tr>
</tbody>
</table>
- each image is represented by a vector, typically 1000-4000 dimension,
Bag-of-features for image classification

- **Step 1**: Extract regions from each image.
- **Step 2**: Compute descriptors and find clusters and frequencies.
- **Step 3**: Compute distance matrix and classification using SVM.
Training data

Vectors are histograms, one from each training image

positive

Train classifier, e.g. SVM
Examples for misclassified images

Books- misclassified into faces, faces, buildings

Buildings- misclassified into faces, trees, trees

Cars- misclassified into buildings, phones, phones
Bag of visual words summary

- **Advantages:**
  - largely unaffected by position and orientation of object in image
  - fixed length vector irrespective of number of detections
  - very successful in classifying images according to the objects they contain

- **Disadvantages:**
  - no explicit use of configuration of visual word positions
  - poor at localizing objects within an image
Evaluation Metrics

\[
\text{precision} = \frac{\text{GT} \cap \text{RM}}{\text{RM}} = \frac{\text{TP}}{\text{RM}}
\]

\[
\text{recall} = \frac{\text{GT} \cap \text{RM}}{\text{GT}} = \frac{\text{TP}}{\text{GT}}
\]

\[
\text{precision} = \frac{\text{GT} \cap \text{RM}}{\text{RM}} = \frac{\text{TP}}{\text{FP} + \text{TP}}
\]

\[
\text{recall} = \frac{\text{GT} \cap \text{RM}}{\text{GT}} = \frac{\text{TP}}{\text{TP} + \text{FN}}
\]

Ground Truth (GT)

Results of Method (RM)

True Positives (TP)

True Negatives (TN)

False Negatives (FN)

False Positives (FP)
Average Precision [TREC] averages precision over the entire range of recall

- Curve interpolated to reduce influence of “outliers”

- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall
Spatial pyramid matching

- Add spatial information to the bag-of-features
- Perform matching in 2D image space

[Lazebnik, Schmid & Ponce, CVPR 2006]
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution

level 0
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution