Image classification
Course announcements

- Homework 5 is available online.
  - Any questions about the homework?
  - How many of you have looked at/started/finished homework 5?

- Extra late day awarded to everyone.
  - You can use this for any homework you want, including retroactively for older homeworks.

- No lecture on Wednesday.

- Extra office hours by Yannis on Friday, 1-3 pm.
  - This are in addition to the usual office hours between 3-5 pm.
  - These will take place in the graphics lounge and/or Smith 225.

- How many of you went to Angela Dai’s talk?

- Vote on Piazza for your favorite faculty candidates!
Overview of today’s lecture

- Bag-of-words.
- K-means clustering.
- Classification.
- K nearest neighbors.
- Naïve Bayes.
- Support vector machine.
Slide credits

Most of these slides were adapted from:

• Kris Kitani (16-385, Spring 2017).
• Noah Snavely (Cornell University).
• Fei-Fei Li (Stanford University).
Image Classification

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

cat
Image Classification: Problem
Data-driven approach

- Collect a database of images with labels
- Use ML to train an image classifier
- Evaluate the classifier on test images

Example training set
Bag of words
What object do these parts belong to?
Some local features are very informative.

An object as a collection of local features (bag-of-features)
- deals well with occlusion
- scale invariant
- rotation invariant
(not so) crazy assumption

spatial information of local features can be ignored for object recognition (i.e., verification)
Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

Works pretty well for image-level classification

<table>
<thead>
<tr>
<th>class</th>
<th>bag of features</th>
<th>bag of features</th>
<th>Parts-and-shape model</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplanes</td>
<td>98.8</td>
<td>97.1</td>
<td>90.2</td>
</tr>
<tr>
<td>cars (rear)</td>
<td>98.3</td>
<td>98.6</td>
<td>90.3</td>
</tr>
<tr>
<td>cars (side)</td>
<td>95.0</td>
<td>87.3</td>
<td>88.5</td>
</tr>
<tr>
<td>faces</td>
<td>100</td>
<td>99.3</td>
<td>96.4</td>
</tr>
<tr>
<td>motorbikes</td>
<td>98.5</td>
<td>98.0</td>
<td>92.5</td>
</tr>
<tr>
<td>spotted cats</td>
<td>97.0</td>
<td>—</td>
<td>90.0</td>
</tr>
</tbody>
</table>

CalTech6 dataset
Bag-of-features

represent a data item (document, texture, image) as a histogram over features

an old idea
(e.g., texture recognition and information retrieval)
Texture recognition

Universal texton dictionary

histogram
Vector Space Model

G. Salton. 'Mathematics and Information Retrieval' Journal of Documentation, 1979

The Newspaper

DARPA Selects Carnegie Mellon

Tartan Tim

Bio-Inspired Robotic Device

http://www.fodey.com/generators/newspaper/snippet.asp
A document (datapoint) is a vector of counts over each word (feature)

\[ \mathbf{v}_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})] \]

\[ n(\cdot) \] counts the number of occurrences

What is the similarity between two documents?

just a histogram over words
A document (datapoint) is a vector of counts over each word (feature)

\[ \mathbf{v}_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})] \]

\( n(\cdot) \) counts the number of occurrences

What is the similarity between two documents?

Use any distance you want but the cosine distance is fast.

\[ d(\mathbf{v}_i, \mathbf{v}_j) = \cos \theta = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \]
but not all words are created equal
TF-IDF

Term Frequency Inverse Document Frequency

\[ \mathbf{v}_d = [n(w_{1,d}) \ n(w_{2,d}) \ \cdots \ n(w_{T,d})] \]

weigh each word by a heuristic

\[ \mathbf{v}_d = [n(w_{1,d}) \alpha_1 \ n(w_{2,d}) \alpha_2 \ \cdots \ n(w_{T,d}) \alpha_T] \]

\[ n(w_{i,d}) \alpha_i = n(w_{i,d}) \log \left\{ \frac{D}{\sum_{d'} 1[w_i \in d']} \right\} \]

(term frequency)

(inverse document frequency)

(down-weights common terms)
Standard BOW pipeline
(for image classification)
Dictionary Learning:
Learn Visual Words using clustering

Encode:
build Bags-of-Words (BOW) vectors for each image

Classify:
Train and test data using BOWs
Dictionary Learning: Learn Visual Words using clustering

1. extract features (e.g., SIFT) from images
Dictionary Learning:
Learn Visual Words using clustering

2. Learn visual dictionary (e.g., K-means clustering)
What kinds of features can we extract?
• Regular grid
  • Vogel & Schiele, 2003
  • Fei-Fei & Perona, 2005

• Interest point detector
  • Csurka et al. 2004
  • Fei-Fei & Perona, 2005
  • Sivic et al. 2005

• Other methods
  • Random sampling (Vidal-Naquet & Ullman, 2002)
  • Segmentation-based patches (Barnard et al. 2003)
Compute SIFT descriptor
[Lowe’99]

Normalize patch

Detect patches
[Mikojaczyk and Schmid ’02]
[Mata, Chum, Urban & Pajdla, ’02]
[Sivic & Zisserman, ’03]
How do we learn the dictionary?
K-means clustering
1. Select initial centroids at random
1. Select initial centroids at random

2. Assign each object to the cluster with the nearest centroid.
1. Select initial centroids at random

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it (go to 2)
1. Select initial centroids at random

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it (go to 2)

2. Assign each object to the cluster with the nearest centroid.
1. Select initial centroids at random

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it (go to 2)

Repeat previous 2 steps until no change
K-means Clustering

Given $k$:

1. Select initial centroids at random.

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it.

4. Repeat previous 2 steps until no change.
From what data should I learn the dictionary?
From what **data** should I learn the dictionary?

- Dictionary can be learned on separate training set
- Provided the training set is sufficiently representative, the dictionary will be “universal”
Example visual dictionary
Example dictionary

Appearance codebook

Source: B. Leibe
Another dictionary

Appearance codebook

Source: B. Leibe
Dictionary Learning:
Learn Visual Words using clustering

Encode:
build Bags-of-Words (BOW) vectors for each image

Classify:
Train and test data using BOWs
1. Quantization: image features get associated to a visual word (nearest cluster center)

**Encode:**
build Bags-of-Words (BOW) vectors for each image
Encode:
build Bags-of-Words (BOW) vectors for each image

2. Histogram: count the number of visual word occurrences
frequency

codewords
Dictionary Learning:
Learn Visual Words using clustering

Encode:
build Bags-of-Words (BOW) vectors for each image

Classify:
Train and test data using BOWs
K nearest neighbors
Distribution of data from two classes
Which class does $q$ belong too?
Distribution of data from two classes

Look at the neighbors
K-Nearest Neighbor (KNN) Classifier

Non-parametric pattern classification approach
Consider a two class problem where each sample consists of two measurements \((x,y)\).

For a given query point \(q\), assign the class of the nearest neighbor

Compute the \(k\) nearest neighbors and assign the class by majority vote.
Nearest Neighbor is competitive

MNIST Digit Recognition
- Handwritten digits
- 28x28 pixel images: d = 784
- 60,000 training samples
- 10,000 test samples

Yann LeCunn

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear classifier (1-layer NN)</td>
<td>12.0</td>
</tr>
<tr>
<td>K-nearest-neighbors, Euclidean</td>
<td>5.0</td>
</tr>
<tr>
<td>K-nearest-neighbors, Euclidean, deskewed</td>
<td>2.4</td>
</tr>
<tr>
<td>K-NN, Tangent Distance, 16x16</td>
<td>1.1</td>
</tr>
<tr>
<td>K-NN, shape context matching</td>
<td>0.67</td>
</tr>
<tr>
<td>1000 RBF + linear classifier</td>
<td>3.6</td>
</tr>
<tr>
<td>SVM deg 4 polynomial</td>
<td>1.1</td>
</tr>
<tr>
<td>2-layer NN, 300 hidden units</td>
<td>4.7</td>
</tr>
<tr>
<td>2-layer NN, 300 HU, [deskewing]</td>
<td>1.6</td>
</tr>
<tr>
<td>LeNet-5, [distortions]</td>
<td>0.8</td>
</tr>
<tr>
<td>Boosted LeNet-4, [distortions]</td>
<td>0.7</td>
</tr>
</tbody>
</table>
What is the best distance metric between data points?

- Typically Euclidean distance
- Locality sensitive distance metrics
- Important to normalize. Dimensions have different scales

How many K?

- Typically k=1 is good
- Cross-validation (try different k!)
Distance metrics

\[ D(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_N - y_N)^2} \quad \text{Euclidean} \]

\[ D(x, y) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{x_1 y_1 + \cdots + x_N y_N}{\sqrt{\sum_n x_n^2} \sqrt{\sum_n y_n^2}} \quad \text{Cosine} \]

\[ D(x, y) = \frac{1}{2} \sum_n \frac{(x_n - y_n)^2}{(x_n + y_n)} \quad \text{Chi-squared} \]
Choice of distance metric

- Hyperparameter

L1 (Manhattan) distance

\[ d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p| \]

L2 (Euclidean) distance

\[ d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2} \]

- Two most commonly used special cases of p-norm

\[ \|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}} \quad p \geq 1, x \in \mathbb{R}^n \]
Visualization: L2 distance
CIFAR-10 and NN results

Example dataset: CIFAR-10
10 labels
50,000 training images
10,000 test images.

For every test image (first column), examples of nearest neighbors in rows
k-nearest neighbor

- Find the $k$ closest points from training data
- Labels of the $k$ points “vote” to classify
Hyperparameters

• What is the best distance to use?
• What is the best value of k to use?

• i.e., how do we set the hyperparameters?

• Very problem-dependent
• Must try them all and see what works best
Try out what hyperparameters work best on test set.
Trying out what hyperparameters work best on test set:
Very bad idea. The test set is a proxy for the generalization performance!
Use only **VERY SPARINGLY**, at the end.
Validation

Validation data
use to tune hyperparameters
evaluate on test set ONCE at the end
Cross-validation

cycle through the choice of which fold is the validation fold, average results.
Example of 5-fold cross-validation for the value of $k$.

Each point: single outcome.

The line goes through the mean, bars indicated standard deviation

(Seems that $k \approx 7$ works best for this data)
How to pick hyperparameters?

- Methodology
  - Train and test
  - Train, validate, test

- Train for original model
- Validate to find hyperparameters
- Test to understand generalizability
Pros

• simple yet effective

Cons

• search is expensive (can be sped-up)
• storage requirements
• difficulties with high-dimensional data
kNN -- Complexity and Storage

• N training images, M test images

• Training: $O(1)$
• Testing: $O(MN)$

• Hmm...
  – Normally need the opposite
  – Slow training (ok), fast testing (necessary)
k-Nearest Neighbor on images never used.

- terrible performance at test time
- distance metrics on level of whole images can be very unintuitive

(original) (shifted) (messed up) (darkened)

(all 3 images have same L2 distance to the one on the left)
Naïve Bayes
Distribution of data from two classes

Which class does $q$ belong to?
Distribution of data from two classes

- Learn parametric model for each class
- Compute probability of query

$q$
This is called the posterior. the probability of a class $z$ given the observed features $X$

$$p(z|X)$$

For classification, $z$ is a discrete random variable (e.g., car, person, building)

$X$ is a set of observed features (e.g., features from a single image)

(it’s a function that returns a single probability value)
This is called the posterior: the probability of a class $z$ given the observed features $X$

$$p(z|x_1, \ldots, x_N)$$

For classification, $z$ is a discrete random variable (e.g., car, person, building)

Each $x$ is an observed feature (e.g., visual words)

(it’s a function that returns a single probability value)
Recall:

The posterior can be decomposed according to Bayes’ Rule

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

In our context…

$$p(z|x_1, \ldots, x_N) = \frac{p(x_1, \ldots, x_N|z)p(z)}{p(x_1, \ldots, x_N)}$$
The naive Bayes’ classifier is solving this optimization

$$\hat{z} = \arg \max_{z \in Z} p(z|X)$$

MAP (maximum a posteriori) estimate

$$\hat{z} = \arg \max_{z \in Z} \frac{p(X|z)p(z)}{p(X)}$$

Bayes’ Rule

$$\hat{z} = \arg \max_{z \in Z} p(X|z)p(z)$$

Remove constants

To optimize this…we need to compute this

Compute the likelihood…
A naive Bayes’ classifier assumes all features are **conditionally independent**.

\[
p(x_1, \ldots, x_N | z) = p(x_1 | z)p(x_2, \ldots, x_N | z)
\]
\[
= p(x_1 | z)p(x_2 | z)p(x_3, \ldots, x_N | z)
\]
\[
= p(x_1 | z)p(x_2 | z) \cdots p(x_N | z)
\]

Recall:

\[
p(x, y) = p(x | y)p(y)
\]
\[
p(x, y) = p(x)p(y)
\]
To compute the MAP estimate

Given (1) a set of known parameters

\[ p(z) \quad p(x|z) \]

(2) observations \[ \{x_1, x_2, \ldots, x_N\} \]

Compute which \( z \) has the largest probability

\[ \hat{z} = \arg \max_{z \in Z} p(z) \prod_{n} p(x_n|z) \]
p(X|z) = \prod_v p(x_v|z)^{c(w_v)}

= (0.09)^1 (0.55)^6 \cdots (0.09)^1

Numbers get really small so use log probabilities

\log p(X|z = 'grandchallenge') = -2.42 - 3.68 - 3.43 - 2.42 - 0.07 - 0.07 - 0.07 - 2.42 = -14.58

\log p(X|z = 'softrobot') = -7.63 - 9.37 - 15.18 - 2.97 - 0.02 - 0.01 - 0.02 - 2.27 = -37.48

* typically add pseudo-counts (0.001)
** this is an example for computing the likelihood, need to multiply times prior to get posterior
* typically add pseudo-counts (0.001)
** this is an example for computing the likelihood, need to multiply times prior to get posterior
Support Vector Machine
Image Classification

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

arrow cat
Score function

class scores
Define a score function

$$f(x_i, W, b) = Wx_i + b$$

data (histogram)

"weights"

"bias vector"

"parameters"

class scores
Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

Convert image to histogram representation

\[
W = \begin{bmatrix}
0.2 & -0.5 & 0.1 & 2.0 \\
1.5 & 1.3 & 2.1 & 0.0 \\
0 & 0.25 & 0.2 & -0.3 \\
\end{bmatrix}
\]

\[
x_i = \begin{bmatrix}
56 \\
231 \\
24 \\
\end{bmatrix}
\]

\[
b = \begin{bmatrix}
1.1 \\
3.2 \\
-1.2 \\
\end{bmatrix}
\]

\[
f(x_i; W, b) = -96.8\text{ (cat score)} + 437.9\text{ (dog score)} + 61.95\text{ (ship score)}
\]
Distribution of data from two classes

Which class does q belong to?
Distribution of data from two classes

Learn the decision boundary
First we need to understand hyperplanes...
Hyperplanes (lines) in 2D

\[ w_1 x_1 + w_2 x_2 + b = 0 \]

A line can be written as the dot product plus a bias

\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \]

\( \mathbf{w} \in \mathbb{R}^2 \)

Another version, add a weight 1 and push the bias inside

\[ \mathbf{w} \cdot \mathbf{x} = 0 \]

\( \mathbf{w} \in \mathbb{R}^3 \)
Hyperplanes (lines) in 2D

\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \quad \text{(offset/bias outside)} \quad \mathbf{w} \cdot \mathbf{x} = 0 \quad \text{(offset/bias inside)} \]

\[ w_1 x_1 + w_2 x_2 + b = 0 \]
Hyperplanes (lines) in 2D

\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \quad \text{(offset/bias outside)} \quad \mathbf{w} \cdot \mathbf{x} = 0 \quad \text{(offset/bias inside)} \]

\[ w_1 x_1 + w_2 x_2 + b = 0 \]

**Important property:**
*Free to choose any normalization of \( w \)*

The line

\[ w_1 x_1 + w_2 x_2 + b = 0 \]

and the line

\[ \lambda (w_1 x_1 + w_2 x_2 + b) = 0 \]

define the same line.
What is the distance to origin?

(hint: use normal form)
distance to origin \[ \frac{b}{||w||} \]

\[ w \cdot x + b = 0 \]

scale \[ w \cdot x + b = 0 \] by \[ \frac{1}{||w||} \]

you get the normal form

\[ x \cos \theta + y \sin \theta = \rho \]
What is the distance between two parallel lines? (hint: use distance to origin)
Distance between two parallel lines

\[ \frac{b + 1}{\|w\|} - \frac{b}{\|w\|} = \frac{1}{\|w\|} \]
Now we can go to 3D ...

Hyperplanes (planes) in 3D

What happens if you change $b$?

$\mathbf{w} \cdot \mathbf{x} + b = 0$

what are the dimensions of this vector?

$\frac{b}{\|\mathbf{w}\|}$
Hyperplanes (planes) in 3D

\[ \frac{b + 1}{\|w\|} \]

\[ w \cdot x + b = -1 \]
Hyperplanes (planes) in 3D

What's the distance between these parallel planes?

\[ \mathbf{w} \cdot \mathbf{x} + b = -1 \]
\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \]
\[ \mathbf{w} \cdot \mathbf{x} + b = 1 \]
Hyperplanes (planes) in 3D

\[ \frac{2}{\| \mathbf{w} \|} \]

\[ \mathbf{w} \cdot \mathbf{x} + b = -1 \]

\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \]

\[ \mathbf{w} \cdot \mathbf{x} + b = 1 \]
What's the best $w$?
What's the best $w$?
What’s the best $w$?
What’s the best $w$?
What's the best $w$?
What’s the best $\mathbf{w}$?

Intuitively, the line that is the farthest from all interior points.
What's the best $w$?

Maximum Margin solution: most stable to perturbations of data
What’s the best \( \mathbf{w} \)?

Want a hyperplane that is far away from ‘inner points’
Find hyperplane $\mathbf{w}$ such that ... 

the gap between parallel hyperplanes $\frac{2}{\|\mathbf{w}\|}$ is maximized.
Can be formulated as a maximization problem

\[
\max_{\mathbf{w}} \frac{2}{\|\mathbf{w}\|}
\]

subject to \( \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \) if \( y_i = +1 \)
\( \leq -1 \) if \( y_i = -1 \) for \( i = 1, \ldots, N \)

What does this constraint mean?

Why is it +1 and -1?
Can be formulated as a maximization problem

\[
\max_{\mathbf{w}} \frac{2}{\|\mathbf{w}\|}
\]

subject to \(\mathbf{w} \cdot \mathbf{x}_i + b \geq +1\) if \(y_i = +1\) \(\leq -1\) if \(y_i = -1\) for \(i = 1, \ldots, N\)

Equivalently,

\[
\min_{\mathbf{w}} \|\mathbf{w}\|
\]

subject to \(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1\) for \(i = 1, \ldots, N\)

*Where did the 2 go?*

*What happened to the labels?*
‘Primal formulation’ of a linear SVM

\[ \min_{\mathbf{w}} \|\mathbf{w}\| \]

Objective Function

subject to \[ y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, \ldots, N \]

Constraints

This is a convex quadratic programming (QP) problem
(a unique solution exists)
‘soft’ margin
What’s the best $w$?
What’s the best $w$?

Very narrow margin
Separating cats and dogs

Very narrow margin
Intuitively, we should allow for some misclassification if we can get more robust classification.
What’s the best $w$?

Trade-off between the MARGIN and the MISTAKES
(might be a better solution)
Adding slack variables

\[ \xi_i \geq 0 \]

misclassified point

\[ \frac{\xi_i}{\|w\|} > \frac{2}{\|w\|} \]
‘soft’ margin

**objective**

\[
\min_{\mathbf{w}, \xi} \|\mathbf{w}\|^2 + C \sum_{i} \xi_i
\]

**subject to**

\[
y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\
\text{for } i = 1, \ldots, N
\]
The slack variable allows for mistakes, as long as the inverse margin is minimized.
`soft' margin

Objective:
\[
\min_{w,\xi} \|w\|^2 + C \sum_i \xi_i
\]

Subject to:
\[
y_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, N
\]

- Every constraint can be satisfied if slack is large
- C is a regularization parameter
  - Small C: ignore constraints (larger margin)
  - Big C: constraints (small margin)
- Still QP problem (unique solution)
$C = \text{Infinity}$  hard margin

**Comment Window**

SVM (L1) by Sequential Minimal Optimizer
- Kernel: linear (-), C: Inf
- Kernel evaluations: 971
- Number of Support Vectors: 3
- Margin: 0.0966
- Training error: 0.00%
C = 10  soft margin

SVM (L1) by Sequential Minimal Optimizer
Kernel: linear (-), C: 10.0000
Kernel evaluations: 2645
Number of Support Vectors: 4
Margin: 0.2265
Training error: 3.70%
References

Basic reading:
• Szeliski, Chapter 14.