HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Computer Vision

Chapter 7: Object recognition (Part 1)

1

Chapter 7: Object recognition (Part 1) Content

- Overview of 'semantic vision'?
- Image classification/ recognition
- Bag-of-words
  - Recall
  - Vocabulary tree
- Classification
  - K nearest neighbours
  - Naïve Bayes
  - Support vector machine (SVM)

2

Is this a street light?
(Recognition / classification)

3

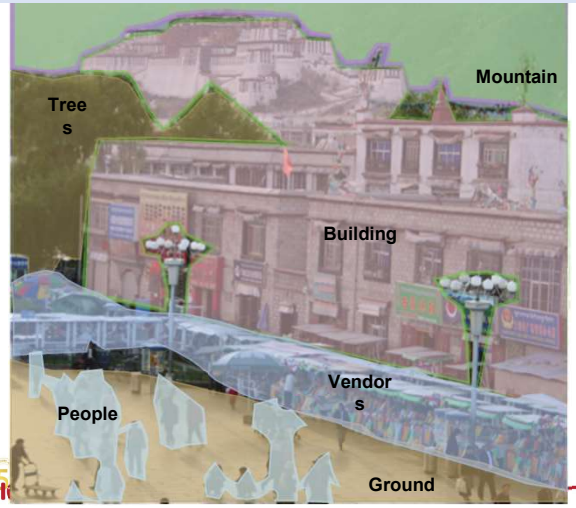
Where are the people?
(Detection)

4

Is that Potala palace?
(Identification)

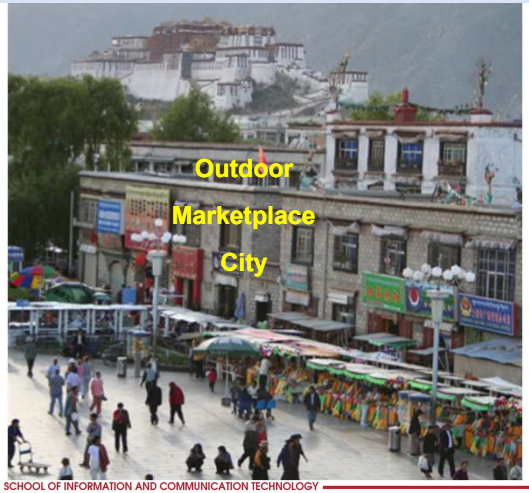
5

What's in the scene?
(semantic segmentation)



6

What type of scene is it?
(Scene categorization)



7

What are these people doing?
(Activity / Event Recognition)

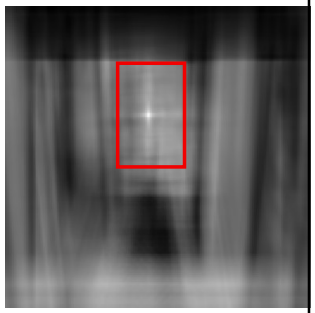
8

Object recognition
Is it really so hard?

This is a chair

Find the chair in this image

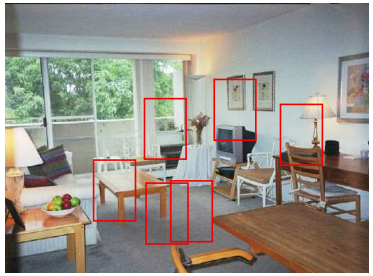
Output of normalized correlation

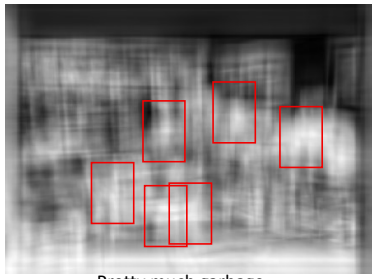


9

Object recognition
Is it really so hard?

Find the chair in this image





Pretty much garbage

Simple template matching is not going to make it

A "popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts." Nivatia & Binford, 1977.

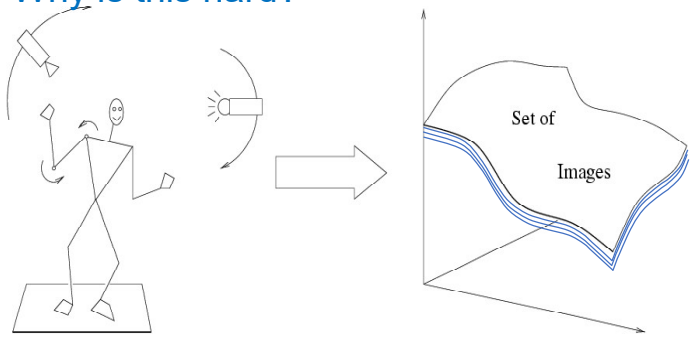
10

And it can get a lot harder

Brady, M. J., & Kersten, D. (2003). Bootstrapped learning of novel objects. J Vis, 3(6), 413-422

11

Why is this hard?

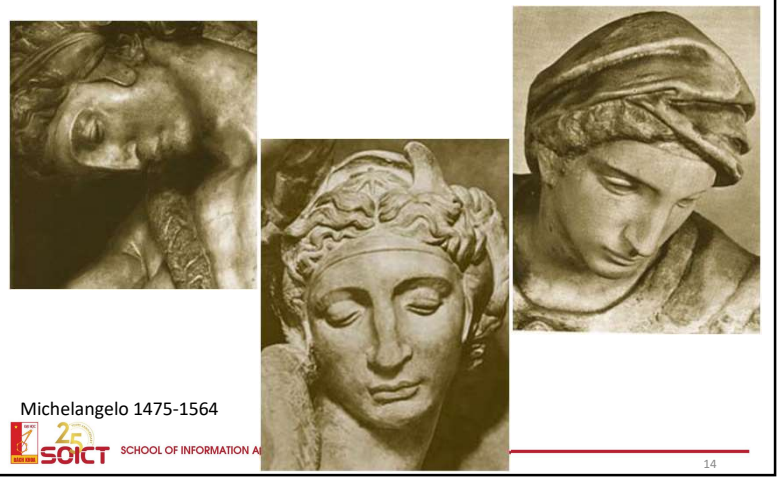


Variability: Camera position
Illumination
Shape parameters

12

13

Challenge: variable viewpoint



Michelangelo 1475-1564

14

Challenge: variable illumination

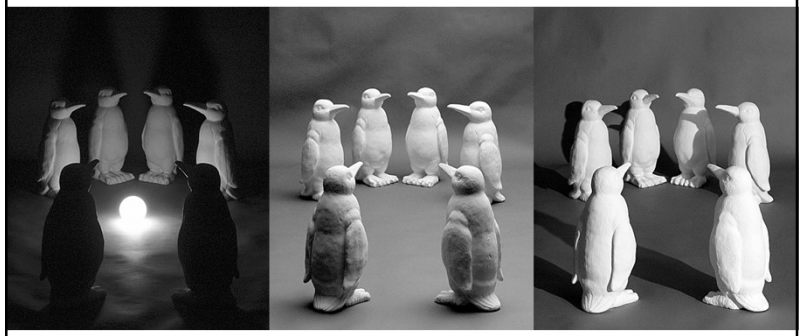


image credit: J. Koenderink

15

Challenge: scale

16

Challenge: deformation

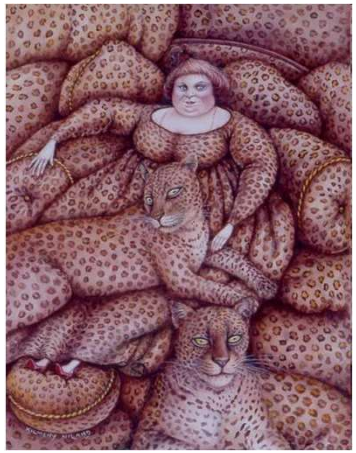
17

Challenge: Occlusion

Magritte, 1957

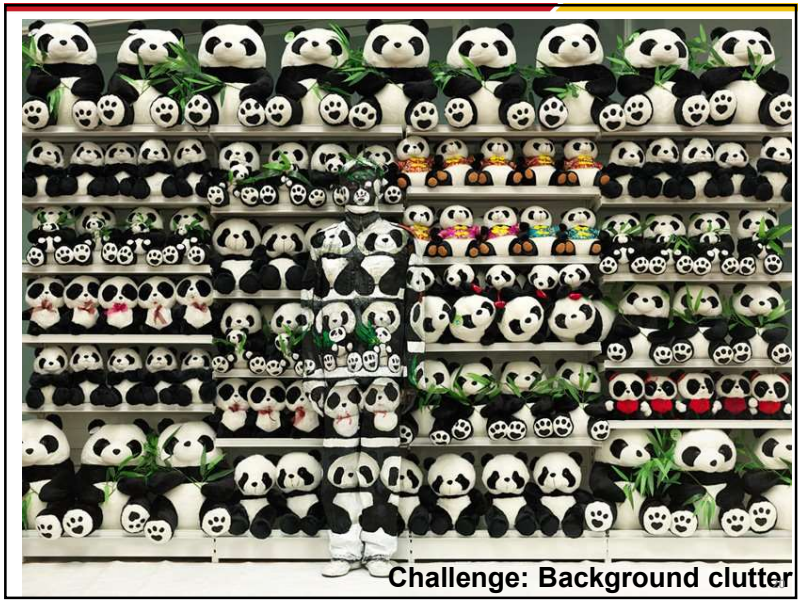
18

Challenge: background clutter



Kilmeny Niland. 1995

19



20

Challenge: intra-class variations

21

Image Classification/ Recognition

22

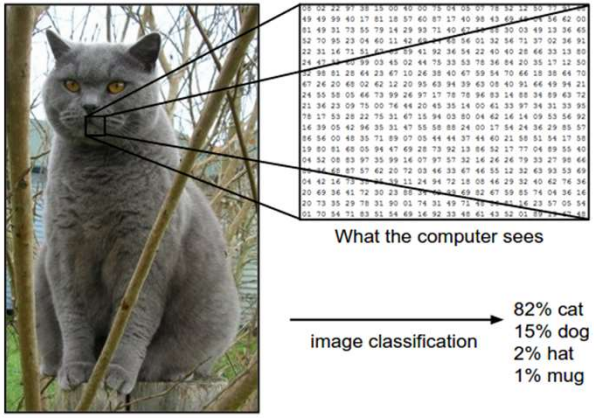
Image Classification/ Recognition

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

→ cat

23

Image Classification: Problem



24

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

25

A simple pipeline - Training

Training Images

Image Features

26

A simple pipeline - Training

Training Images

Image Features

Training

Training Labels

27

A simple pipeline - Training

Training Images

Image Features

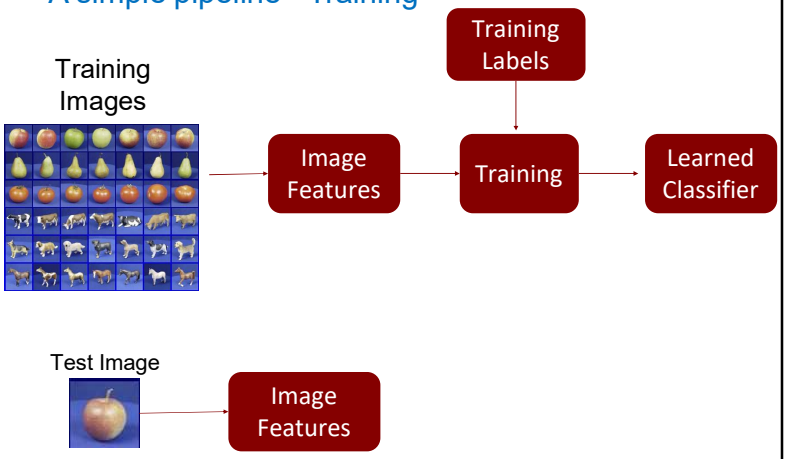
Training

Learned Classifier

Training Labels

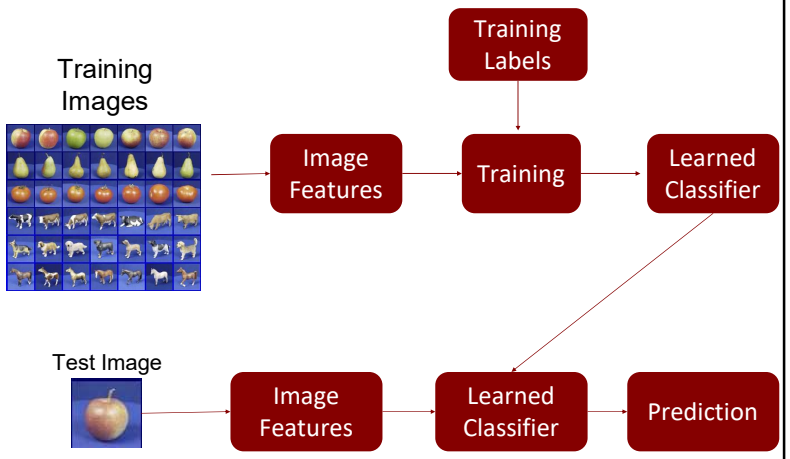
28

A simple pipeline - Training



29

A simple pipeline - Training



30

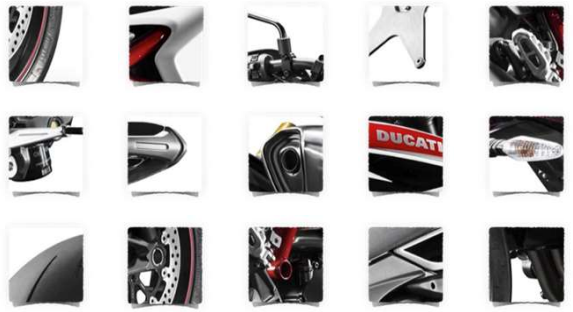
Bag of words

Basic model: recall

Vocabulary tree

31

What object do these parts belong to?



32

Some local feature are very informative

An object as

a collection of local features
(bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant

33

33

(not so) crazy assumption

spatial information of local features
can be ignored for object recognition (i.e., verification)

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY 34

34

CalTech6 dataset

class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Parts-and-shape model Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Works pretty well for image-level classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY 35

35

Bag-of-features

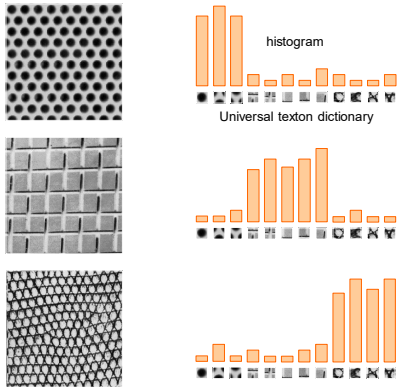
Represent a data item (document, texture, image)
as a histogram over features

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

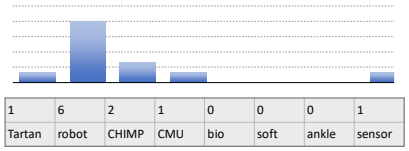
SOICT SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY 36

36

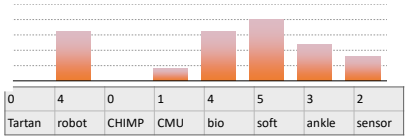
Texture recognition



Vector Space Model



1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor



0	4	0	1	4	5	3	2
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

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

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

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

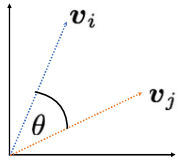
$n(\cdot)$ counts the number of occurrences

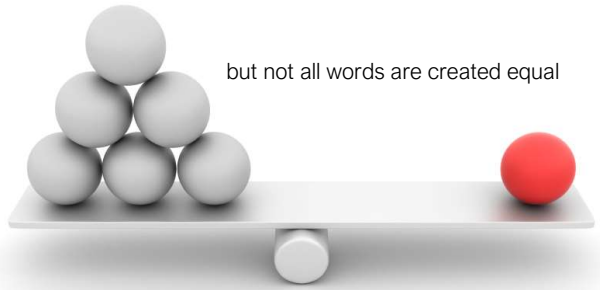
just a histogram over words

What is the similarity between two documents?

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

$$d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \cos\theta = \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{\|\boldsymbol{v}_i\|\|\boldsymbol{v}_j\|}$$





TF-IDF

Term **F**requency **I**nverse **D**ocument **F**requency

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

weigh each word by a heuristic

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

term frequency

inverse document frequency

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

(down-weights **common** terms)

Standard BOW pipeline

(for image classification)

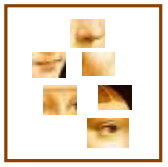
Dictionary Learning:
Learn Visual Words using clustering

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

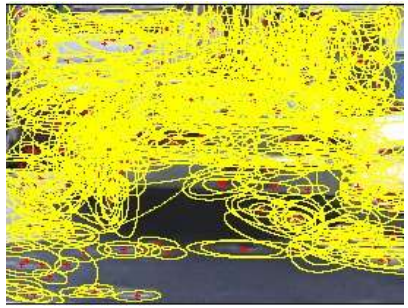


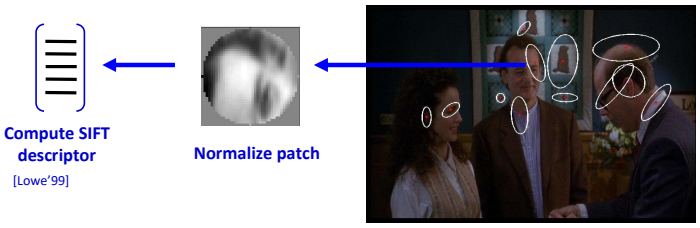
Dictionary Learning:
Learn Visual Words using clustering

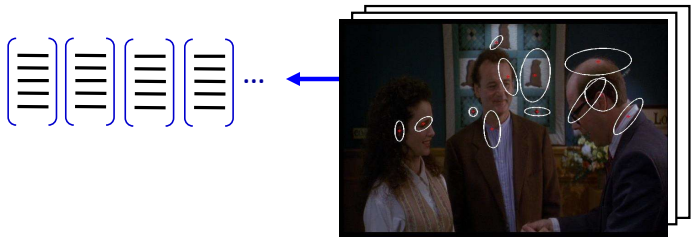
2. Learn visual dictionary (e.g., K-means clustering)

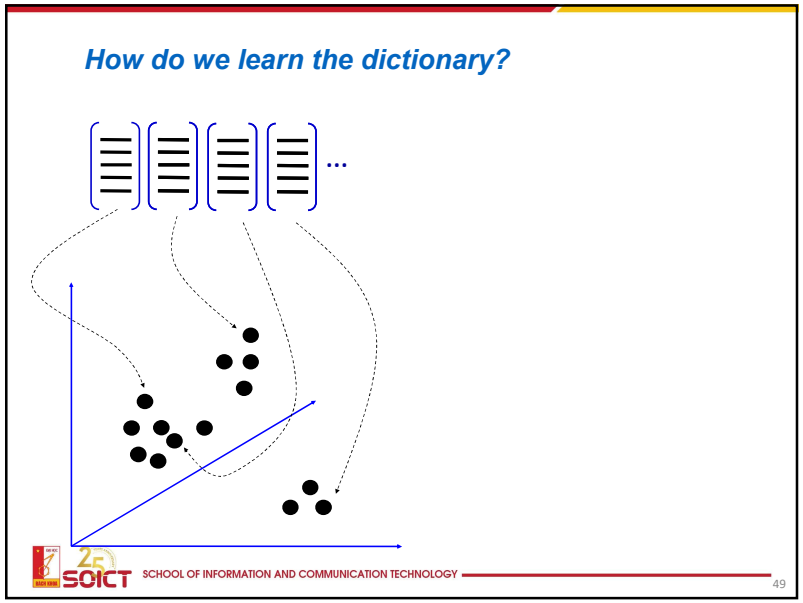
What kinds of features can we extract?

- 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)

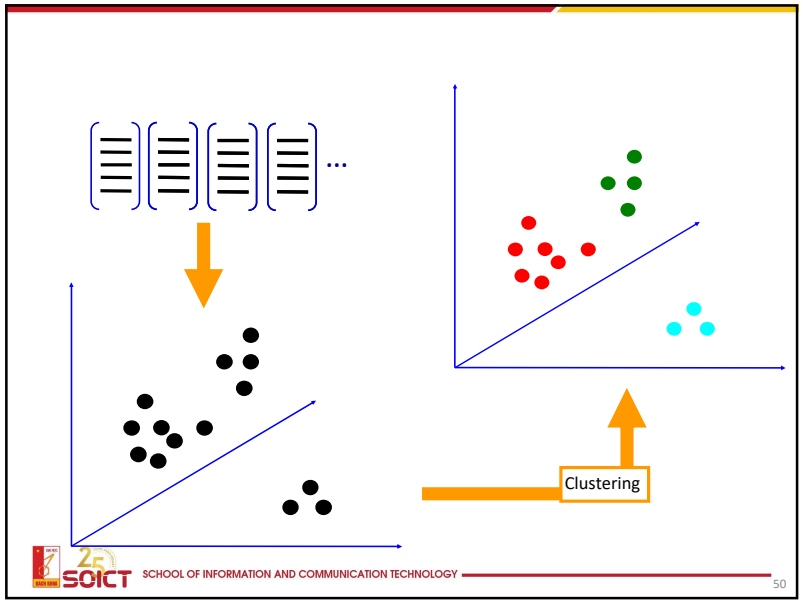




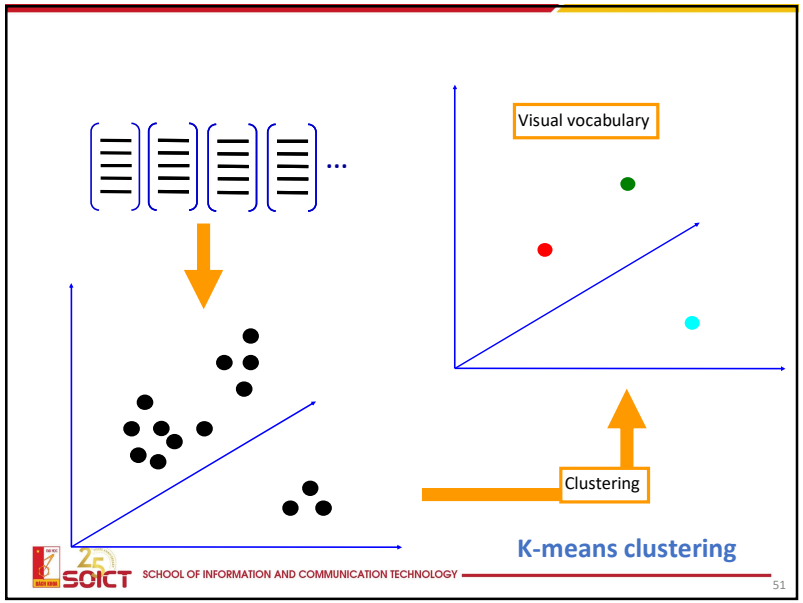



How do we learn the dictionary?
SOICT
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
49

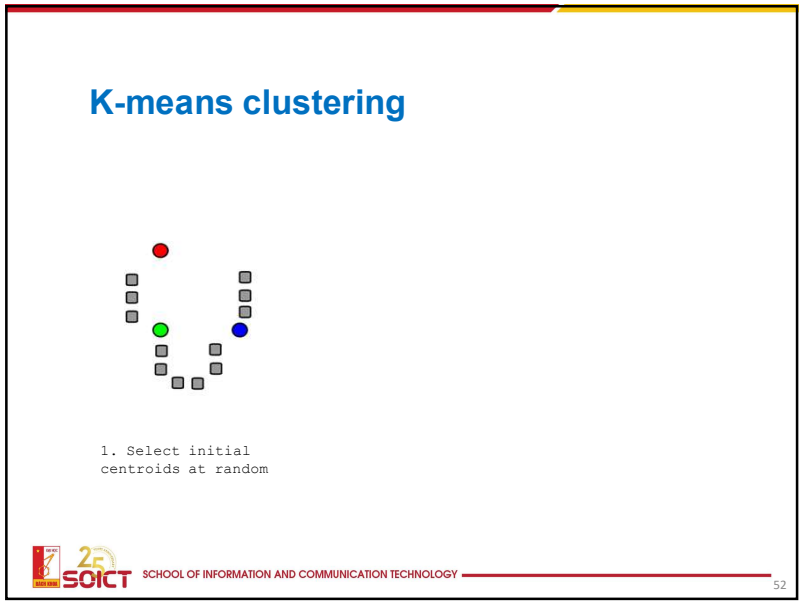
49


Clustering
SOICT
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
50

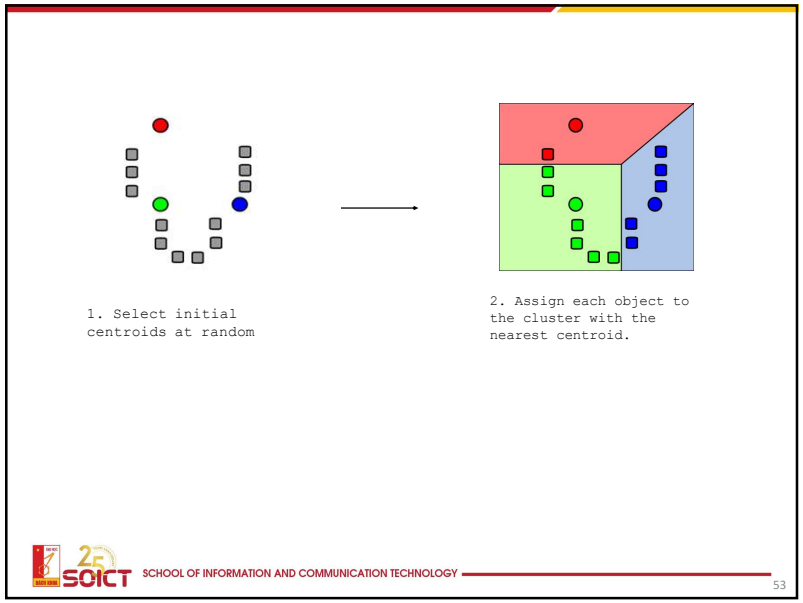
50


Visual vocabulary
Clustering
K-means clustering
SOICT
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
51

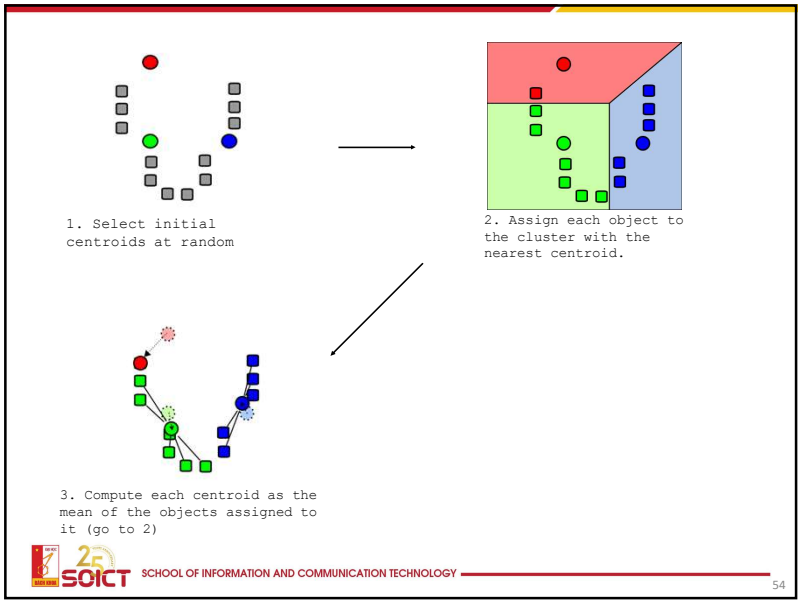
51


K-means clustering
1. Select initial
centroids at random
SOICT
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
52

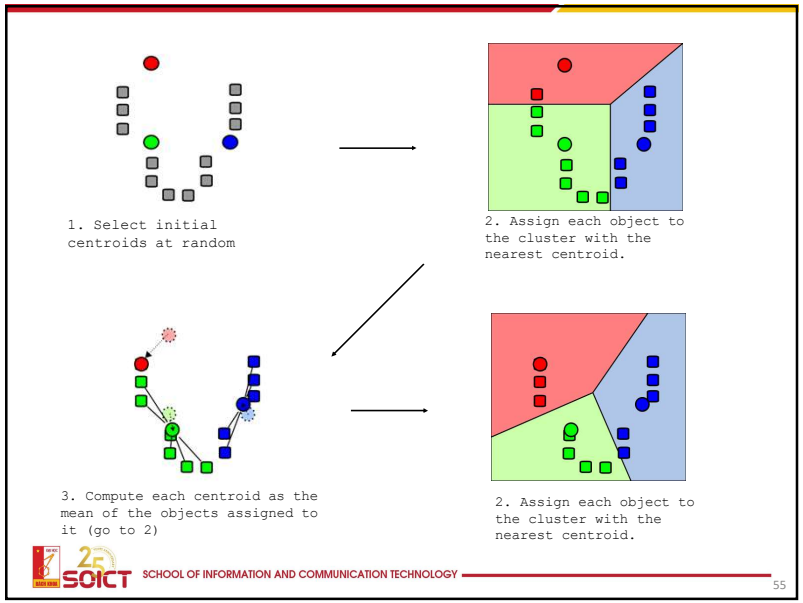
52


1. Select initial centroids at random
2. Assign each object to the cluster with the nearest centroid.
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
53

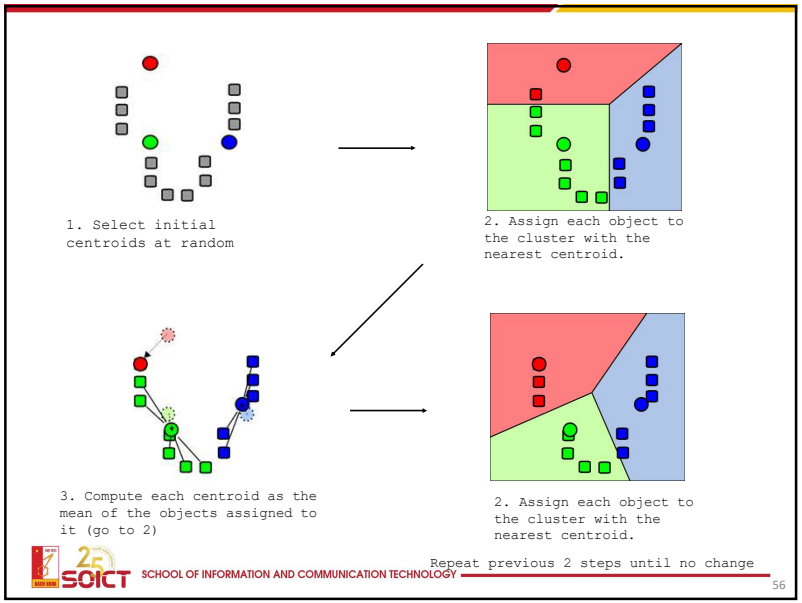
53


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)
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
54

54


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.
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
55

55


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.
Repeat previous 2 steps until no change
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
56

56

K-means Clustering Algorithm

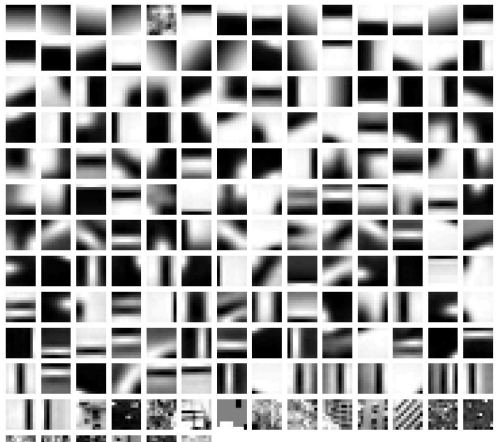
Given k:

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.

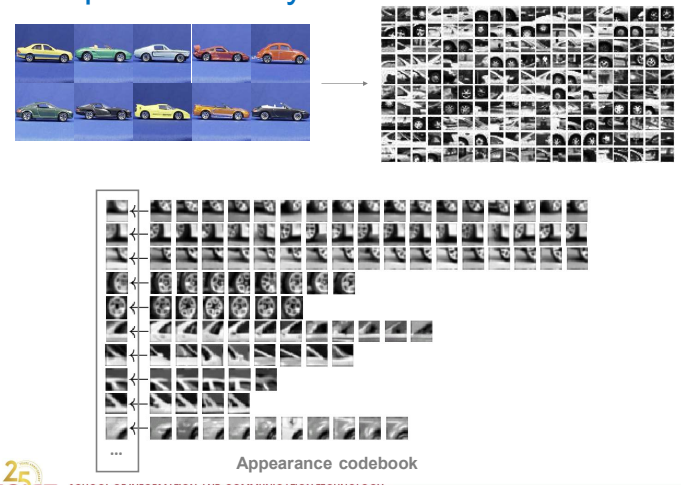
*From what **data** should I learn the dictionary?*

- 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



Source: B. Leibe

Another dictionary

Source: B. Leibe

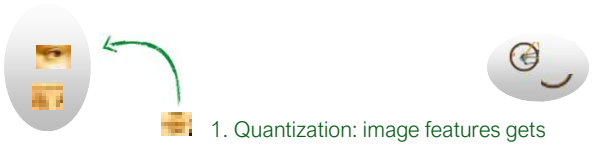
61

Dictionary Learning:
Learn Visual Words using clustering

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

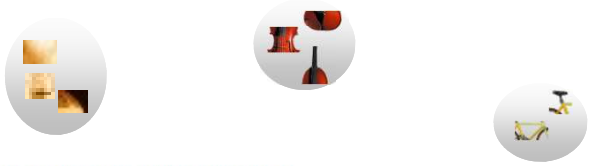
Classify:
Train and test data using BOWs

62



1. Quantization: image features gets associated to a visual word (nearest cluster center)

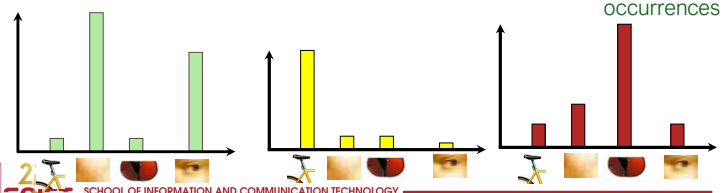
Encode:
build Bags-of-Words (BOW) vectors
for each image



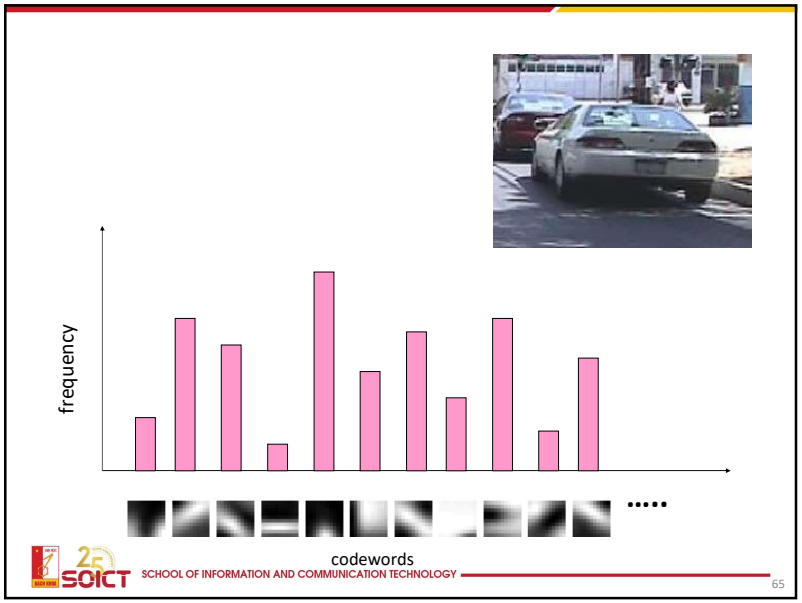
63

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

2. Histogram: count the number of visual word occurrences



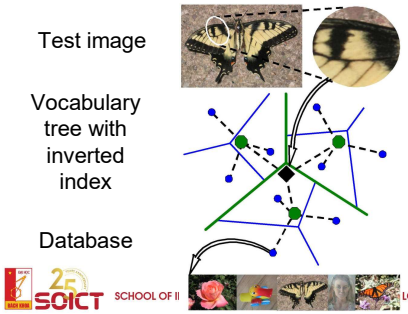
64



65

Scalability: Alignment to large databases

- What if we need to align a test image with thousands or millions of images in a model database?
  - Efficient putative match generation
    - Fast nearest neighbor search, inverted indexes

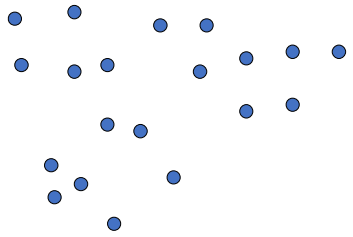


D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

slide: S. Lazebnik

66

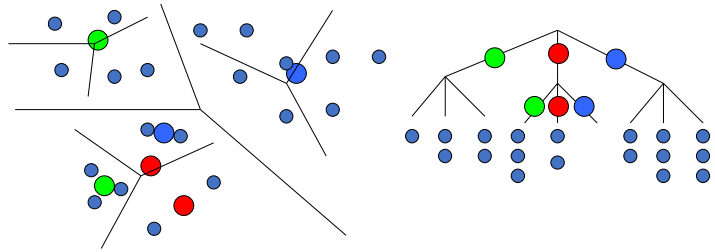
What is a Vocabulary Tree?



Nister and Stewenius CVPR 2006

67

What is a Vocabulary Tree?

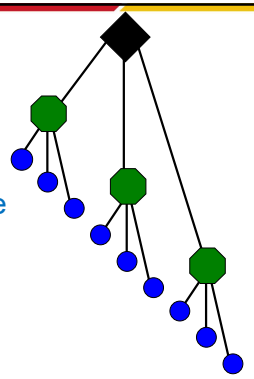


- Multiple rounds of K-Means to compute decision tree (offline)
- Fill and query tree online

Nister and Stewenius CVPR 2006

68

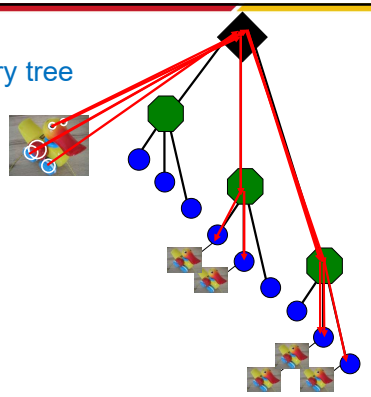
Populating the vocabulary tree /inverted index



69

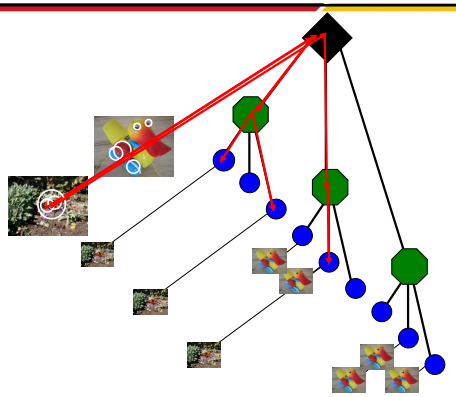
Populating the vocabulary tree /inverted index

Model images



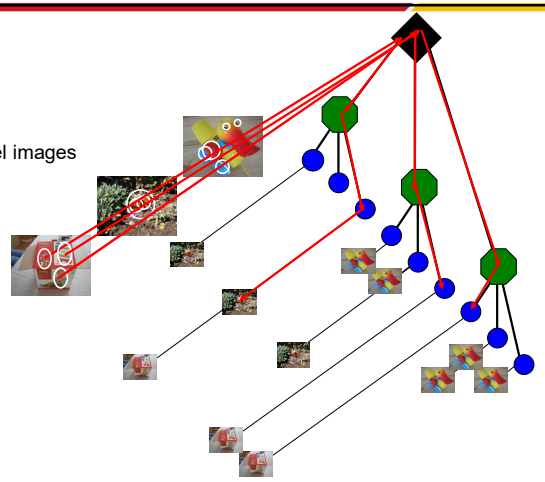
70

Model images

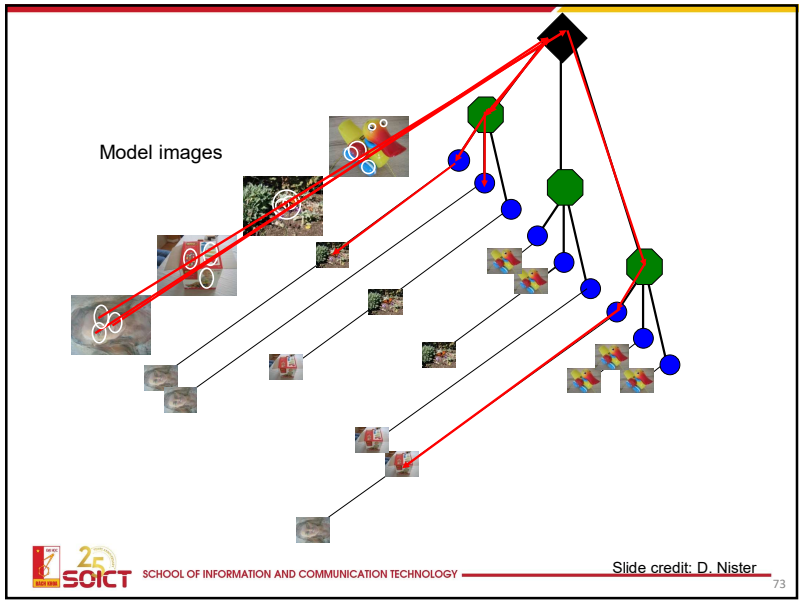


71

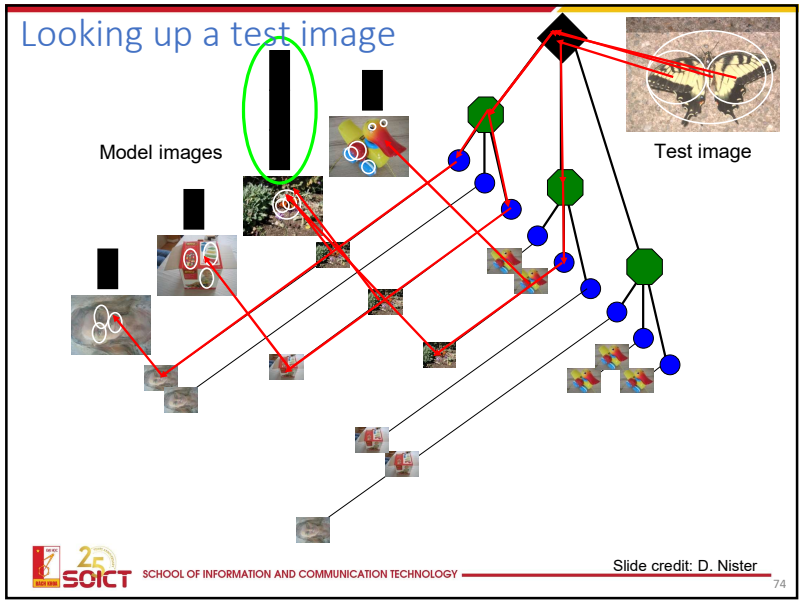
Model images



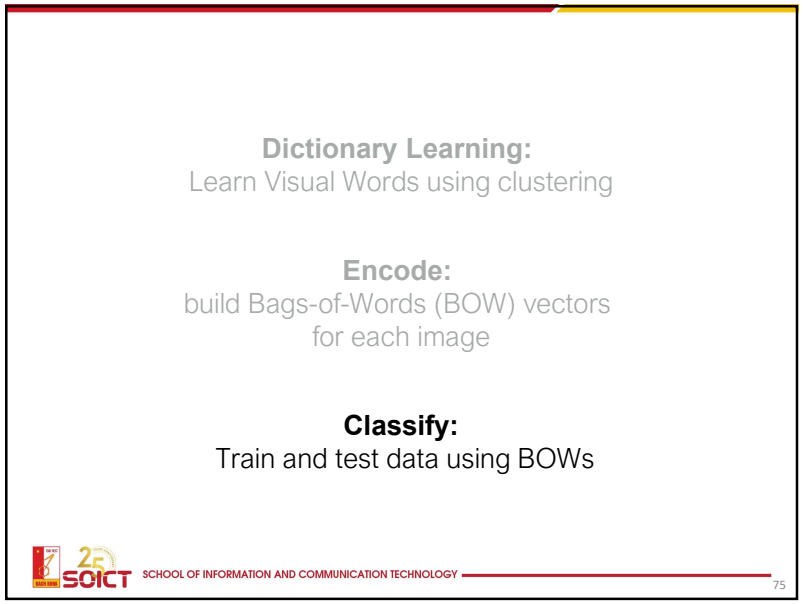
72


Model images
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
Slide credit: D. Nister
73

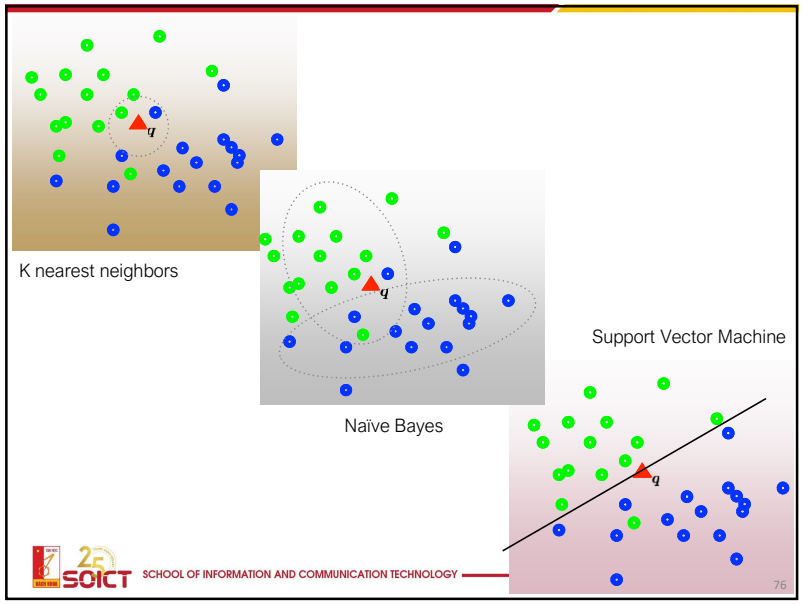
73


Looking up a test image
Model images
Test image
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
Slide credit: D. Nister
74

74


Dictionary Learning:
Learn Visual Words using clustering
Encode:
build Bags-of-Words (BOW) vectors
for each image
Classify:
Train and test data using BOWs
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
75

75

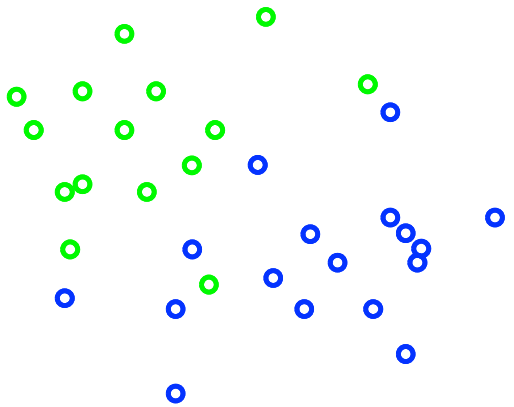

K nearest neighbors
Naïve Bayes
Support Vector Machine
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
76

76

Classification
K nearest neighbors

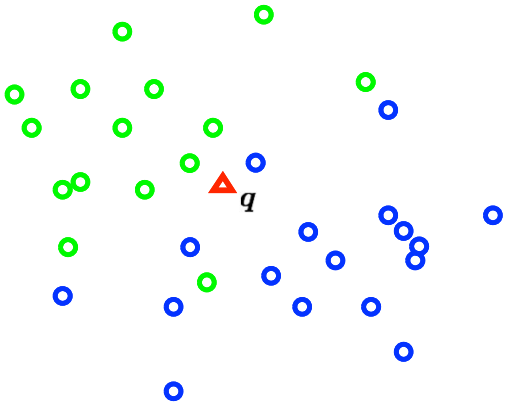
77

Distribution of data from two classes



78

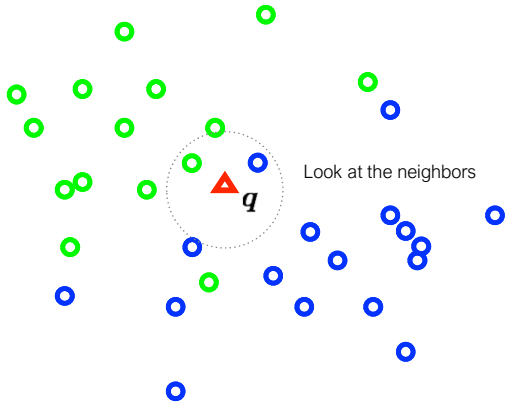
Distribution of data from two classes



Which class does q belong too?

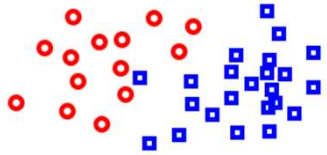
79

Distribution of data from two classes



80

K-Nearest Neighbor (KNN) Classifier

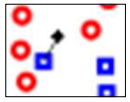


Non-parametric pattern classification approach

Consider a two class problem where each sample consists of two measurements (x,y).

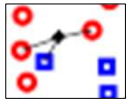
For a given query point q, assign the class of **the nearest neighbor**

k = 1



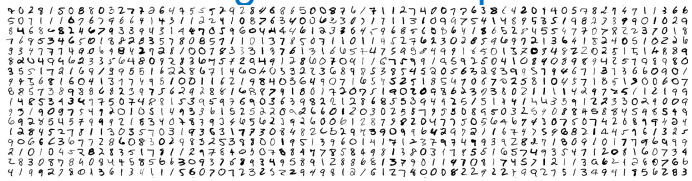
Compute the **k nearest neighbors** and assign the class by majority vote.

k = 3



81

Nearest Neighbor is competitive



MNIST Digit Recognition

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

Yann LeCunn

	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

82

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!)

83

Distance metrics

$$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(x_1-y_1)^2+\cdots+(x_N-y_N)^2}$$ Euclidean

$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = \frac{x_1y_1+\cdots+x_Ny_N}{\sqrt{\sum_n x_n^2}\sqrt{\sum_n y_n^2}}$$ Cosine

$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2}\sum_n \frac{(x_n-y_n)^2}{(x_n+y_n)}$$ Chi-squared

84

Distance metrics

L1 (Manhattan) distance

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

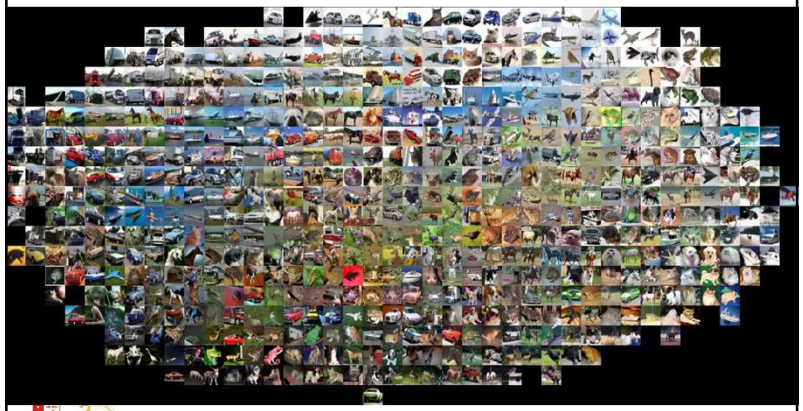
L2 (Euclidean) distance

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

- Two most commonly used special cases of p-norm

$$||x||_p = \left(|x_1|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}} \qquad p \geq 1, x \in \mathbb{R}^n$$

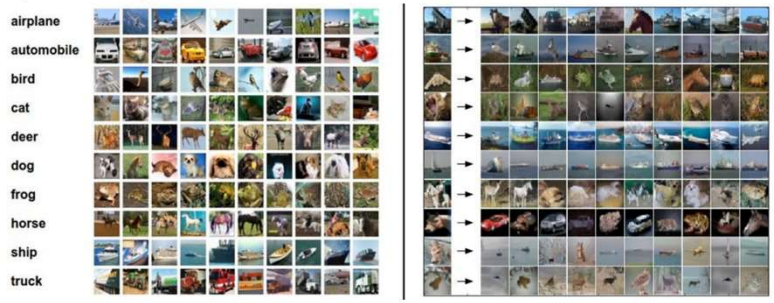
Visualization: L2 distance



CIFAR-10 and NN results

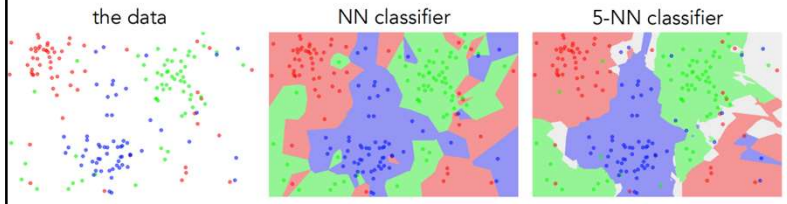
Example dataset: **CIFAR-10**
10 labels
50,000 training images
10,000 test images.

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

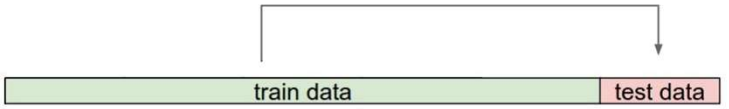
- What is the best distance to use?
- What is the best value of k to use?

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

- Very problem-dependent
- Must try them all and see what works best

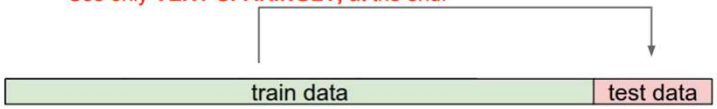
89

Try out what hyperparameters work best on test set.



90

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.



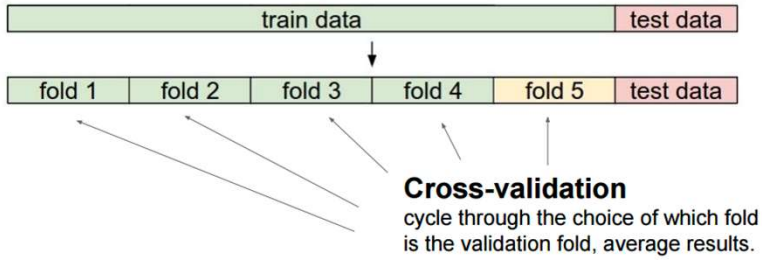
91

Validation

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

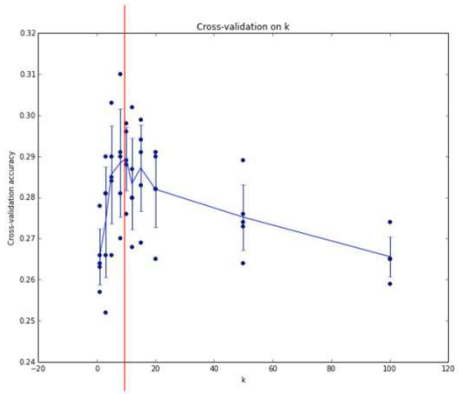
92

Cross-validation



Cross-validation
cycle through the choice of which fold is the validation fold, average results.

93



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 ~= 7 works best for this data)

94

How to pick hyper-parameters?

- Methodology
  - Train and test
  - Train, validate, test
- Train for original model
- Validate to find hyperparameters
- Test to understand generalizability

95

kNN

Pros

- simple yet effective

Cons

- search is expensive (can be sped-up)
- storage requirements
- difficulties with high-dimensional data

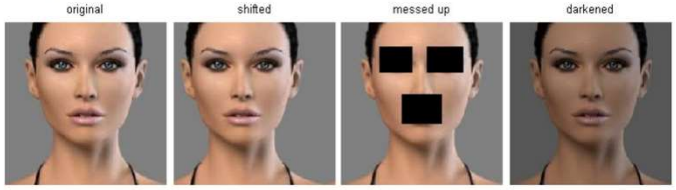
96

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.**

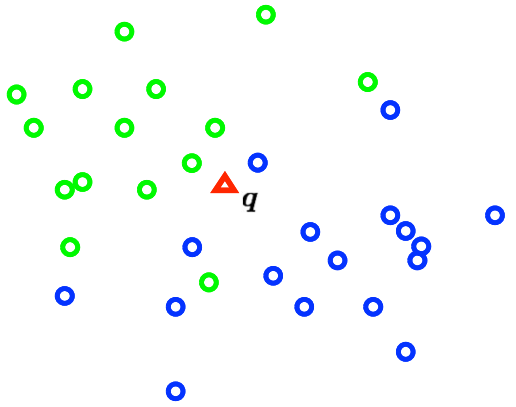
- terrible performance at test time
- distance metrics on level of whole images can be very unintuitive



(all 3 images have same L2 distance to the one on the left)

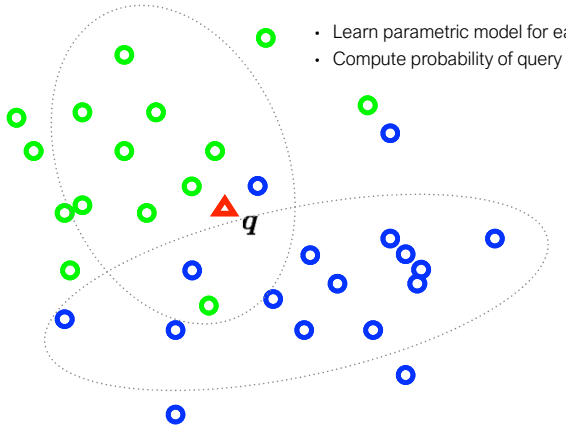
Classification
Naïve Bayes

Distribution of data from two classes



Which class does q belong too?

Distribution of data from two classes



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

101

This is called the posterior:

the probability of a class z given the observed features X

$$p(z|\boldsymbol{X})$$

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)

102

This is called the posterior:

the probability of a class z given the observed features X

$$p(z|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$$

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)

103

Recall:

The posterior can be decomposed according to **Bayes' Rule**

$$\underset{\text{posterior}}{p(A|B)} = \frac{\overset{\text{likelihood}}{p(B|A)}\overset{\text{prior}}{p(A)}}{p(B)}$$

In our context…

$$p(z|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \frac{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N|z)p(z)}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)}$$

104

The naive Bayes' classifier is solving this optimization

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

MAP (maximum a posteriori) estimate

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

Bayes' Rule

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

Remove constants

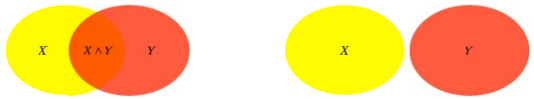
To optimize this…we need to compute this

Compute the likelihood…

A naive Bayes' classifier assumes all features are ***conditionally independent***

$$\begin{aligned} p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) &= p(\boldsymbol{x}_1|\boldsymbol{z})p(\boldsymbol{x}_2, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\ &= p(\boldsymbol{x}_1|\boldsymbol{z})p(\boldsymbol{x}_2|\boldsymbol{z})p(\boldsymbol{x}_3, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\ &= p(\boldsymbol{x}_1|\boldsymbol{z})p(\boldsymbol{x}_2|\boldsymbol{z}) \cdots p(\boldsymbol{x}_N|\boldsymbol{z}) \end{aligned}$$

Recall:



$$p(x, y) = p(x|y)p(y) \qquad p(x, y) = p(x)p(y)$$

To compute the MAP estimate

Given (1) a set of known parameters

$$p(\boldsymbol{z}) \quad p(\boldsymbol{x}|\boldsymbol{z})$$

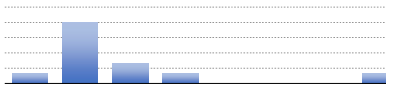
(2) observations

$$\{x_1, x_2, \ldots, x_N\}$$

Compute which z has the largest probability

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z) \prod_n p(x_n|z)$$


The Newsp
DARPA Selects Carnegie Me



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

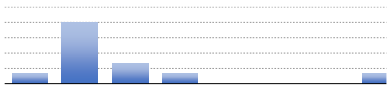
$$\begin{aligned} p(X|z) &= \prod_v p(x_v|z)^{c(w_v)} \\ &= (0.09)^1(0.55)^6 \cdots (0.09)^1 \end{aligned}$$

Numbers get really small so use log probabilities

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

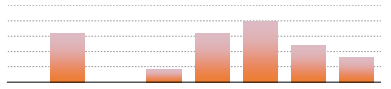
$$\log p(X|z = \text{'softrobot'}) = -7.63 - 9.37 - 15.18 - 2.97 - 0.02 - 0.01 - 0.02 - 2.27 = -37.48$$

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



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

log p(X|z=grand challenge) = **- 14.58**
log p(X|z=bio inspired) = - 37.48



count	0	4	0	1	4	5	3	2
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.0	0.21	0.0	0.05	0.21	0.26	0.16	0.11

log p(X|z=grand challenge) = - 94.06
log p(X|z=bio inspired) = **- 32.41**

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

Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*
- Calculate *P*(*z*) terms
  - For each *z* in *C* do
    $docs_z \leftarrow$ all docs with class =*z*

$$p(z) = \frac{|docs_z|}{total\ number\ of\ documents}$$

- Calculate $P(x_i \mid z)$ terms
  - $Text_z \leftarrow$ single doc containing all $docs_z$
  - For each word x_i in *Vocabulary*
    $n_i \leftarrow$ # of occurrences of x_i in $Text_z$

$$p(x_i|z) = (n_i + 1)/(\sum_{x_i \in vocabulary}(n_i + 1))$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:
P(z =*c*) = 3/4
P(z=*j*)= 1/4

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Conditional Probabilities:
P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7
P(Tokyo|*c*) = (0+1) / (8+6) = 1/14
P(Japan|*c*) = (0+1) / (8+6) = 1/14
P(Chinese|*j*) = (1+1) / (3+6) = 2/9
P(Tokyo|*j*) = (1+1) / (3+6) = 2/9
P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

P (z=c|d5) $\propto$ 3/4 * $(3/7)^3$ * 1/14 * 1/14
≈ 0.0003

P (z = j|d5) $\propto$ 1/4 * $(2/9)^3$ * 2/9 * 2/9
≈ 0.0001

Support Vector Machine (SVM)

Image Classification

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

→ cat

Score function

→ class scores

Linear Classifier

define a **score function**

data (histogram)

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

class scores

"weights"

"bias vector"

"parameters"

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

Convert image to histogram representation

input image

0.2	-0.5	0.1	2.0
1.5	1.3	2.1	0.0
0	0.25	0.2	-0.3

W

56
231
24
2

x_i

+

1.1
3.2
-1.2

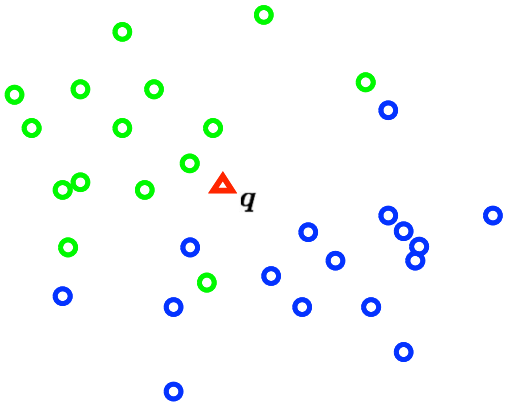
b

→

-96.8	cat score
437.9	dog score
61.95	ship score

$f(x_i; W, b)$

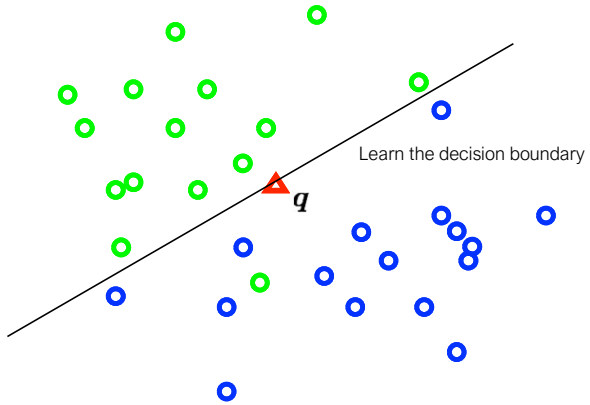
Distribution of data from two classes



Which class does q belong too?

117

Distribution of data from two classes



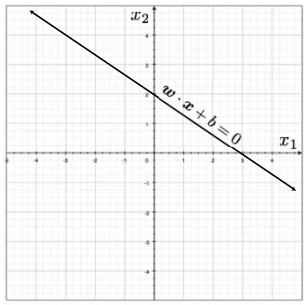
118

First we need to understand hyperplanes…

119

Hyperplanes (lines) in 2D

$$w_1 x_1 + w_2 x_2 + b = 0$$



a line can be written as dot product plus a bias

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

$$\boldsymbol{w} \in \mathcal{R}^2$$

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

$$\boldsymbol{w} \cdot \boldsymbol{x} = 0$$

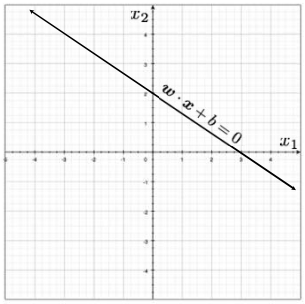
$$\boldsymbol{w} \in \mathcal{R}^3$$

120

Hyperplanes (lines) in 2D

$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ (offset/bias outside) $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ (offset/bias inside)

$w_1 x_1 + w_2 x_2 + b = 0$

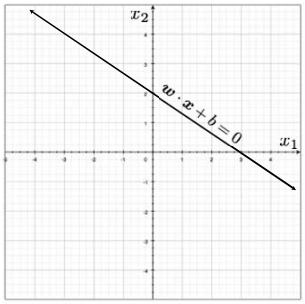


121

Hyperplanes (lines) in 2D

$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ (offset/bias outside) $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ (offset/bias inside)

$w_1 x_1 + w_2 x_2 + b = 0$



Important property:
Free to choose any normalization of w

The line

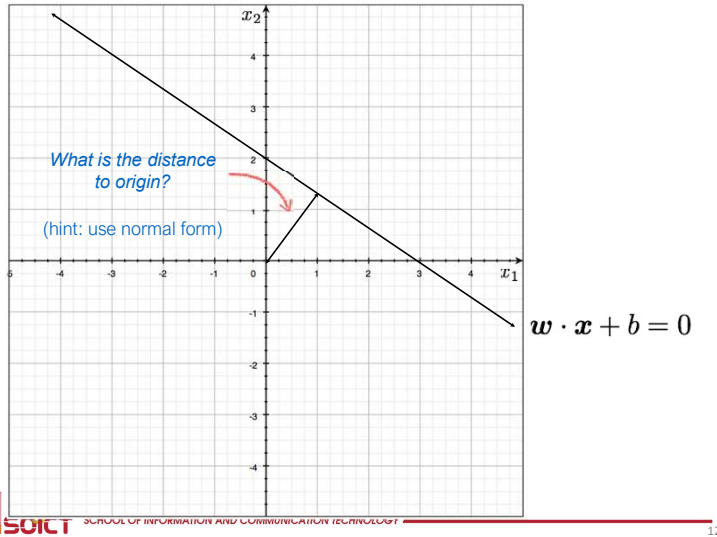
$w_1 x_1 + w_2 x_2 + b = 0$

and the line

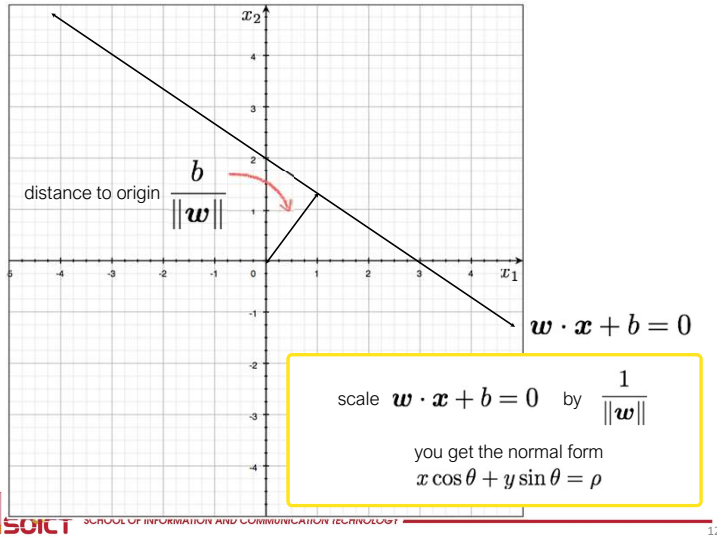
$\lambda(w_1 x_1 + w_2 x_2 + b) = 0$

define the same line

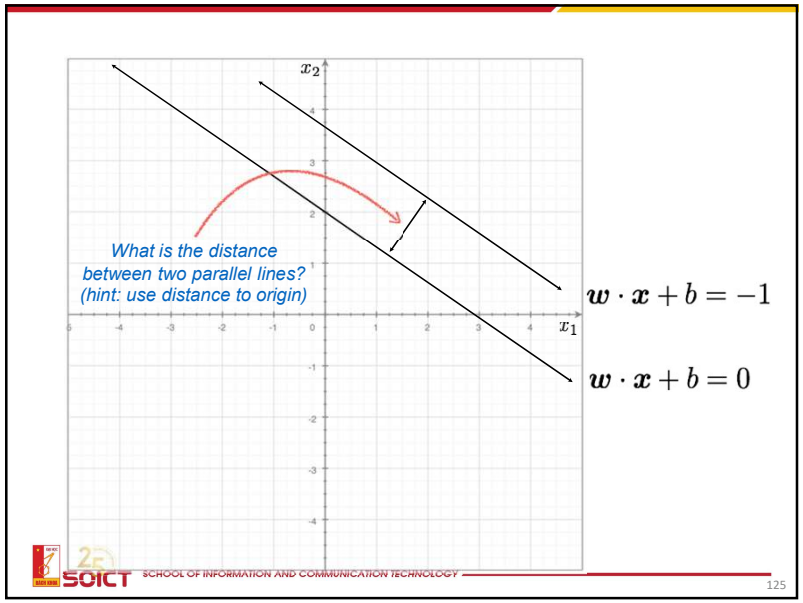
122



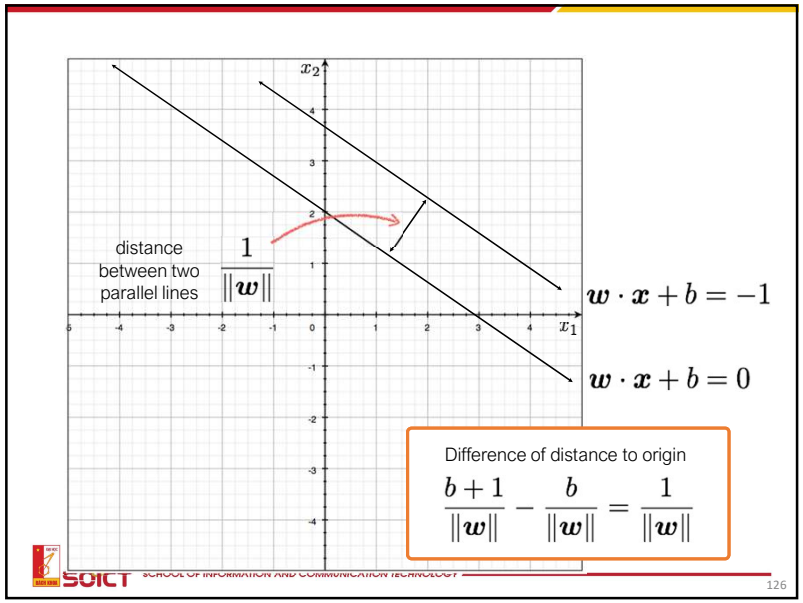
123



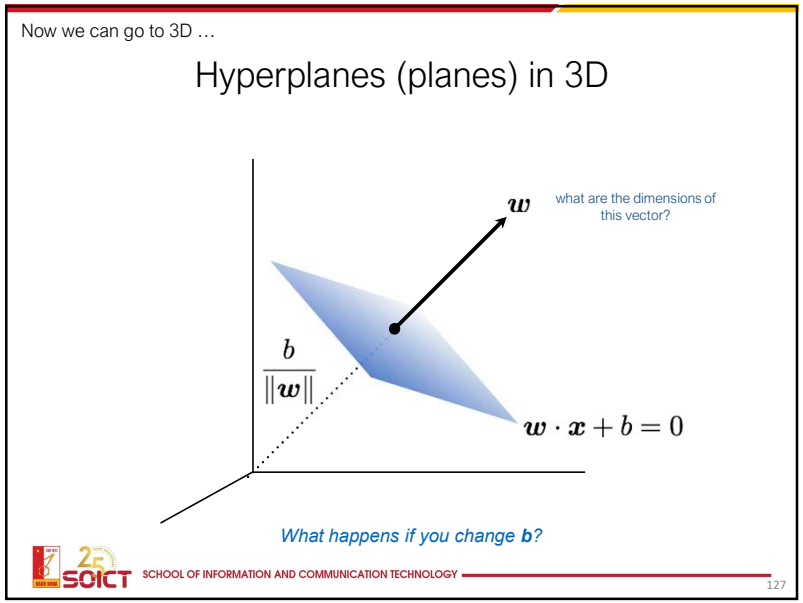
124



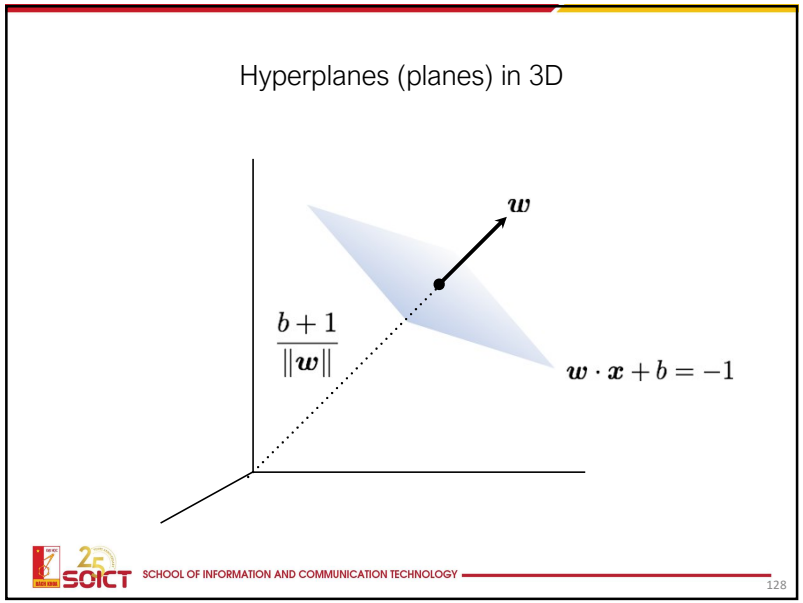
125



126

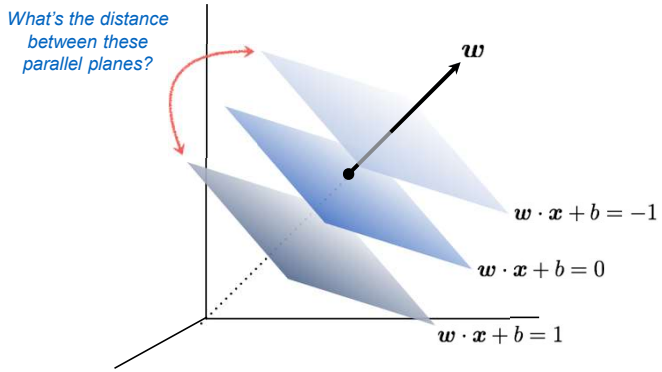


127



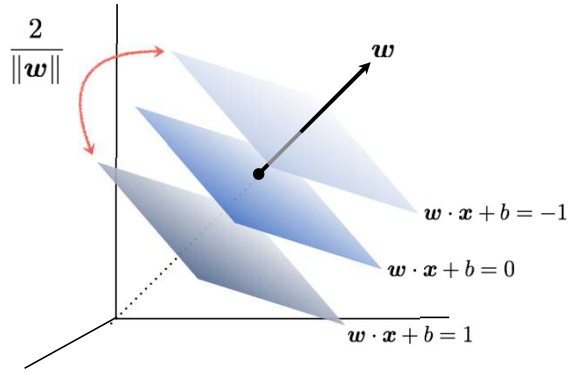
128

Hyperplanes (planes) in 3D



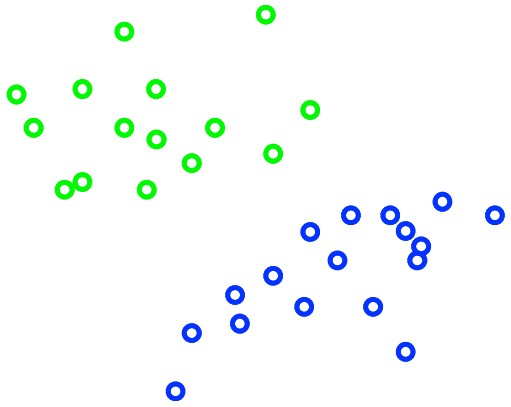
129

Hyperplanes (planes) in 3D



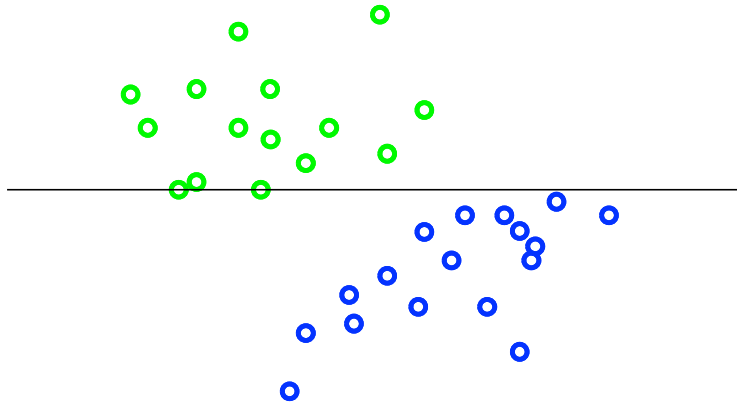
130

What's the best **w**?

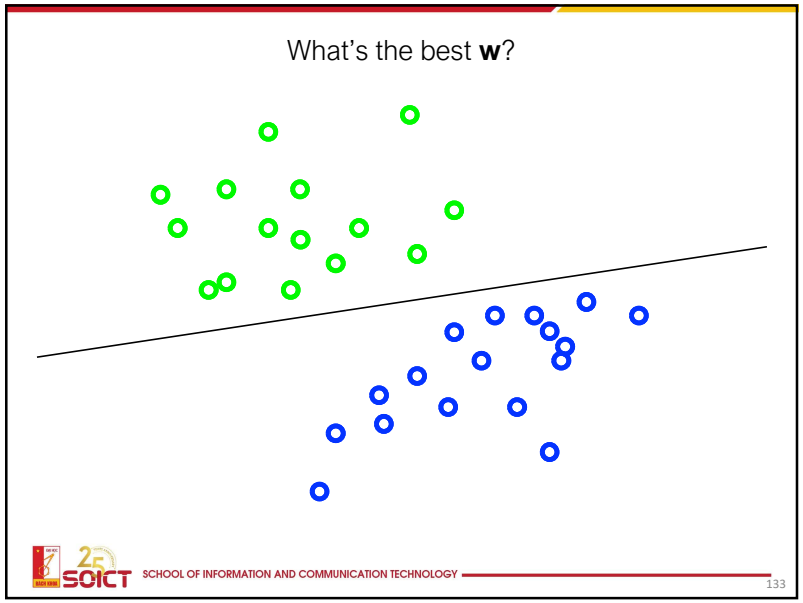


131

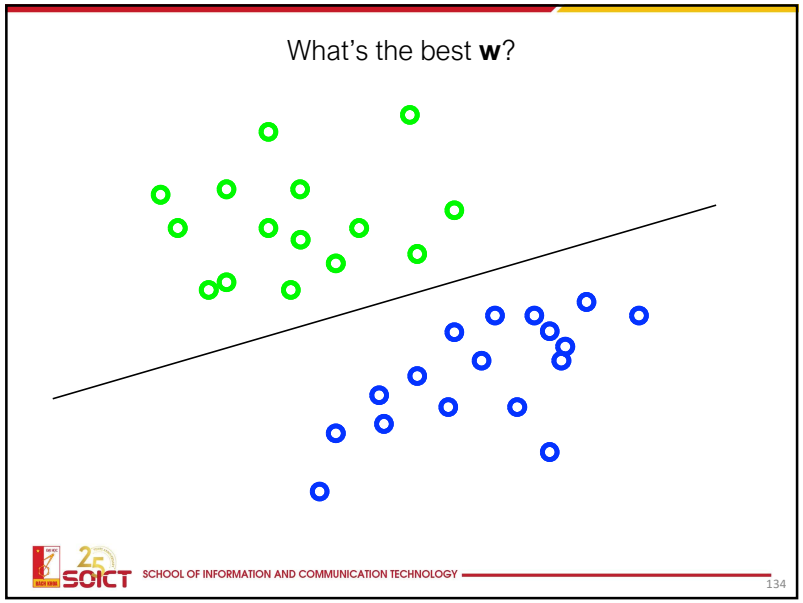
What's the best **w**?



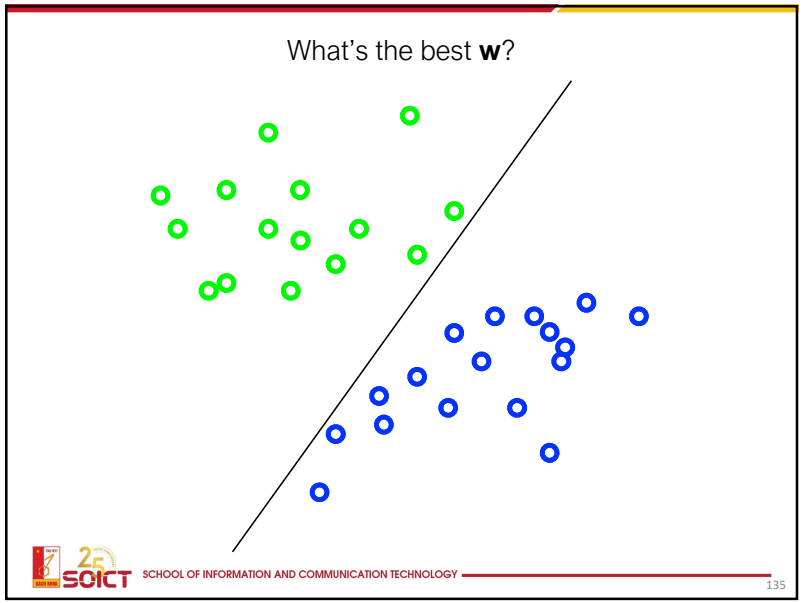
132



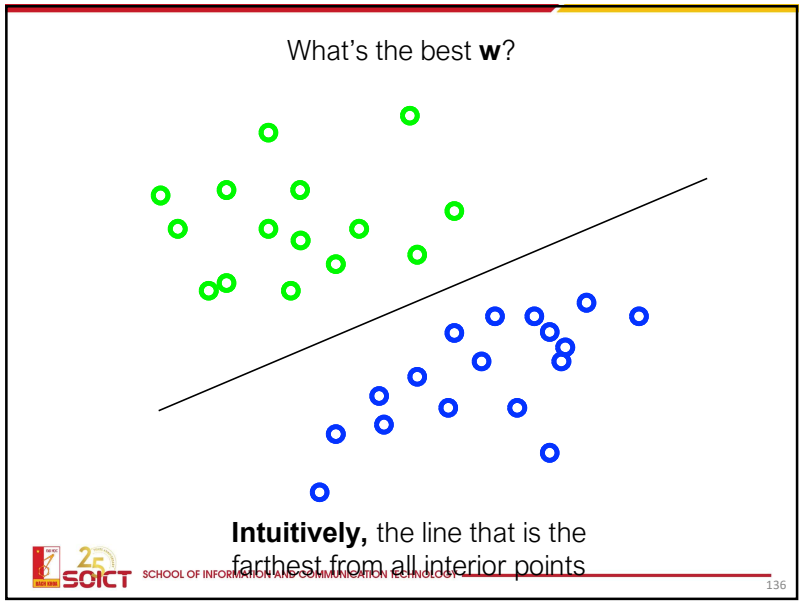
133



134

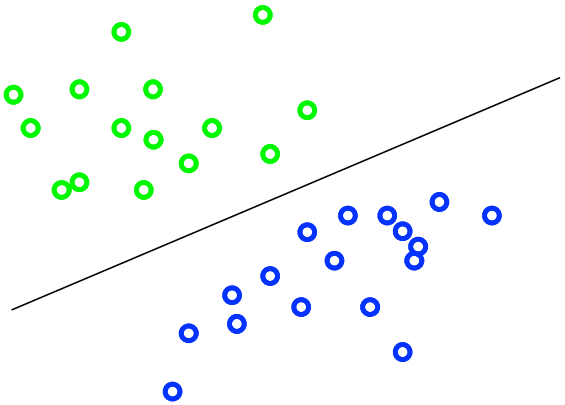


135



136

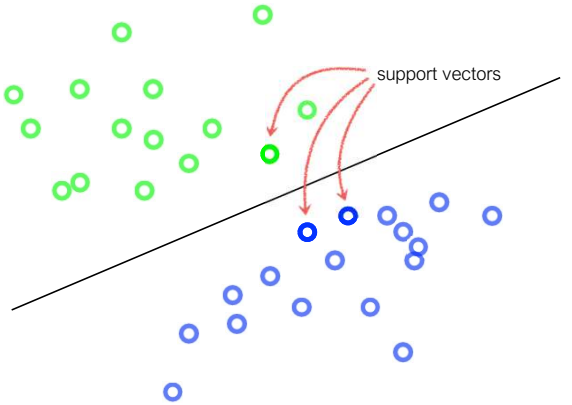
What's the best **w**?



Maximum Margin solution:
most stable to perturbations of data

137

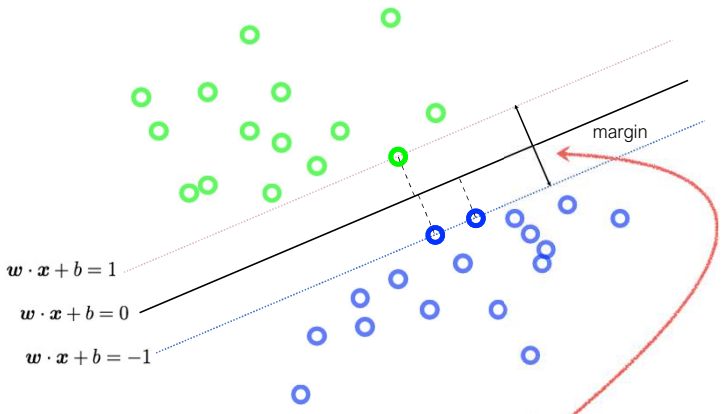
What's the best **w**?



Want a hyperplane that is far away from 'inner points'

138

Find hyperplane **w** such that …



the gap between parallel hyperplanes $\frac{2}{\|\boldsymbol{w}\|}$ is maximized

139

Can be formulated as a maximization problem

$$\max_{\boldsymbol{w}} \frac{2}{\|\boldsymbol{w}\|}$$

$$\text{subject to } \boldsymbol{w} \cdot \boldsymbol{x}_i + b \begin{matrix} \geq +1 \text{ if } y_i = +1 \\ \leq -1 \text{ if } y_i = -1 \end{matrix} \text{ for } i = 1, \ldots, N$$

What does this constraint mean?

label of the data point

Why is it +1 and -1?

140

Can be formulated as a maximization problem

$$\max_{\boldsymbol{w}} \frac{2}{\|\boldsymbol{w}\|}$$

$$\text{subject to } \boldsymbol{w}\cdot\boldsymbol{x}_i + b \begin{matrix} \geq +1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{matrix} \text{ for } i = 1,\ldots,N$$

Equivalently,

Where did the 2 go?

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|$$

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

What happened to the labels?

141

'Primal formulation' of a linear SVM

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|$$

Objective Function

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

Constraints

This is a convex quadratic programming (QP) problem

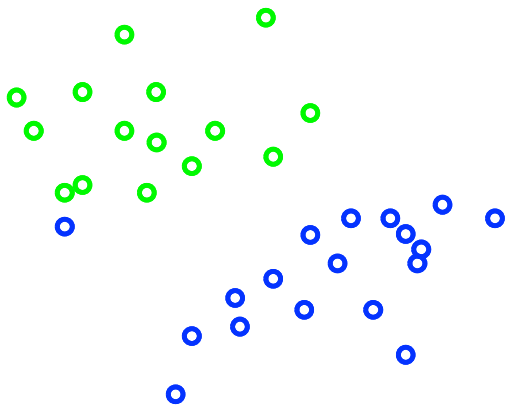
(a unique solution exists)

142

SVM: 'soft' margin

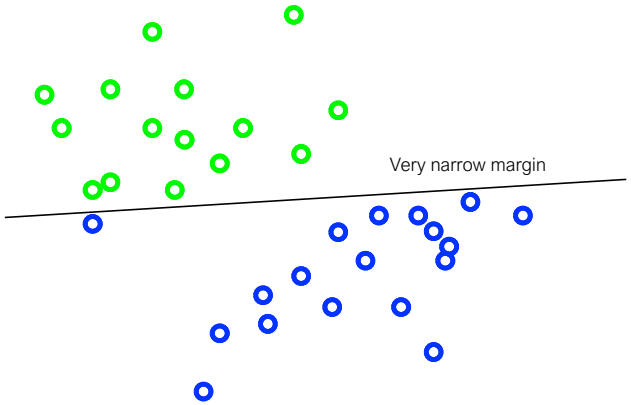
143

What's the best **w**?



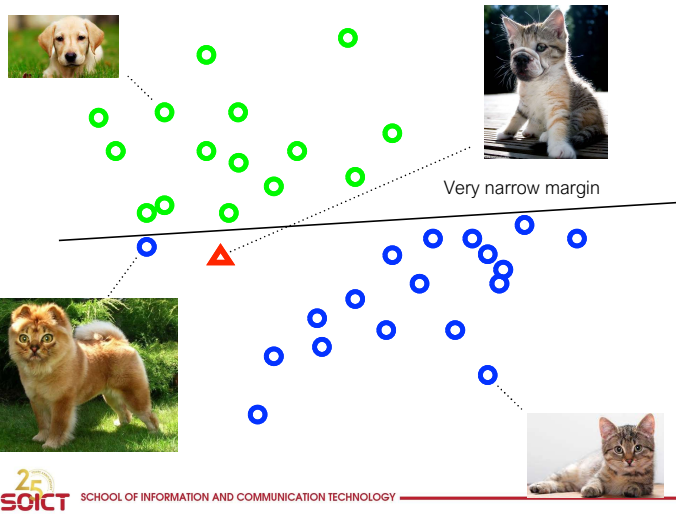
144

What's the best **w**?



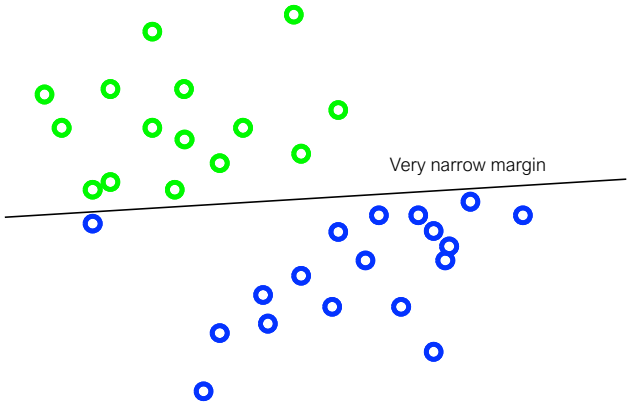
145

Separating cats and dogs



146

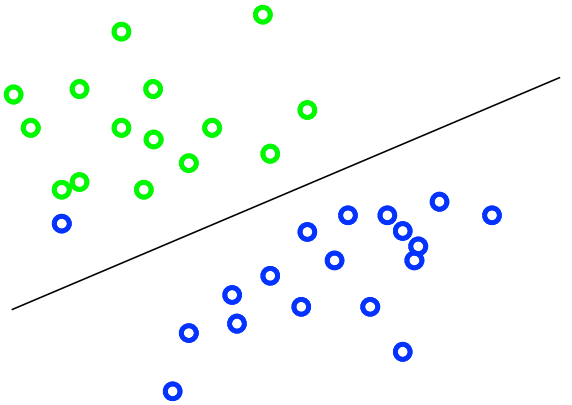
What's the best **w**?



Intuitively, we should allow for some misclassification if we can get more robust classification

147

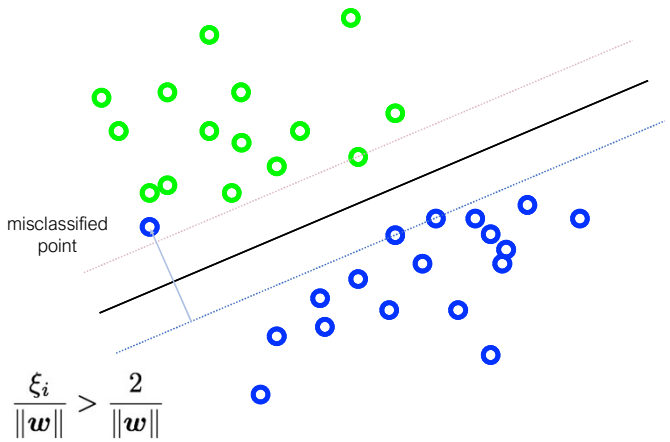
What's the best **w**?



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

148

Adding slack variables $\xi_i \geq 0$



149

'soft' margin

objective

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, N$$

150

'soft' margin

objective

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, N$$

The slack variable allows for mistakes, as long as the inverse margin is minimized.

151

'soft' margin

objective

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

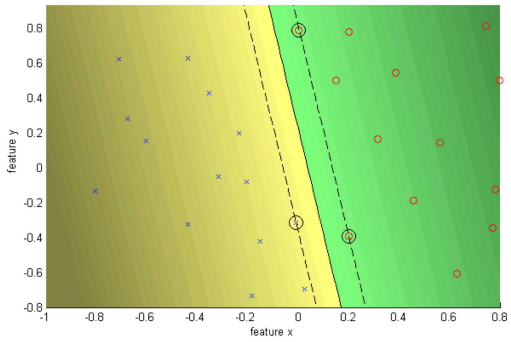
subject to

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, N$$

- 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)

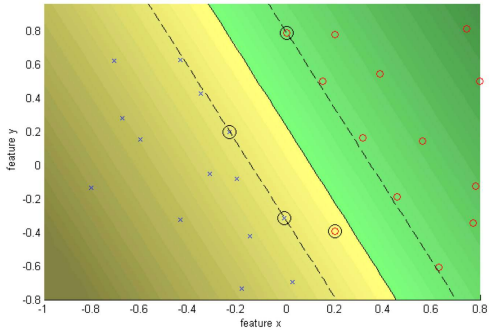
152

C = Infinity hard margin



153

C = 10 soft margin



154

References

Most of these slides were adapted from:

1. Ioannis Yannis, Gkioulekas (16-385 Computer Vision, Spring 2020, CMU)

2. Kristen Grauman (CS 376: Computer Vision, Spring 2018, The University of Texas at Austin)

3. Noah Snavely (Cornell University)

4. Fei-Fei Li (Stanford University)

155

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you!

soict.hust.edu.vn/ fb.com/groups/soict

156