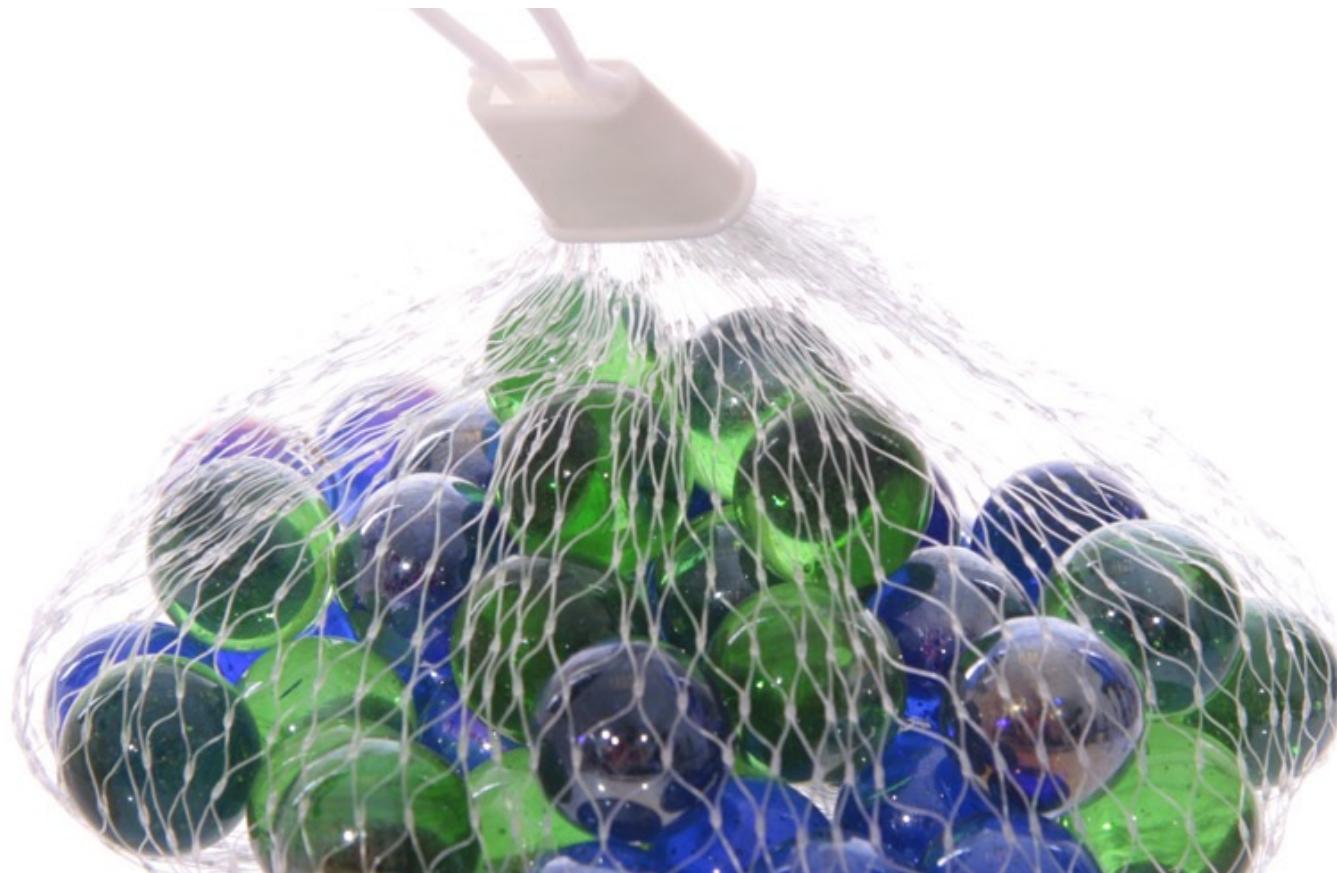


Bag of Words Model



Overview of today's lecture

- Bag-of-words.
- K-means clustering.
- Classification.
- K nearest neighbors.
- Support vector machine.

Image Classification



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



cat

Image Classification: Problem



08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	66
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	58	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	62	21	65	56	01	32	56	71	37	02	36	91
22	31	16	71	51	62	02	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	33	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
03	44	68	67	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	35	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	67	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	45	55	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	23	57	48

What the computer sees

image classification

82% cat
15% dog
2% hat
1% mug

Data-driven approach

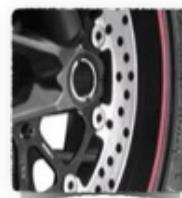
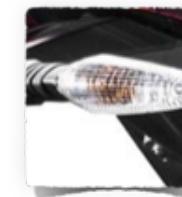
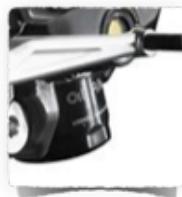
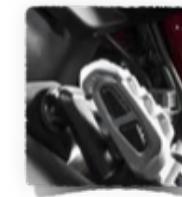
- Collect a database of images with labels
- Use ML to train an image classifier
- Evaluate the classifier on test images

Example training set

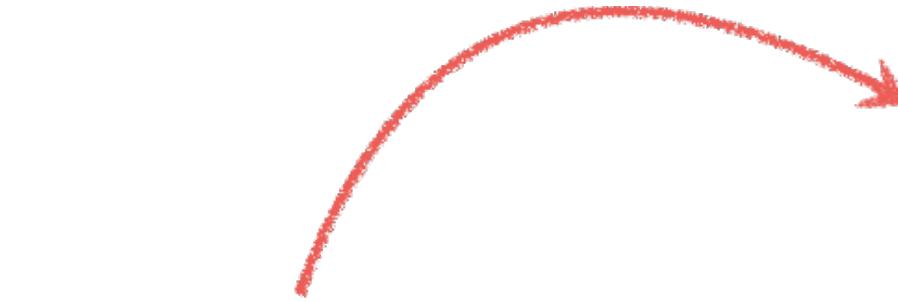


Bag of words

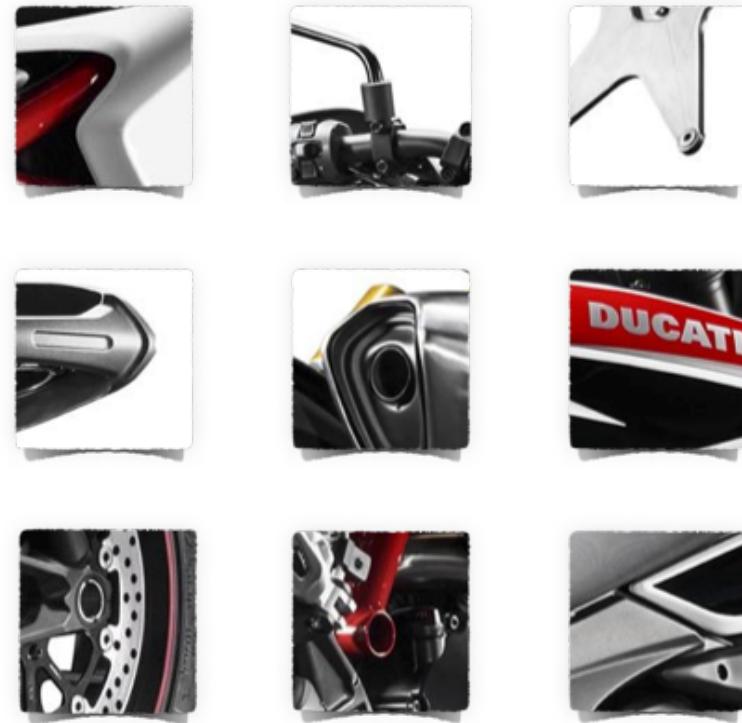
What object do these parts belong to?



Some local feature are very informative



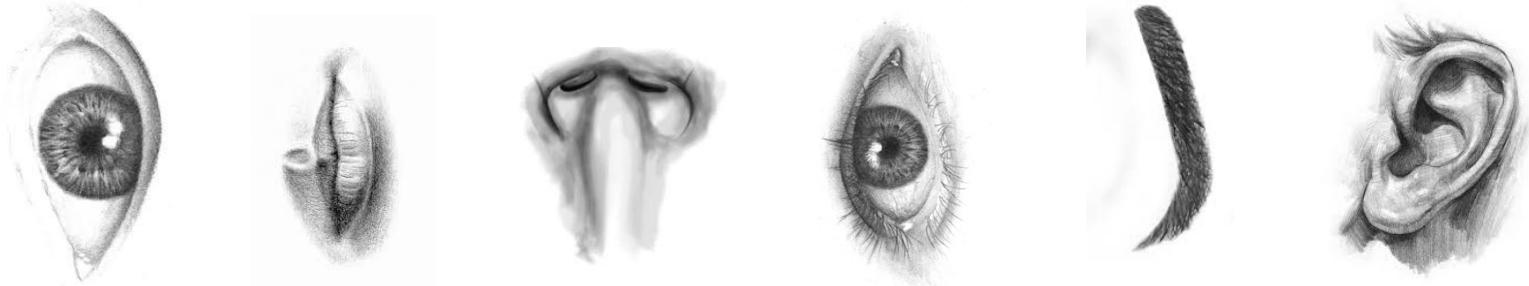
An object as



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)

CalTech6 dataset



class	bag of features	bag of features	Parts-and-shape model
	Zhang et al. (2005)	Willamowski et al. (2004)	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)

Standard BOW pipeline

(for image classification)

Dictionary Learning:
Learn Visual Words using clustering

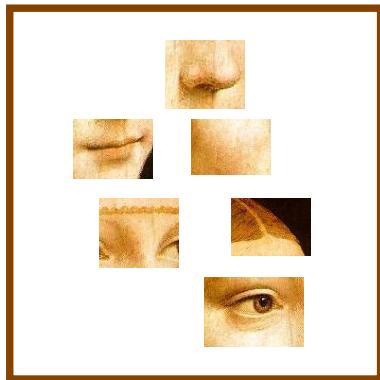
Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Dictionary Learning:

Learn Visual Words using clustering

1. extract features (e.g., SIFT) from images



Dictionary Learning:

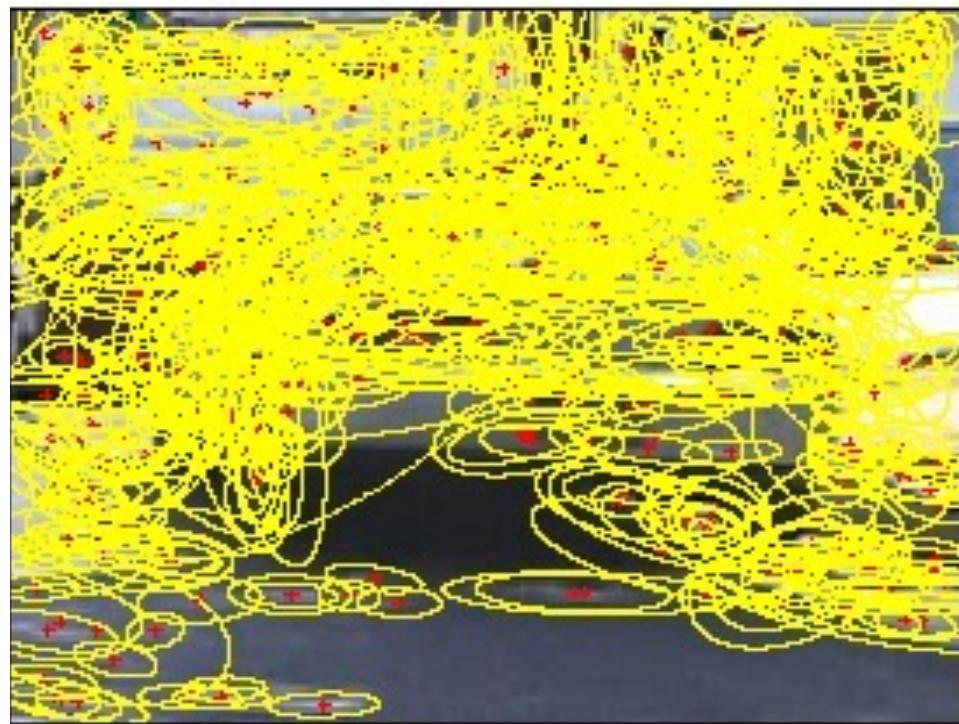
Learn Visual Words using clustering

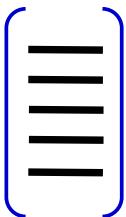
2. Learn visual dictionary (e.g., K-means clustering)



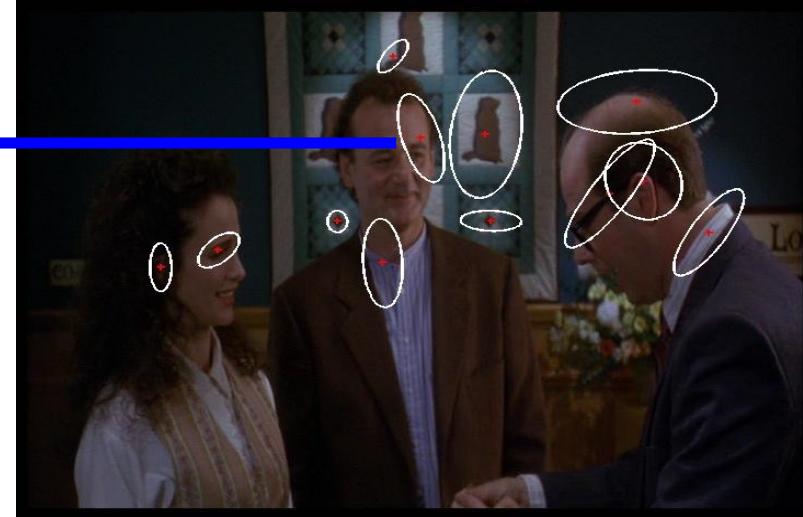
What kinds of features can we extract?

- 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)



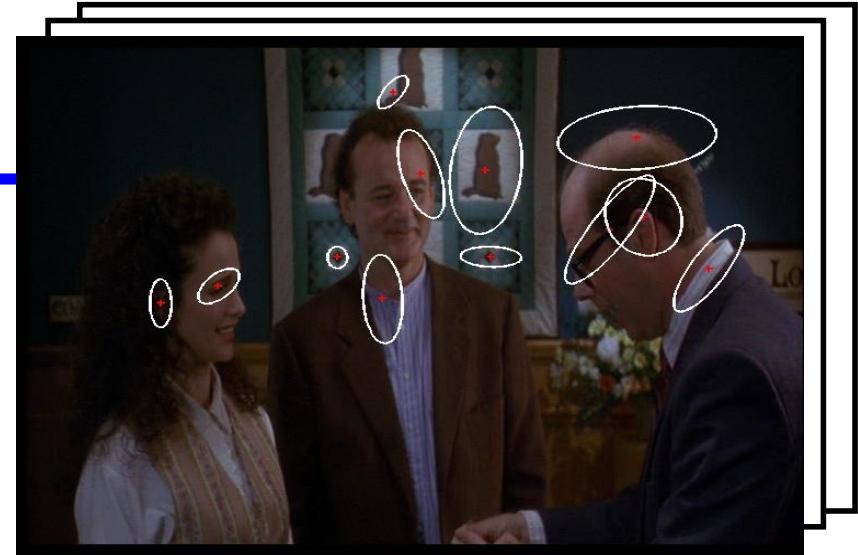
 ←
**Compute SIFT
descriptor**
[Lowe'99]

←
Normalize patch

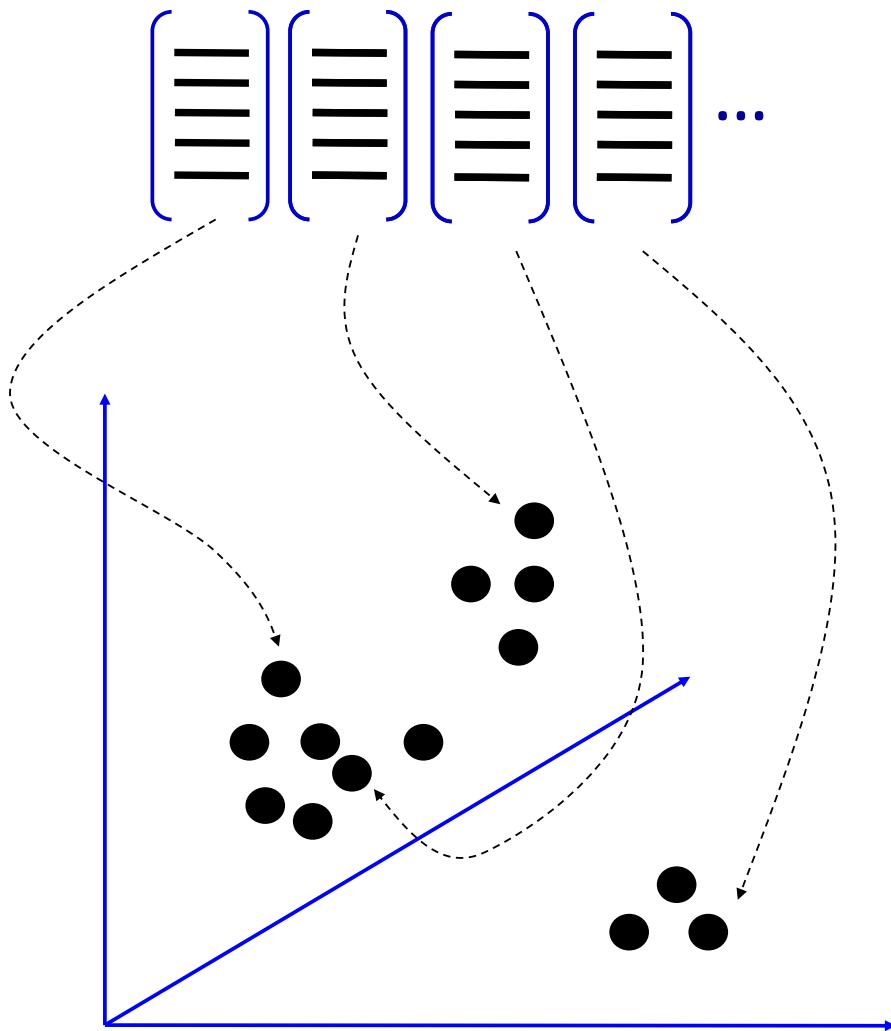


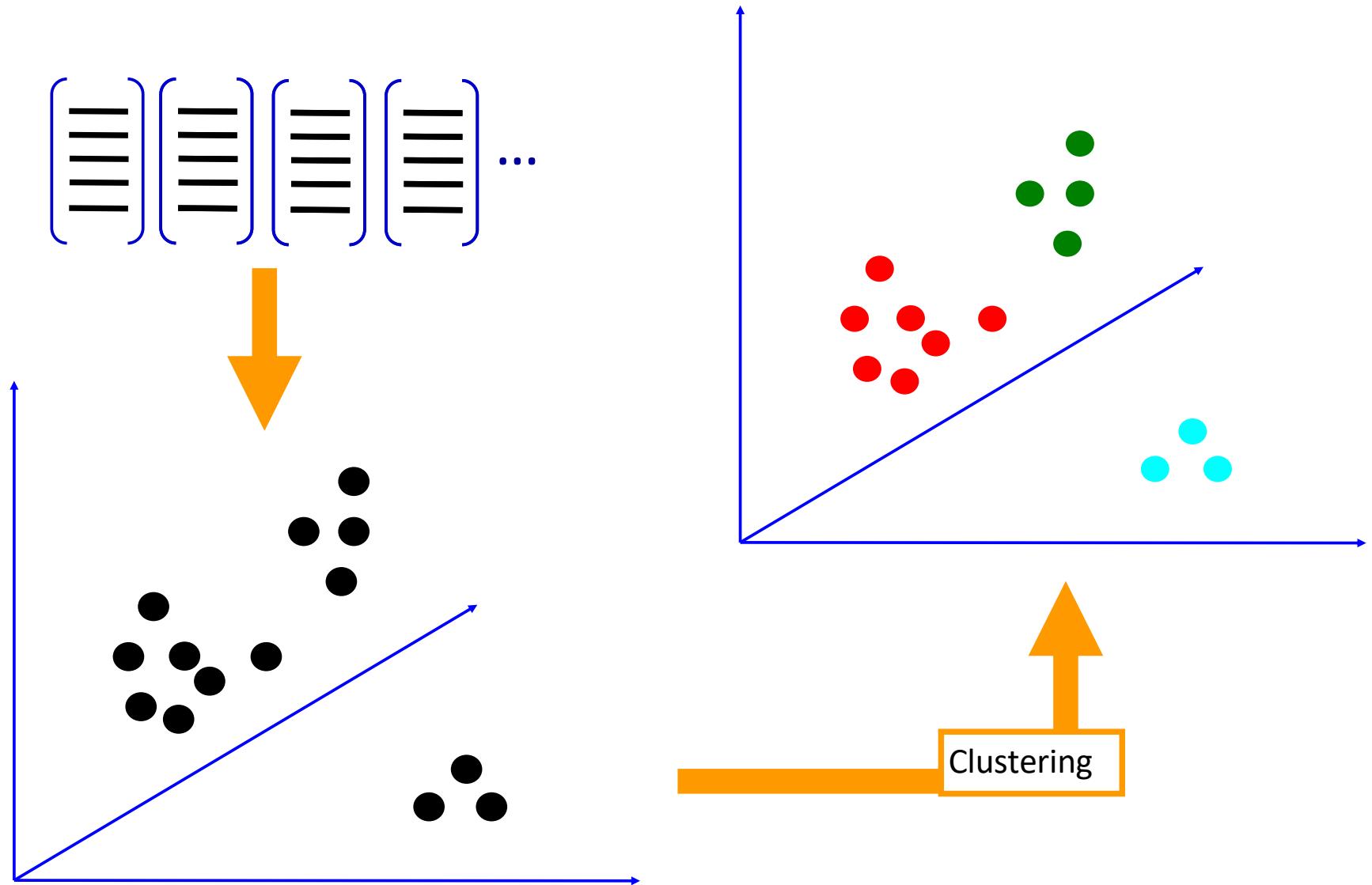
Detect patches
[Mikojaczyk and Schmid '02]
[Mata, Chum, Urban & Pajdla, '02]
[Sivic & Zisserman, '03]

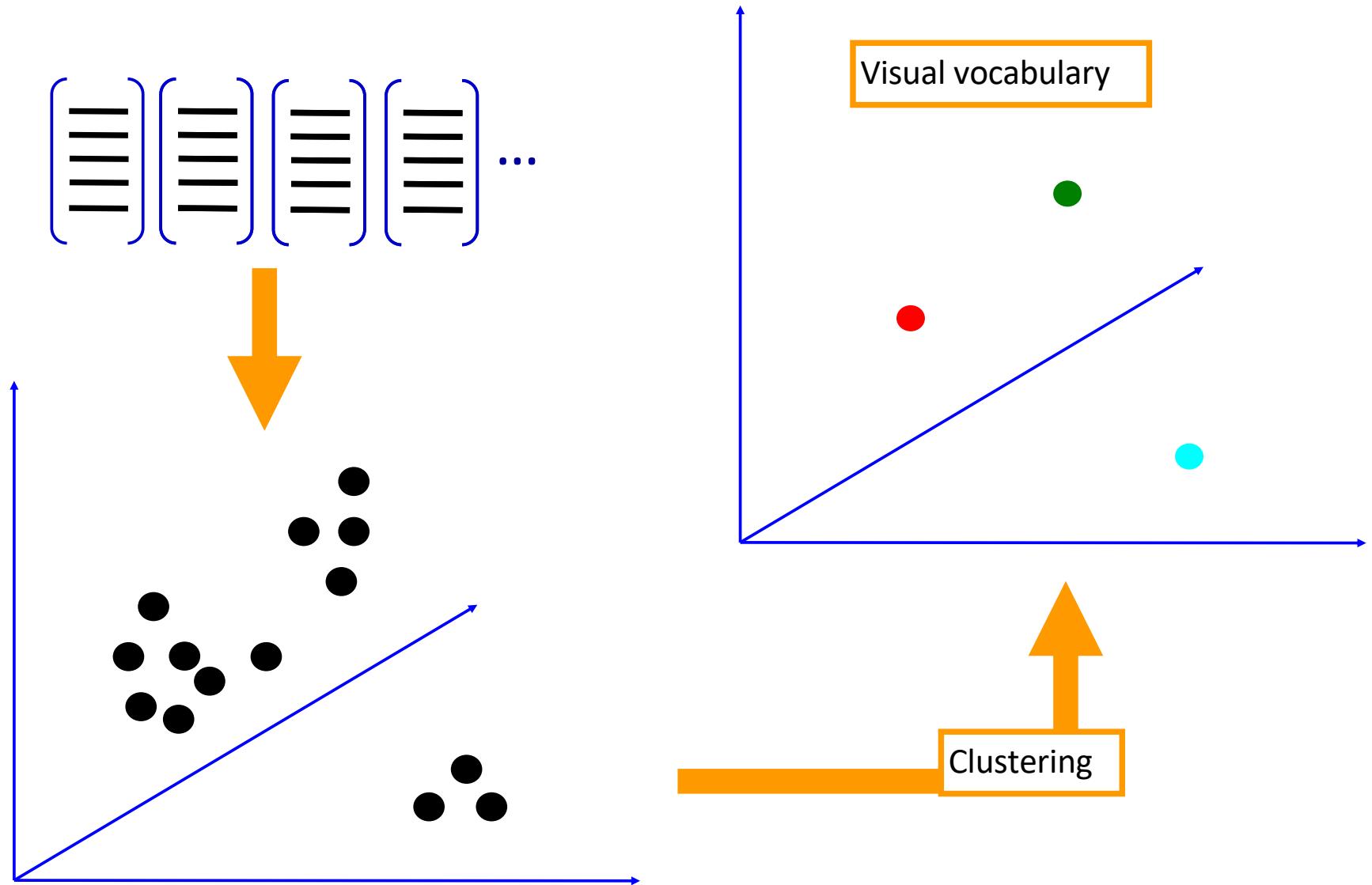
$$\left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \dots$$



How do we learn the dictionary?







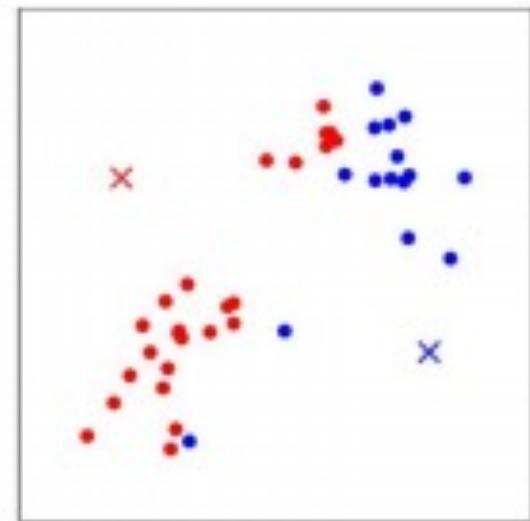
K-means clustering



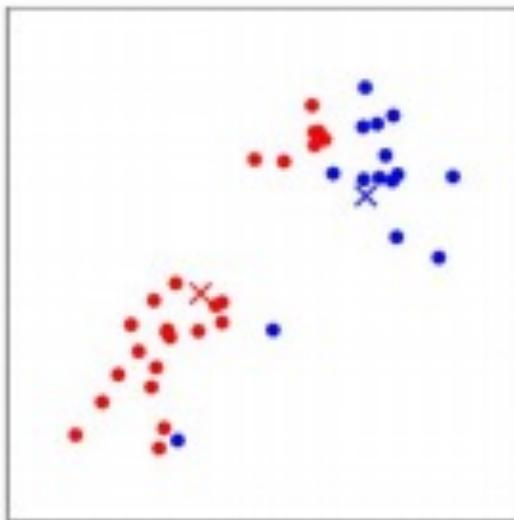
(a)



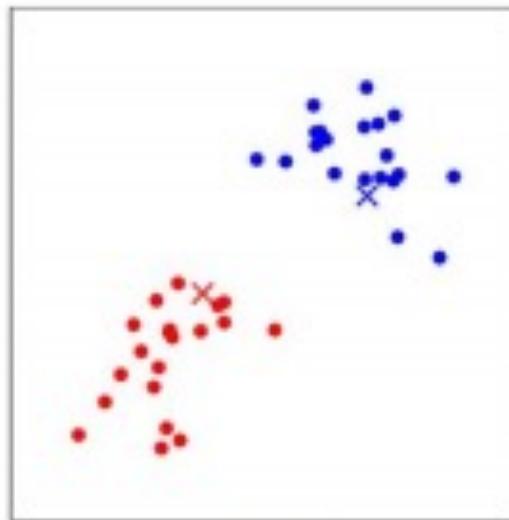
(b)



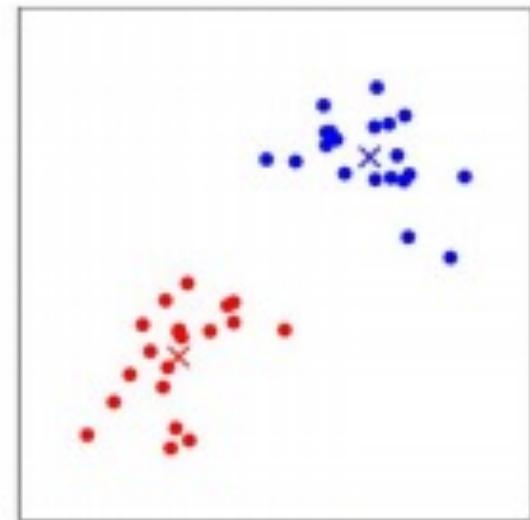
(c)



(d)



(e)



(f)

[Stanford CS221]

K-means Clustering

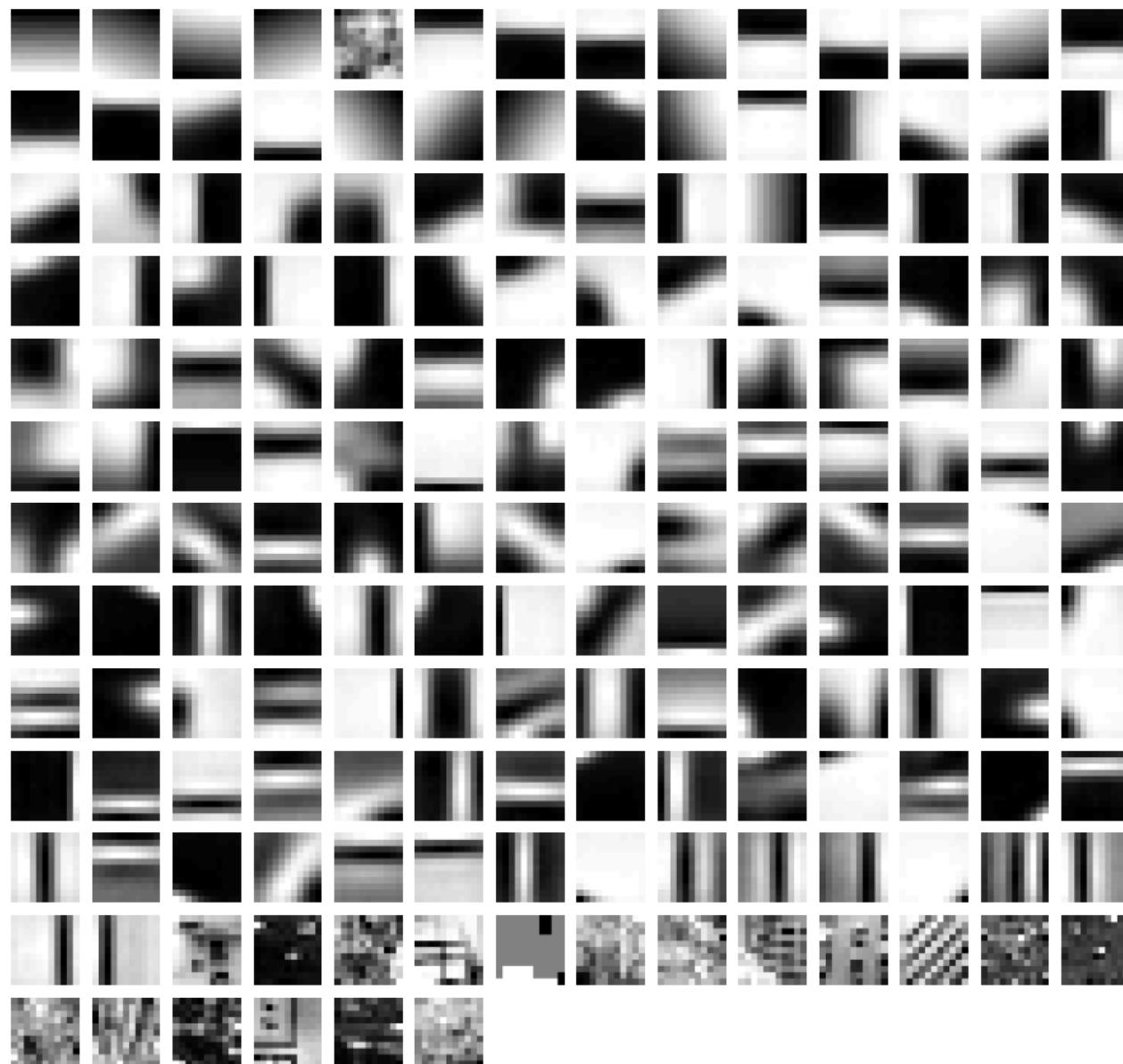
Given k:

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.

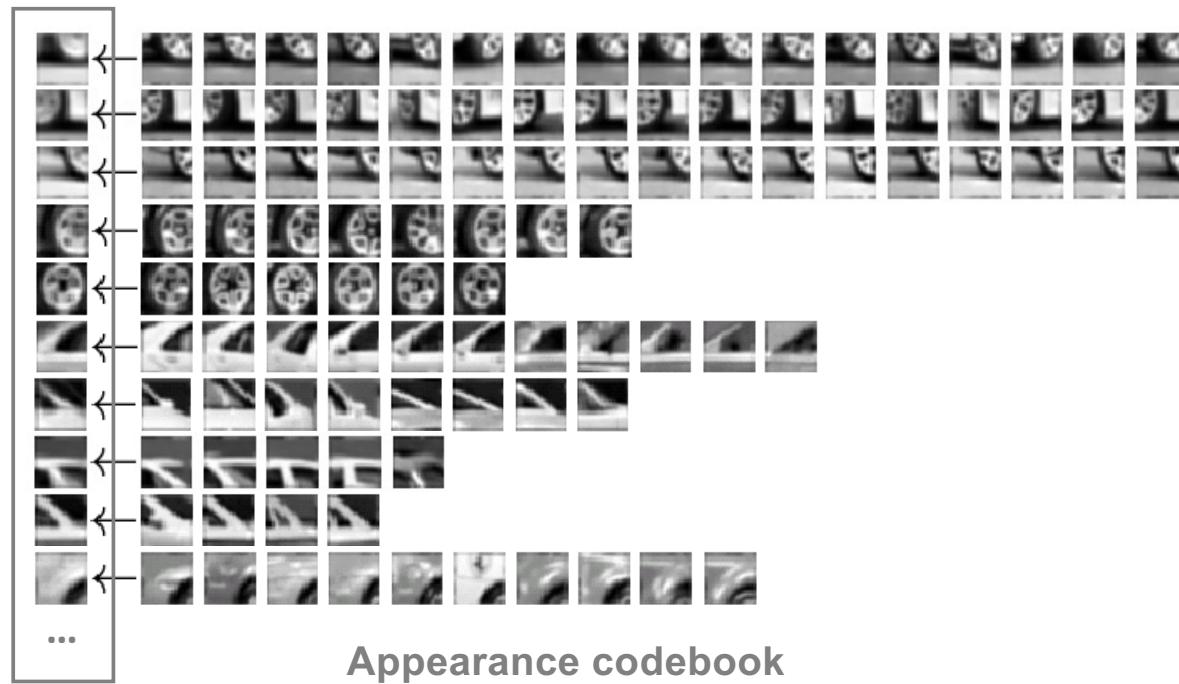
*From what **data** should I learn the dictionary?*

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

Example visual dictionary



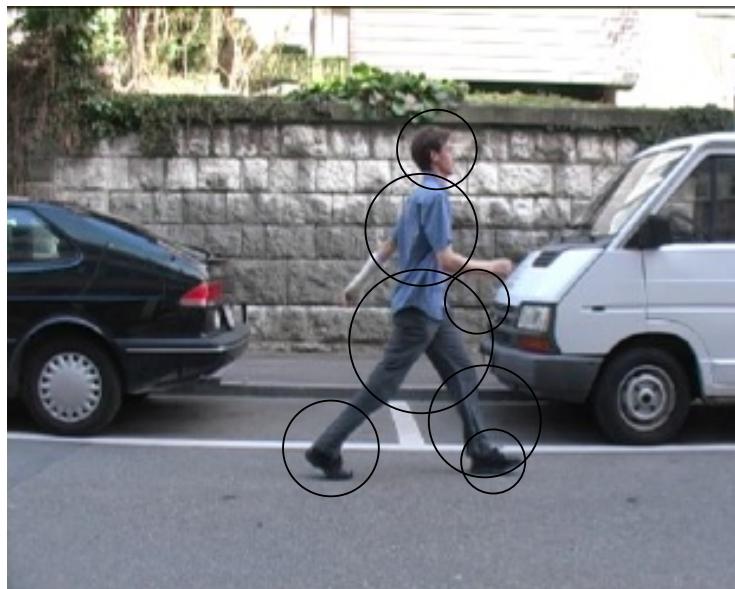
Example dictionary



Appearance codebook

Source: B. Leibe

Another dictionary



Source: B. Leibe

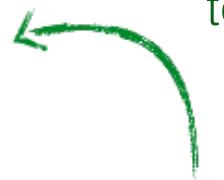
Dictionary Learning:

Learn Visual Words using clustering

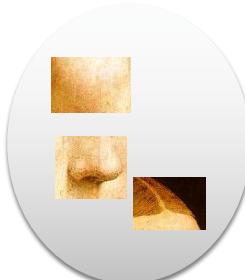
Encode:
build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

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



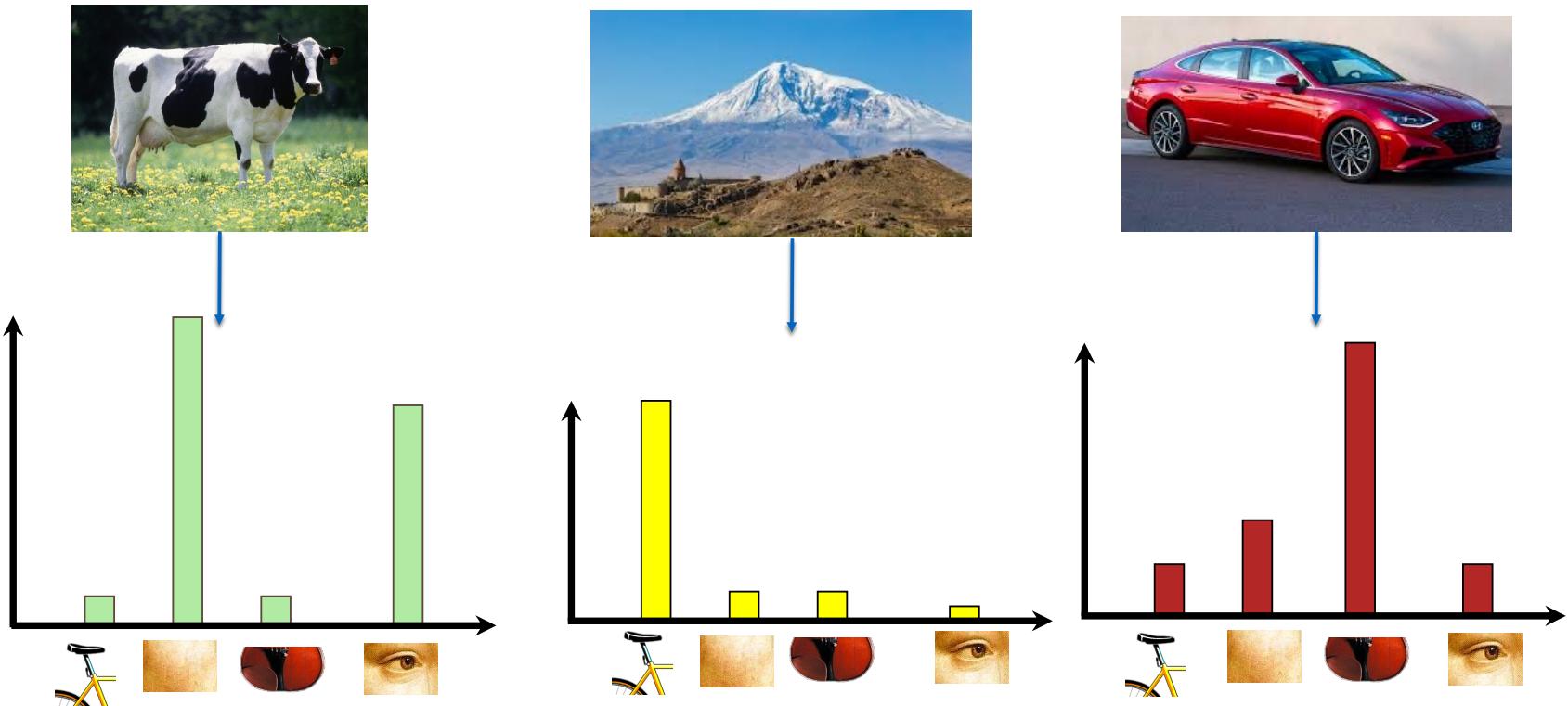
Encode:
build Bags-of-Words (BOW) vectors
for each image

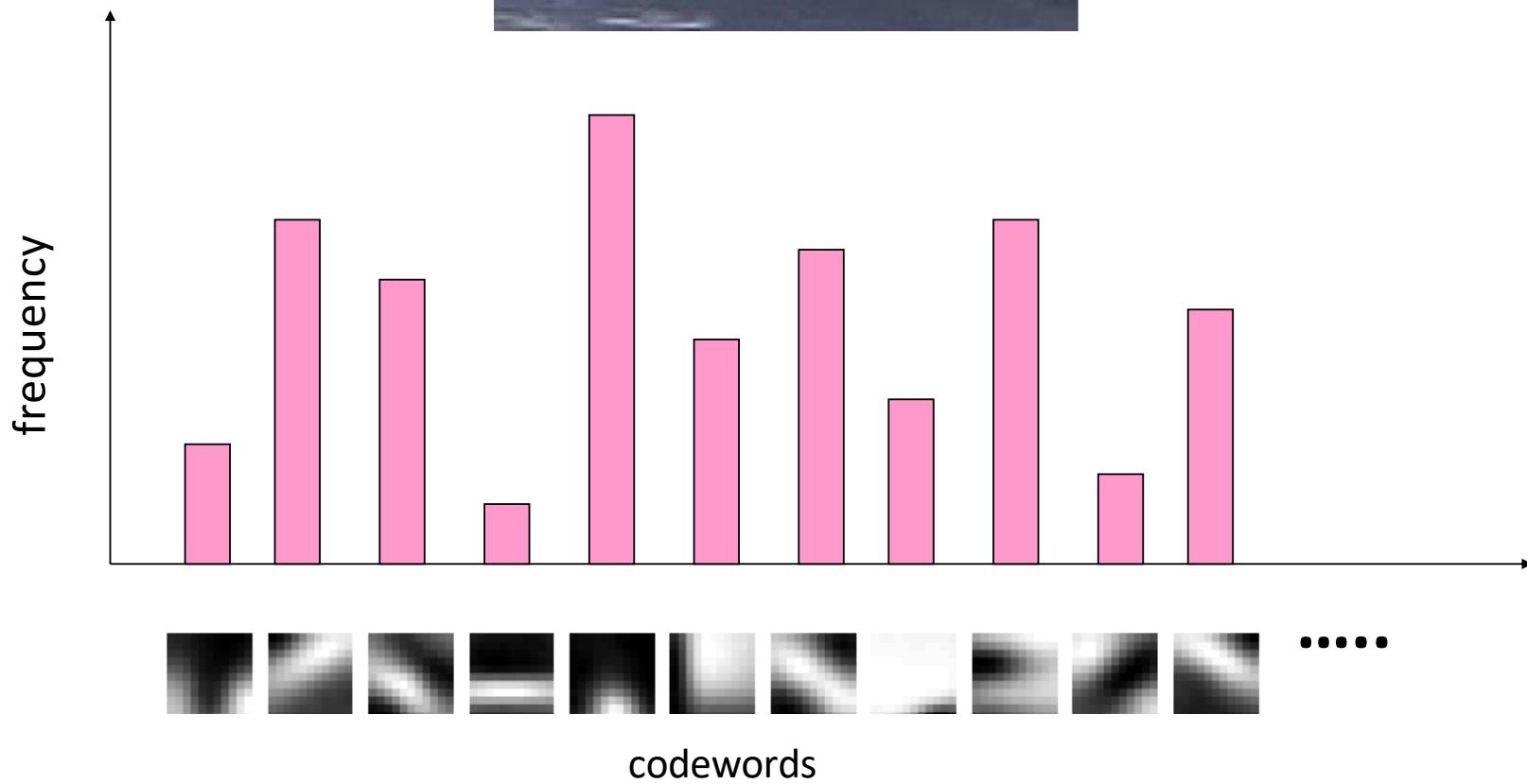


2. Histogram: count the number of visual word occurrences

Encode:

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



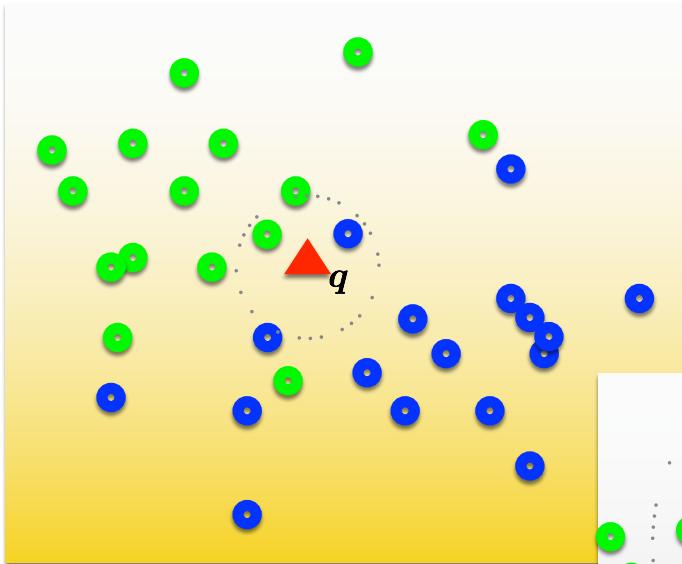


Dictionary Learning:

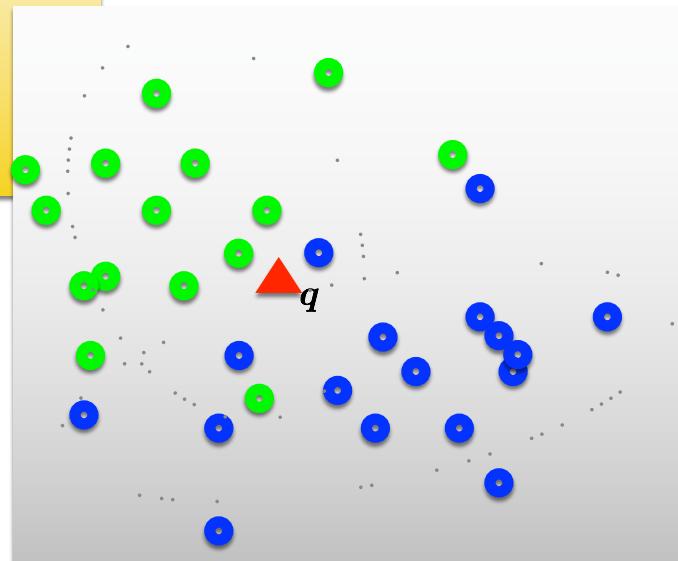
Learn Visual Words using clustering

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

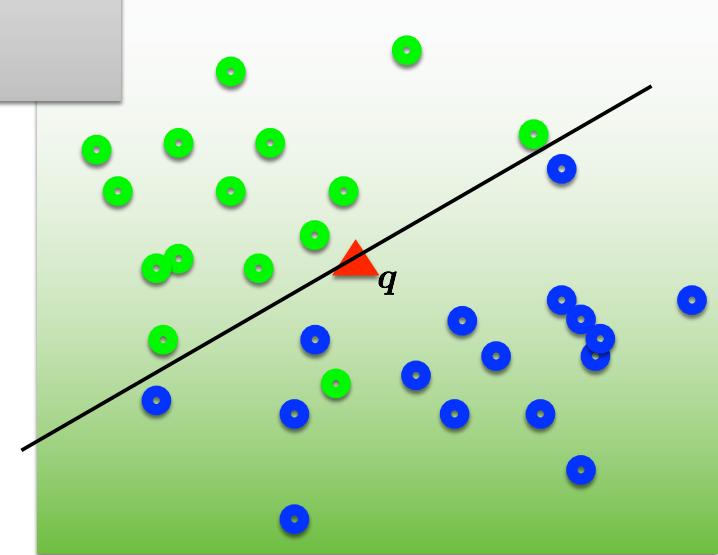
Classify:
Train and test data using BOWs



K nearest neighbors



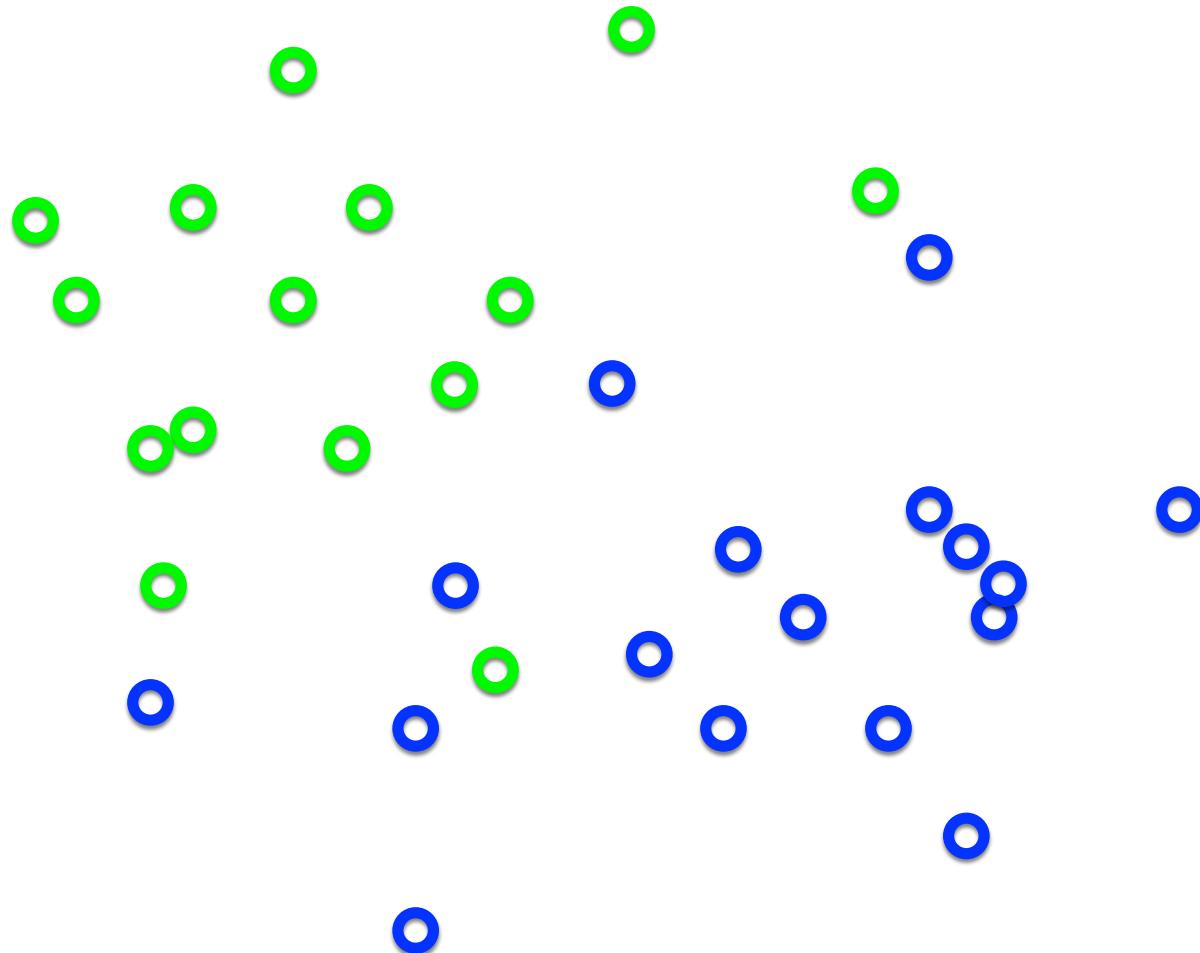
Naïve Bayes



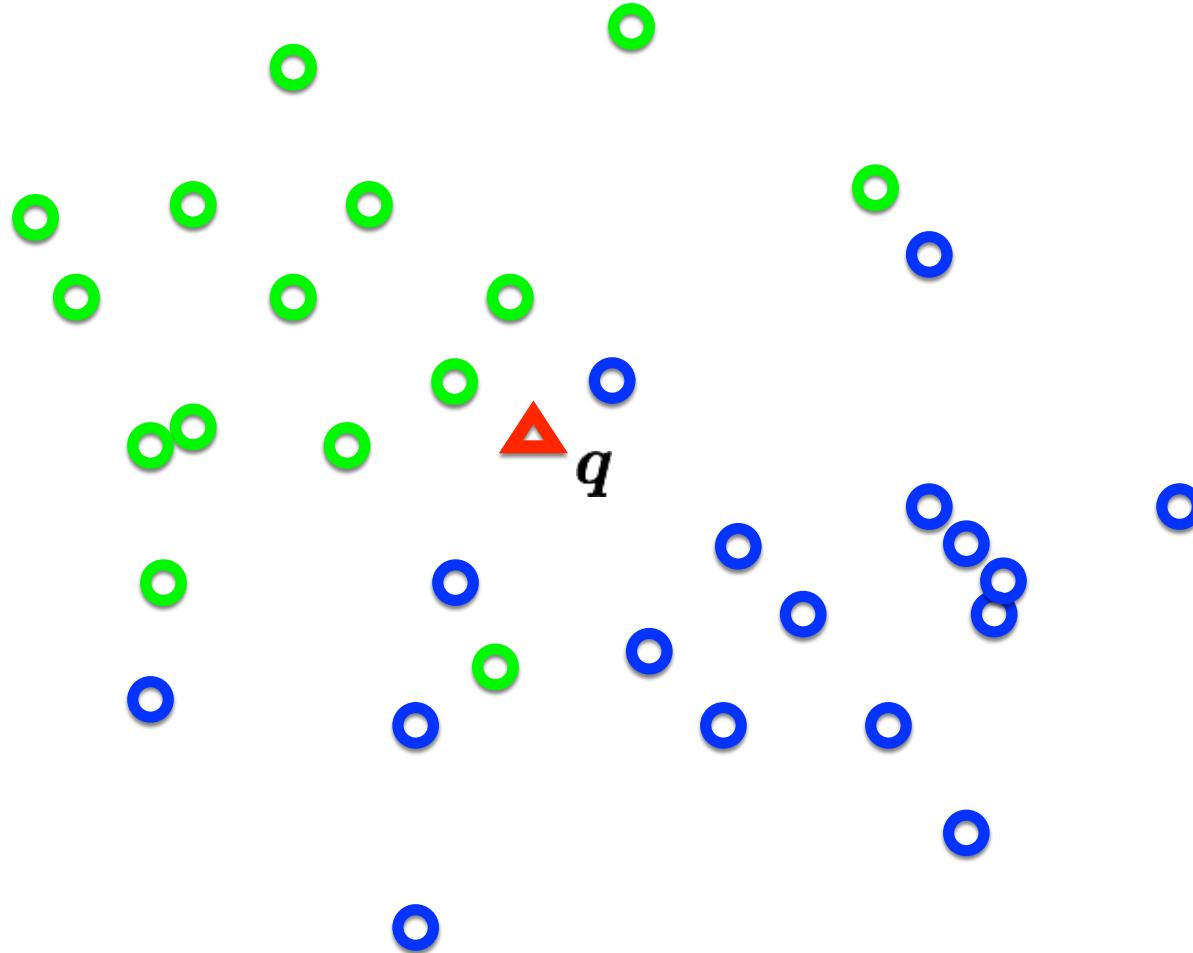
Support Vector Machine

K nearest neighbors

Distribution of data from two classes

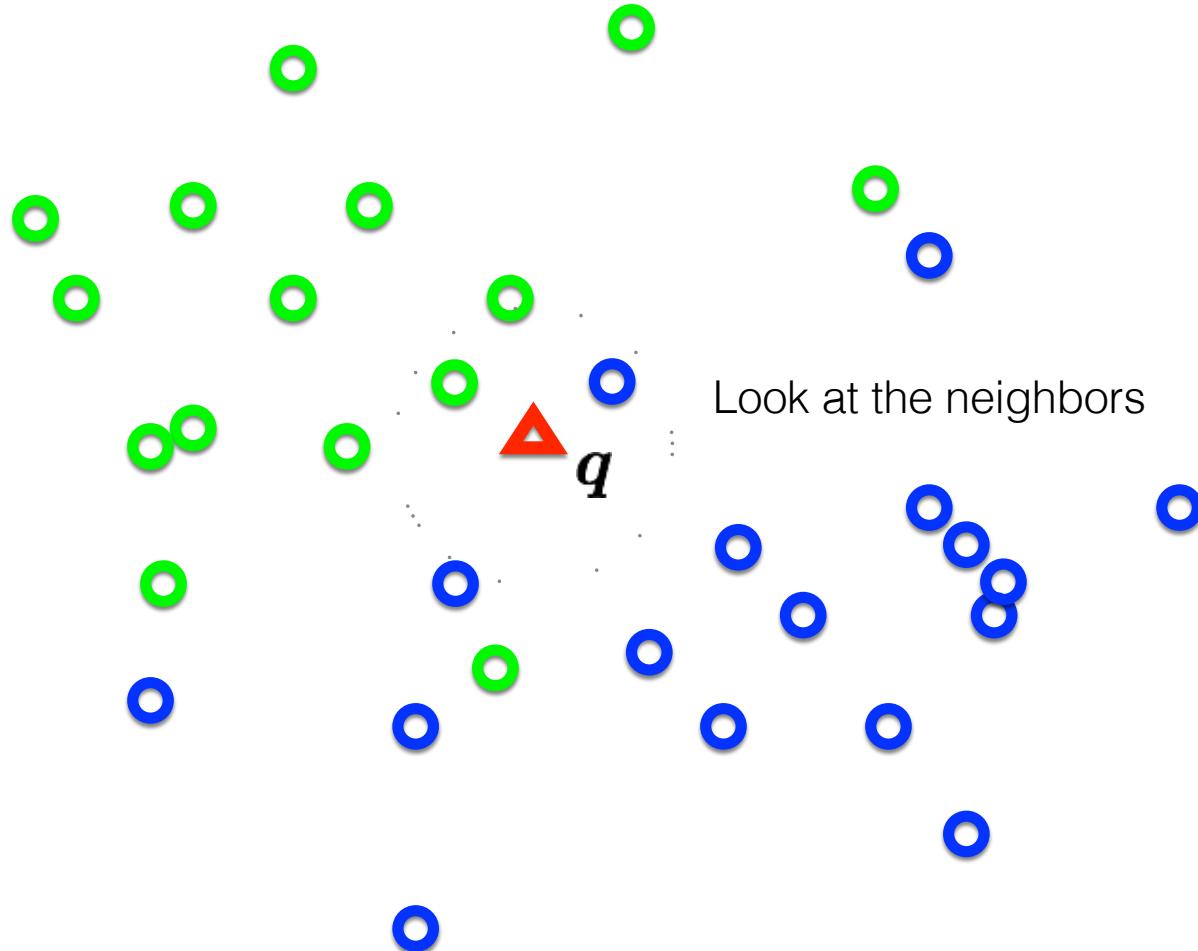


Distribution of data from two classes

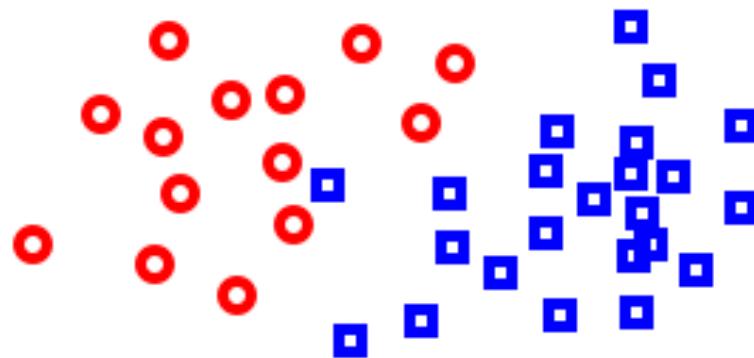


Which class does q belong to?

Distribution of data from two classes



K-Nearest Neighbor (KNN) Classifier



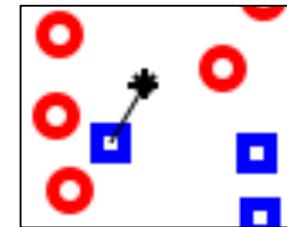
For a given query point q , assign the class of the nearest neighbor

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

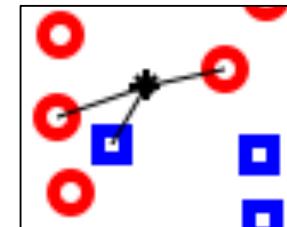
Non-parametric pattern classification approach

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

$k = 1$



$k = 3$



Nearest Neighbor is competitive

40281508803277064755729284686500876/71127400176386420140578214711366
507111167679664143112410826340006330171113109975414895351982739901029
84686824679339431447059604446123364596856560641865284554770782237018
76953465018828357808571101378507110114527623028596972136418240510226
93477149064842728100783331976131605747595849916501380348220251514889
82049962335648092836457294912860704116759914592504108908989425798980
35517216919955162286714604033223689853854520563283995794671313660901
94568160413179951001162198403649071654525185470670258104571851900607
88573898868239756292881688879180172075190209862393802111142972512199
1485343475074881539593690363982128685539494251514414435912233029009
9319097549201051493361525220026601203025579550895032590884684546549
69285457999218340783934656219260061287982047750564674307507420899404
12845278113035703193631773084826529739099642972116747598821445161325
9066436772860830298325398001951396014171237974939282718091017796999
21010452828351781129784050788477858498138031745516574935471208160734
28308784084458566309376893495891288681379011470817457121130621280766
41992780136134111560707232522949812/61278000822922799275134941856283

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

What is the best distance metric between data points?

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

How many K?

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

Distance metrics

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

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

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

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

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

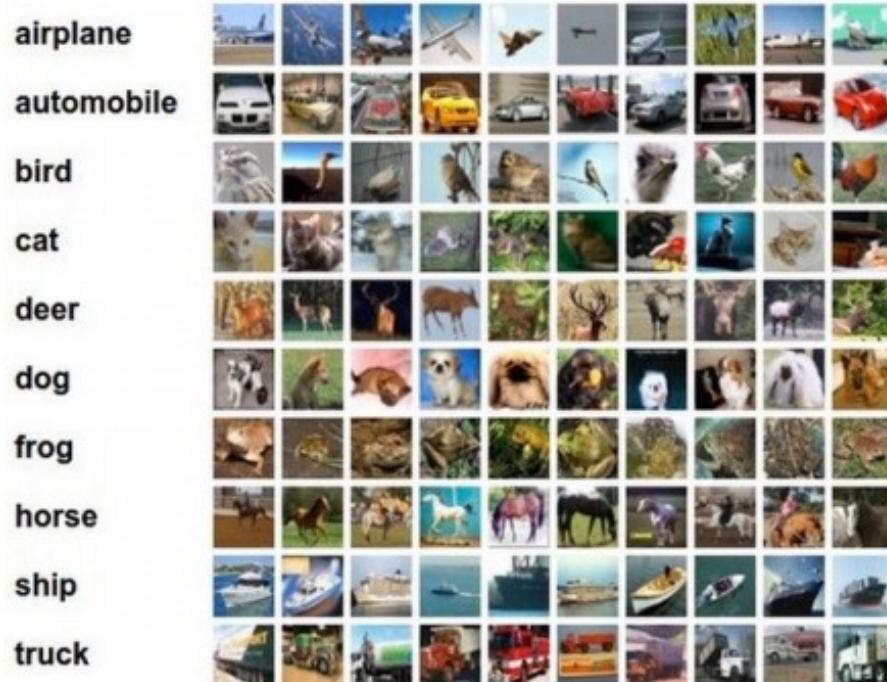
CIFAR-10 and NN results

Example dataset: **CIFAR-10**

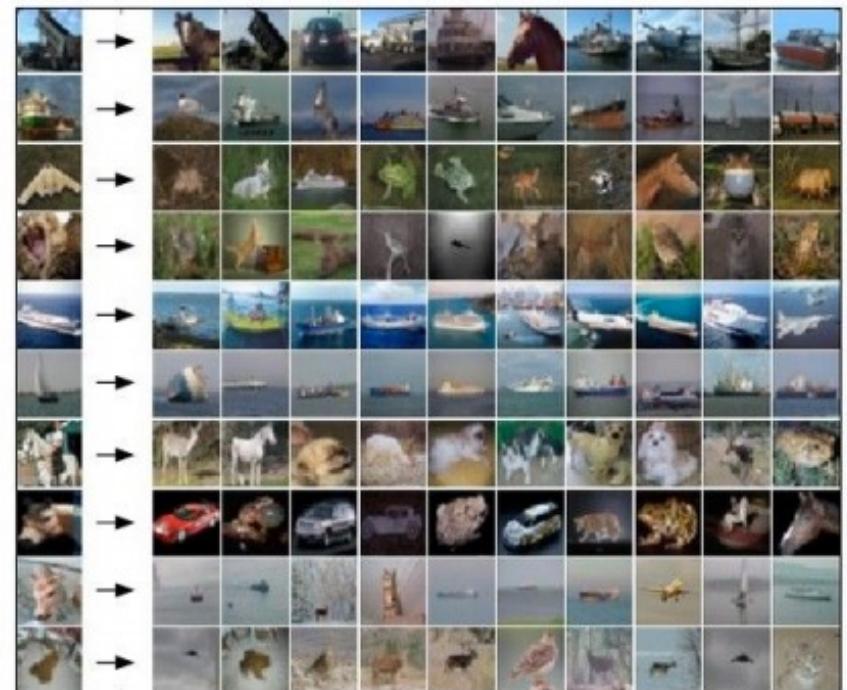
10 labels

50,000 training images

10,000 test images.



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



kNN

Pros

- simple yet effective

Cons

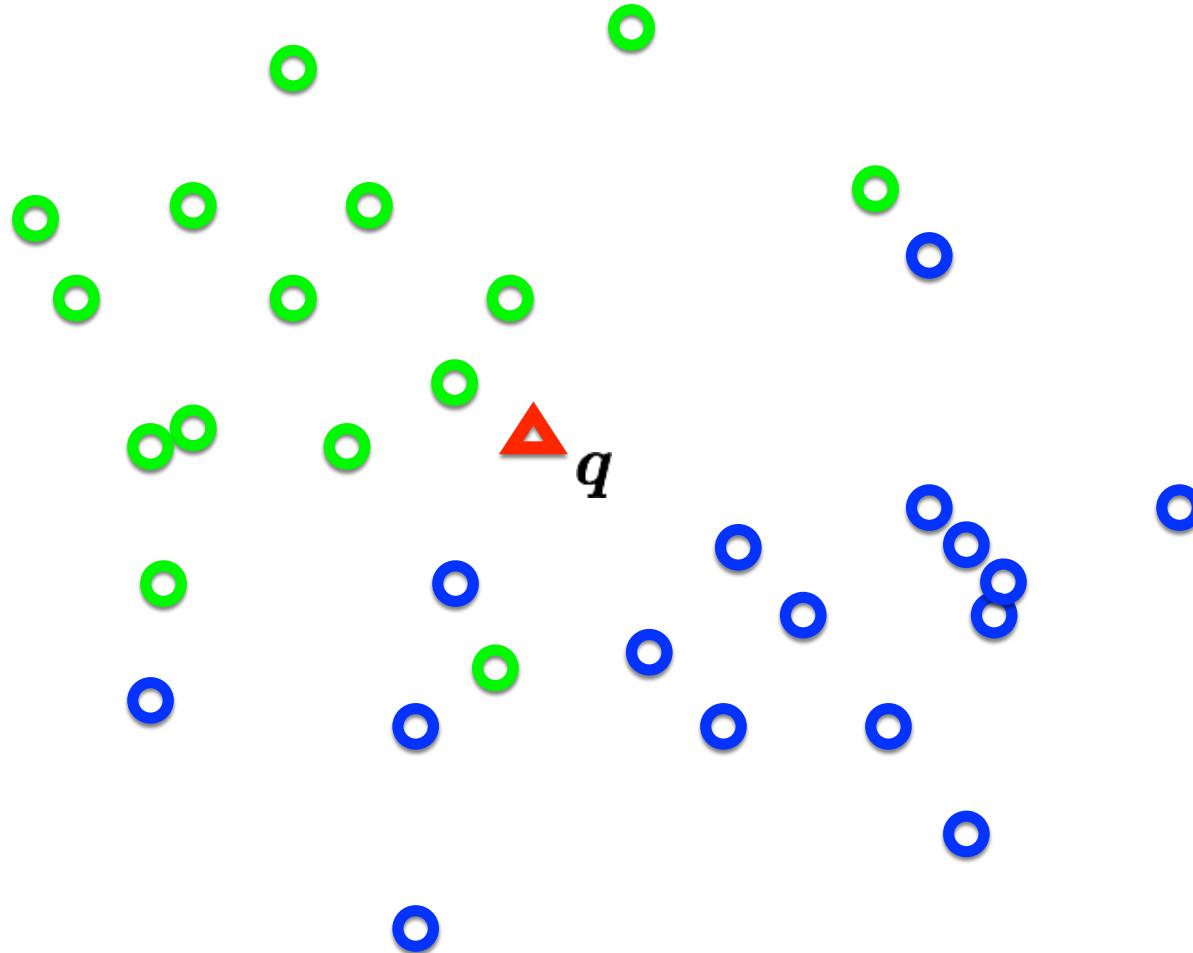
- search is expensive (can be sped-up)
- storage requirements
- difficulties with high-dimensional data

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)

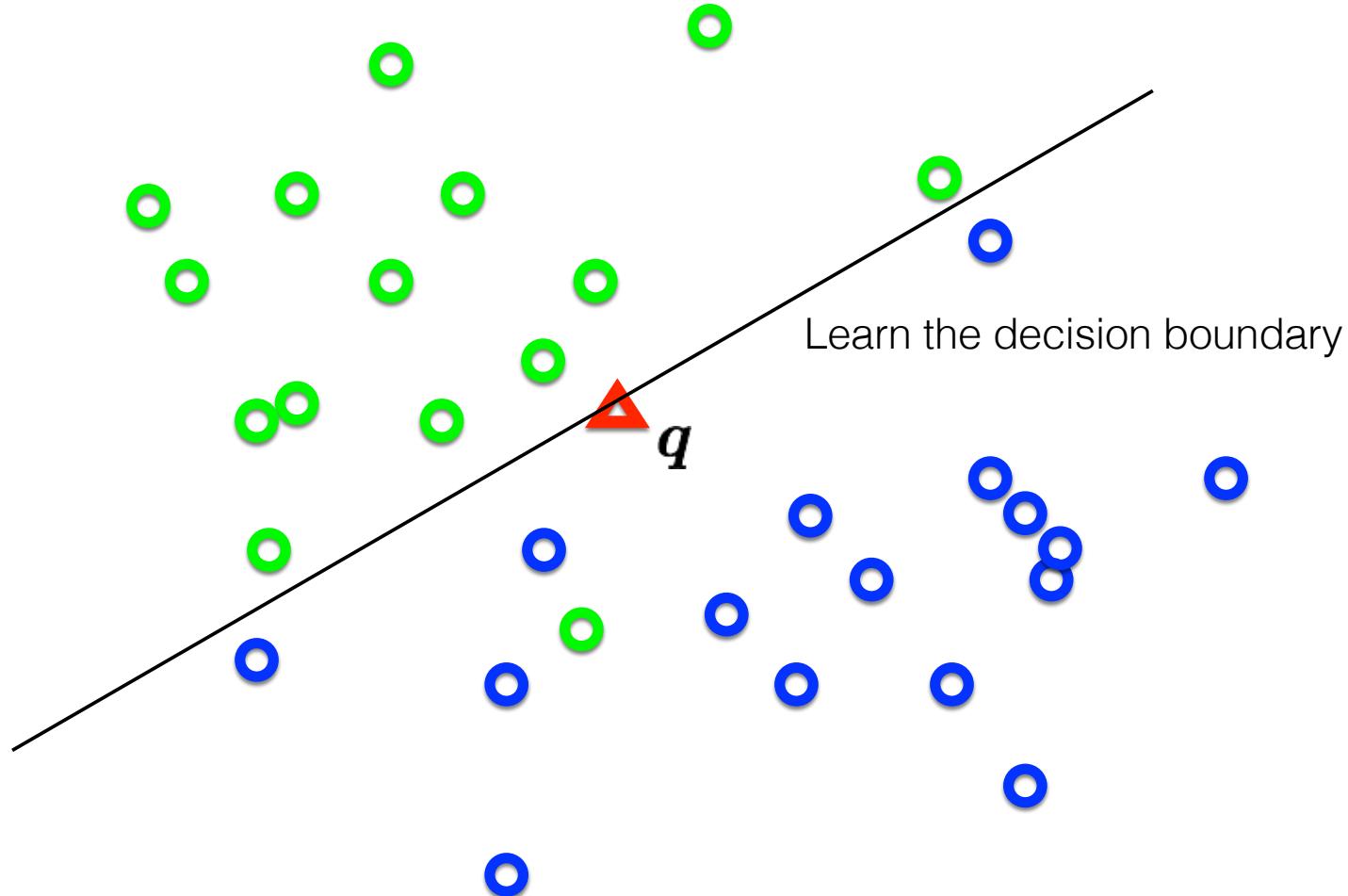
Support Vector Machine

Distribution of data from two classes

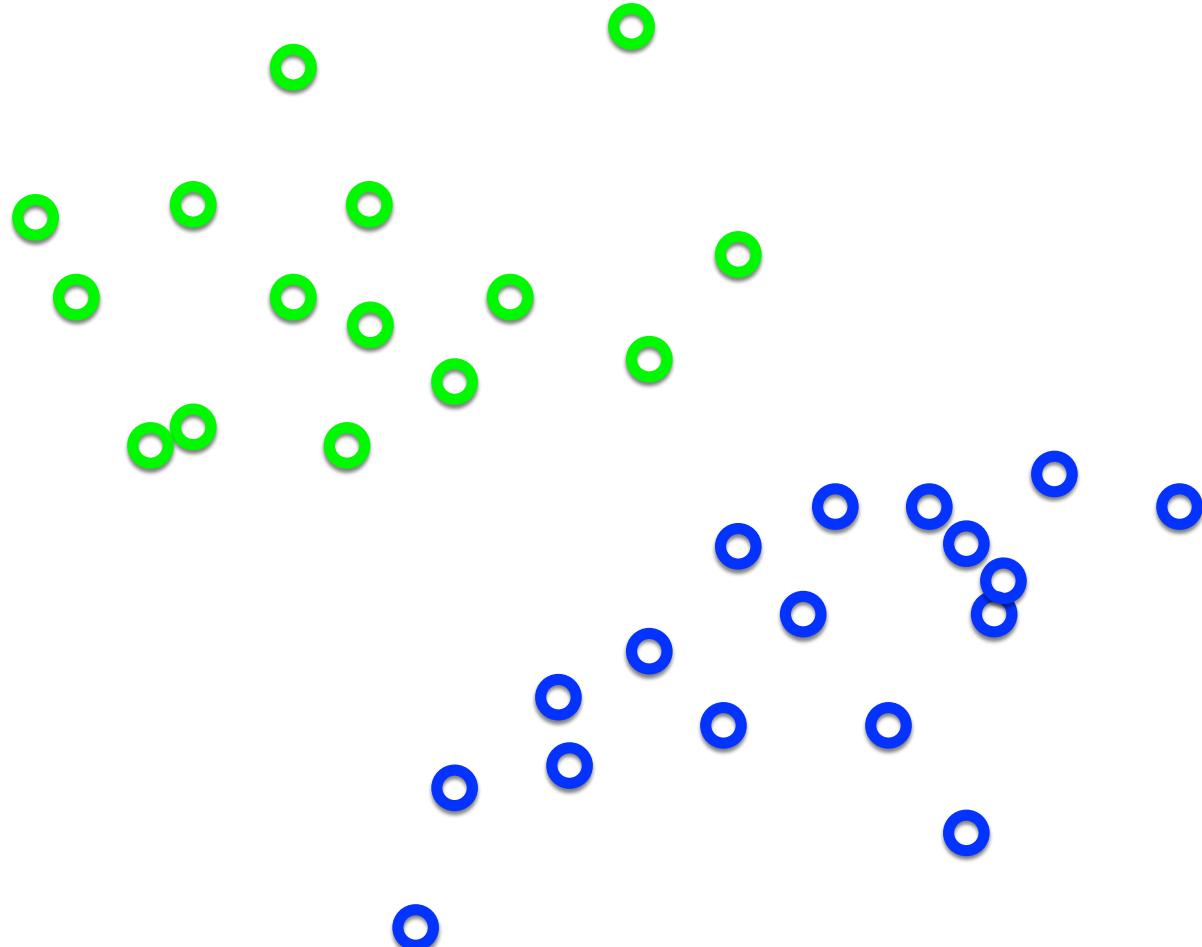


Which class does q belong to?

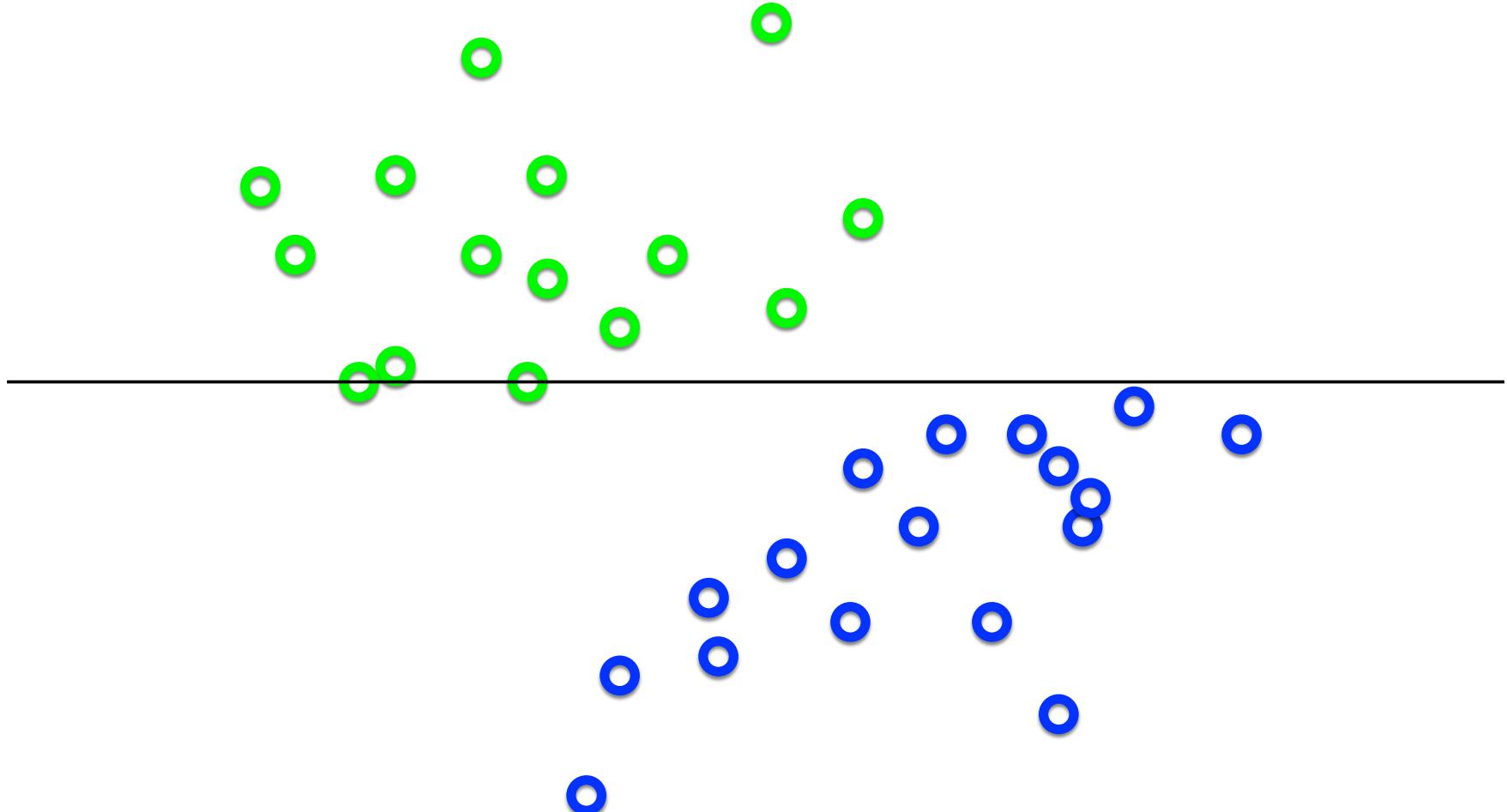
Distribution of data from two classes



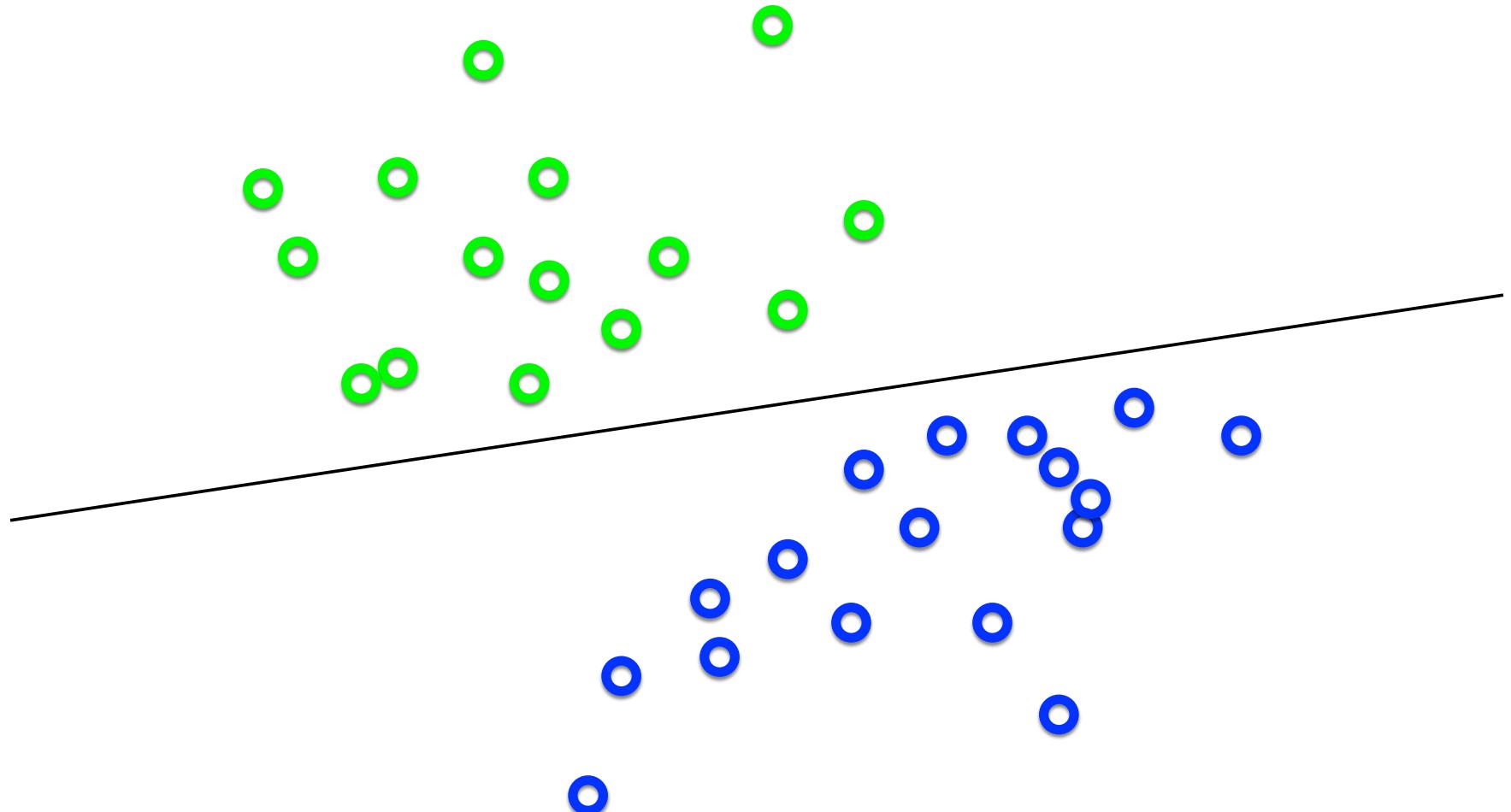
What's the best \mathbf{w} ?



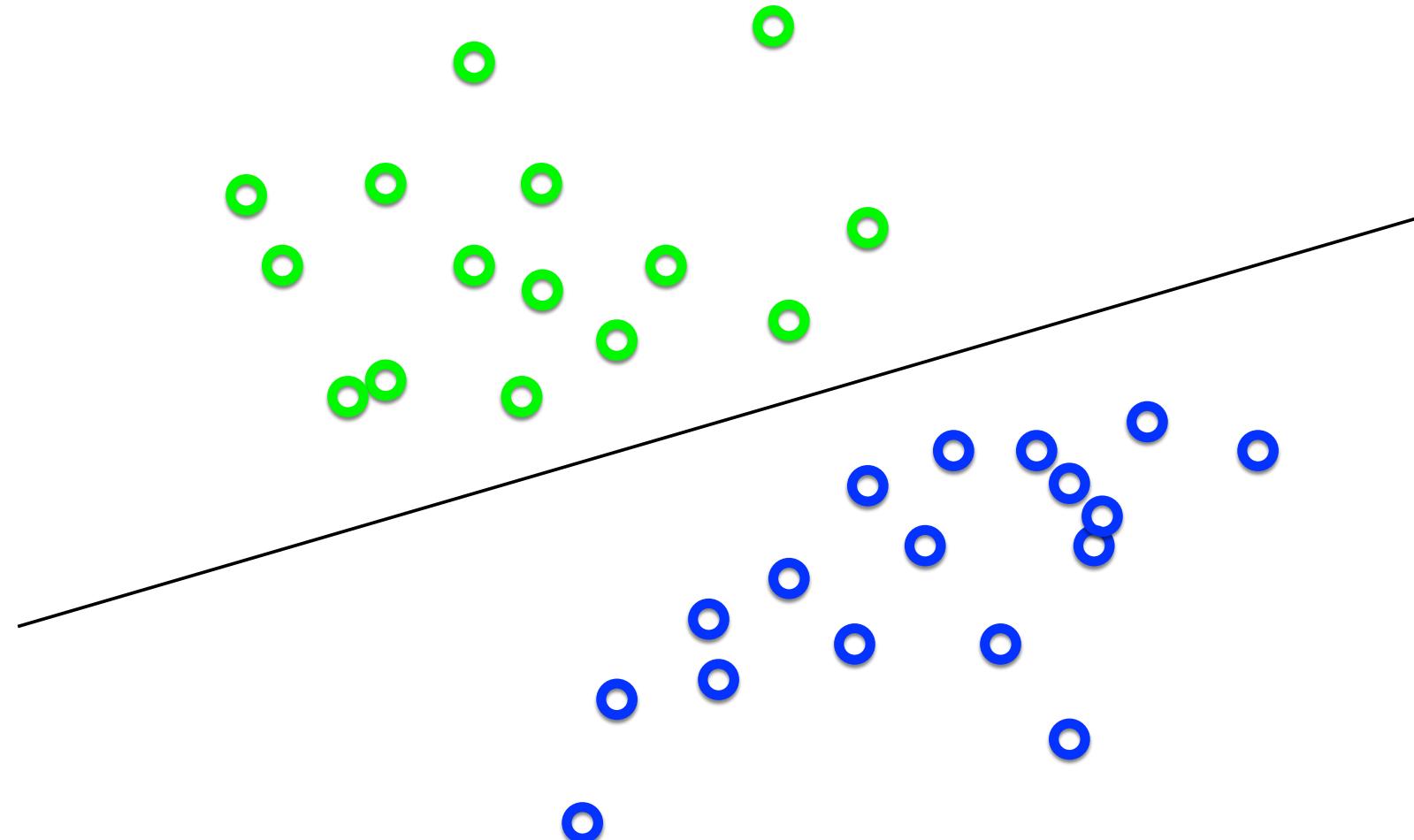
What's the best w ?



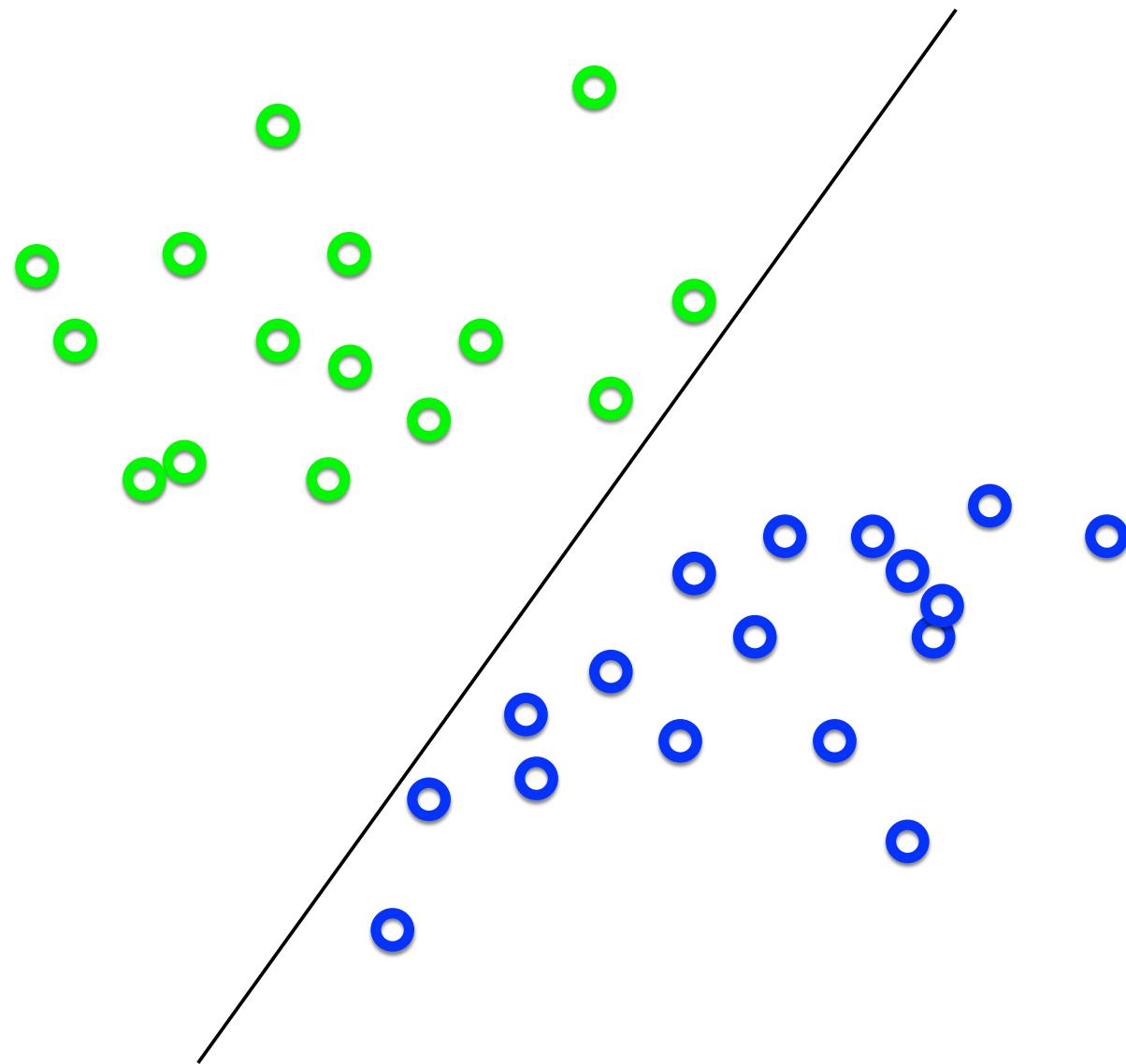
What's the best w ?



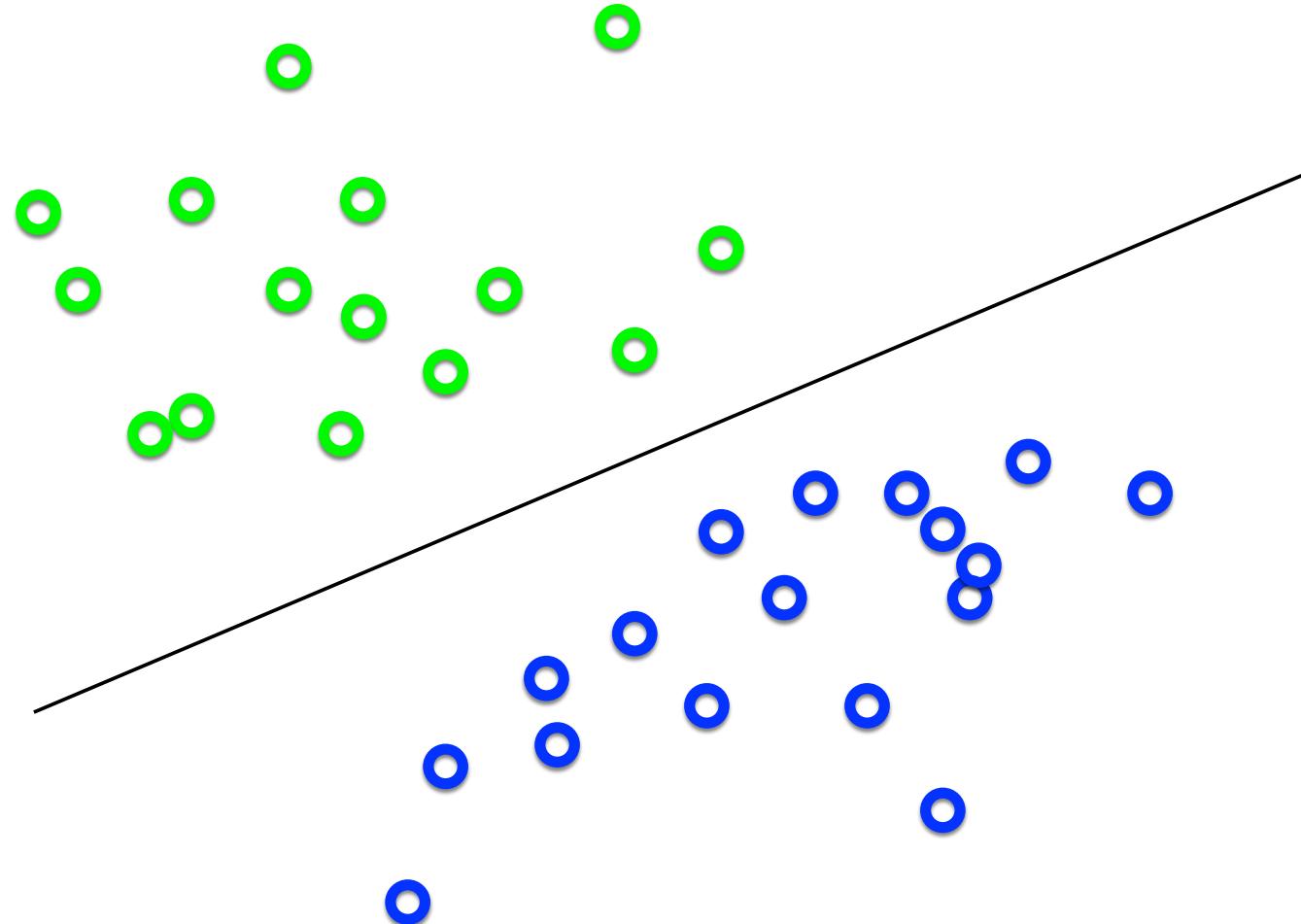
What's the best \mathbf{w} ?



What's the best \mathbf{w} ?

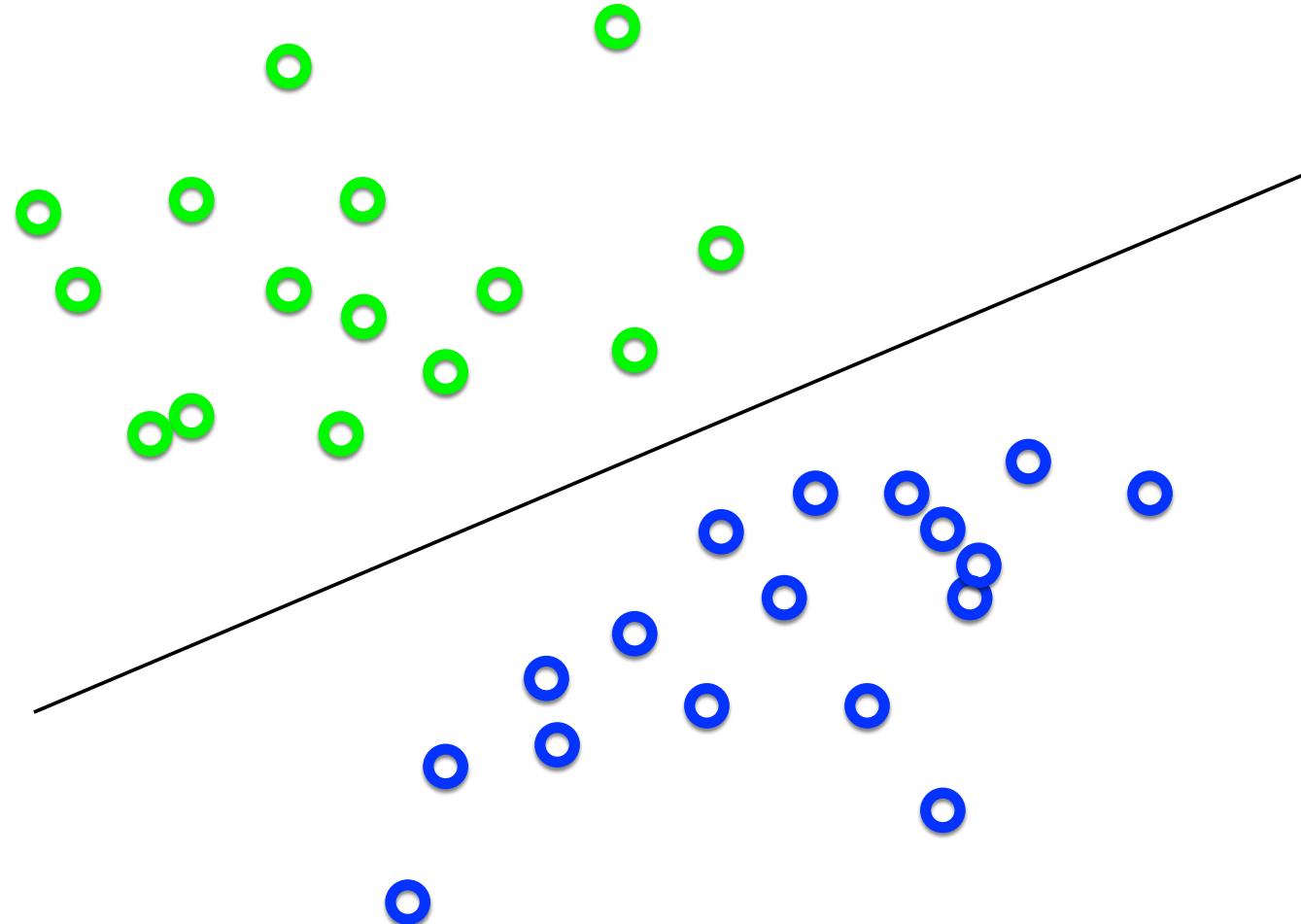


What's the best w ?



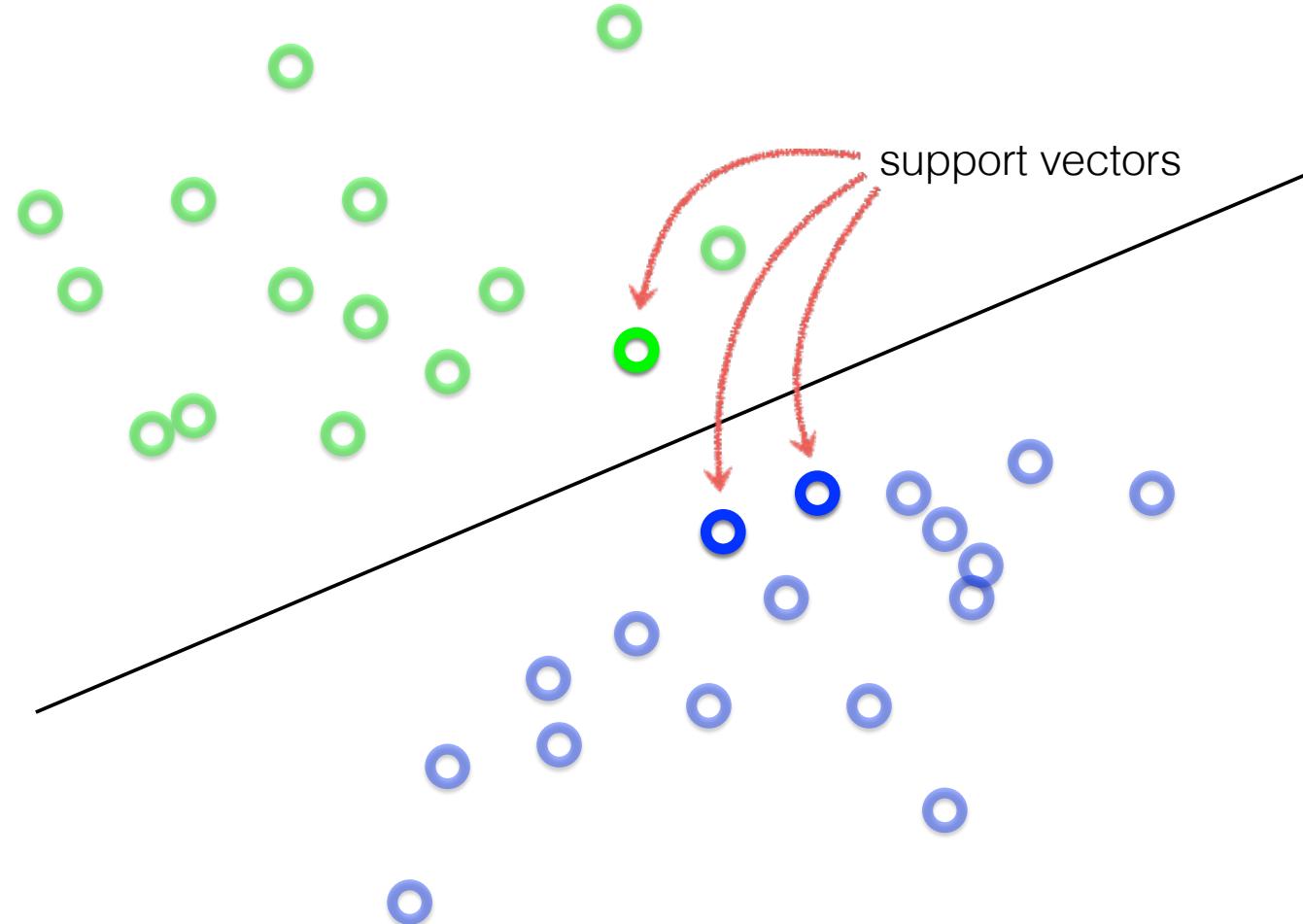
Intuitively, the line that is the
farthest from all interior points

What's the best w ?



Maximum Margin solution:
most stable to perturbations of data

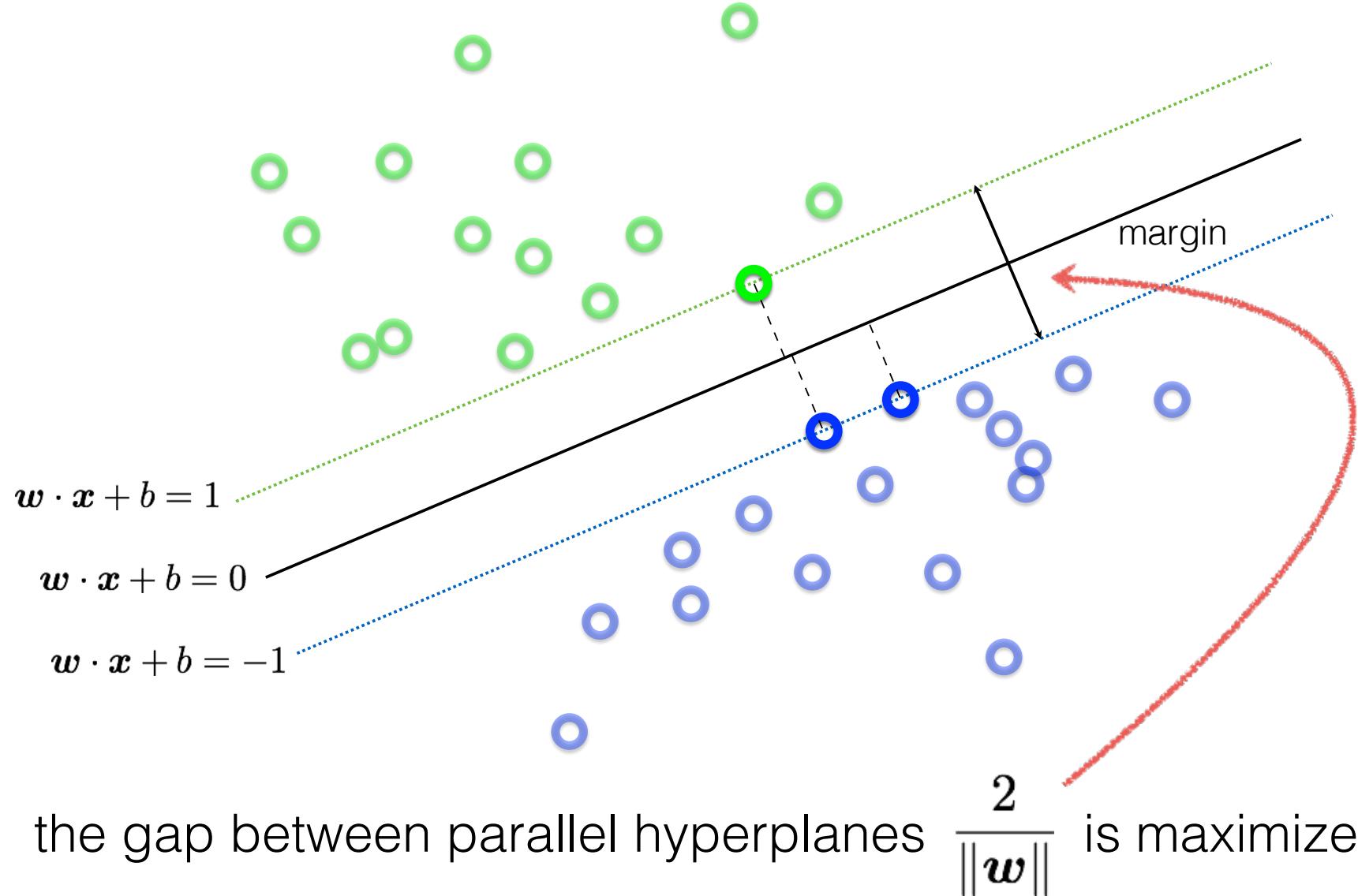
What's the best \mathbf{w} ?



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

Skipping some details.

Find hyperplane \mathbf{w} such that ...



Can be formulated as a maximization problem

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

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

What does this constraint mean?



label of the data point

Why is it +1 and -1?

Can be formulated as a maximization problem

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

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

Equivalently,

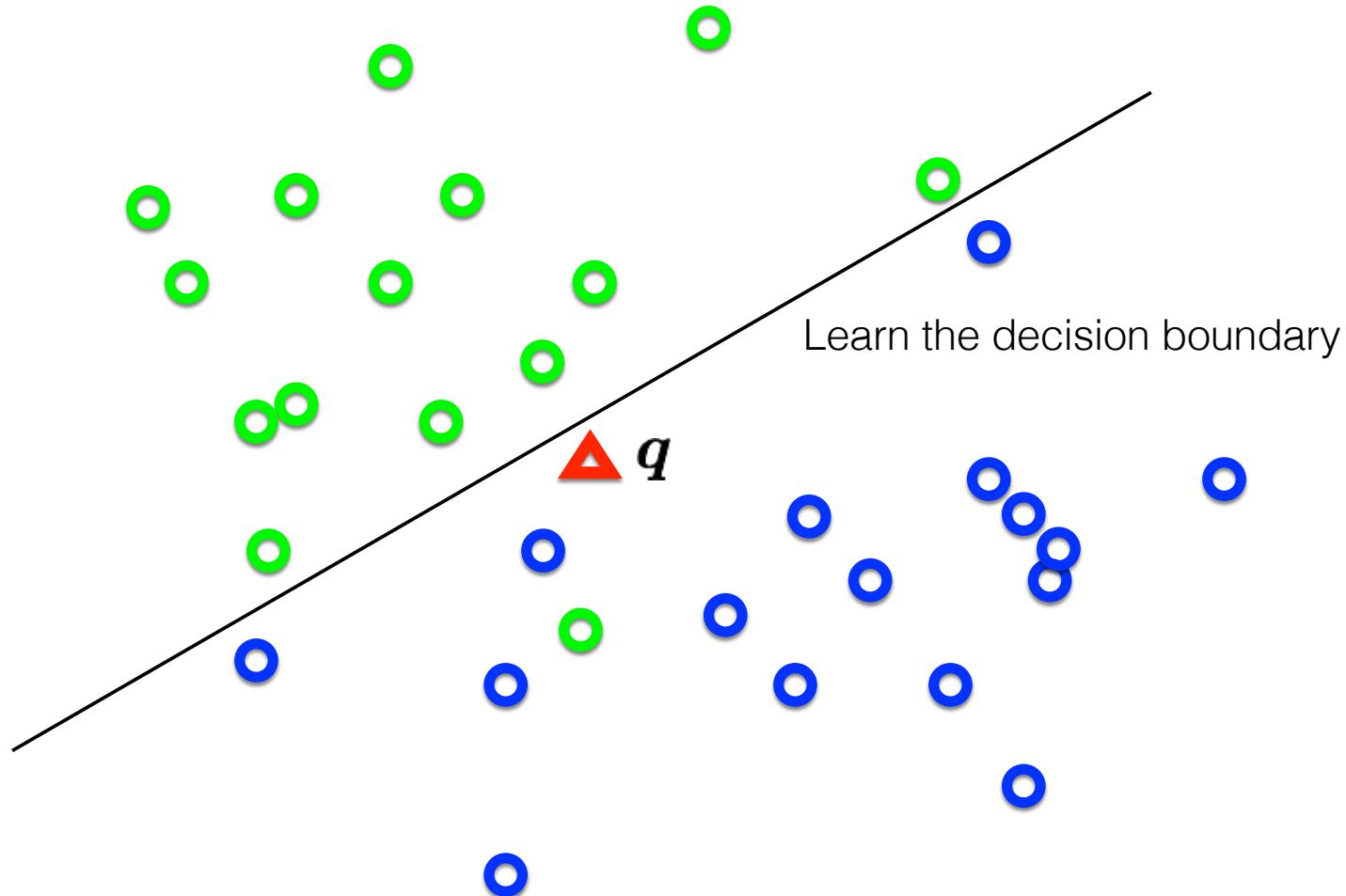
Where did the 2 go?

$$\min_{\mathbf{w}} \|\mathbf{w}\|$$

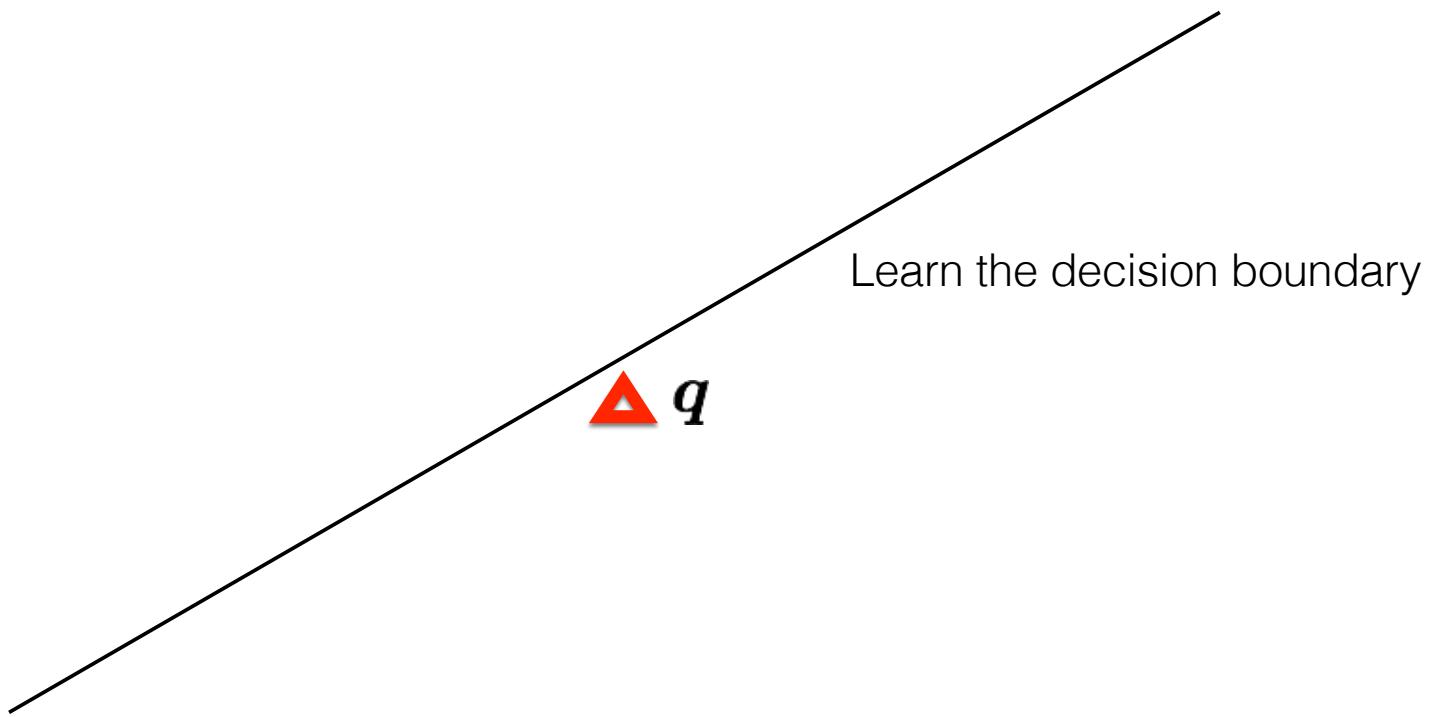
subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ for $i = 1, \dots, N$

What happened to the labels?

Distribution of data from two classes



Distribution of data from two classes



Multi-class case

Cat

Dog

Airplane

Chair

Multi-class case

Cat

Dog

Airplane

Chair

SVM

SVM

SVM

SVM

Cat



Dog

Airplane

Chair

Dog



Cat

Airplane

Chair

Airplane



Cat

Dog

Chair

Chair



Cat

Dog

Airplane

Multi-class case



0.5

SVM

Cat



Dog

Airplane

Chair

0.9

SVM

Dog



Cat

Airplane

Chair

0.1

SVM

Airplane



Cat

Dog

Chair

0.2

SVM

Chair



Cat

Dog

Airplane

References

Basic reading:

- Szeliski, Chapter 14.