# Machine Learning Overview

Slides: James Jays,
Isabelle Guyon,
Erik Sudderth,
Mark Johnson,
Derek Hoiem

1

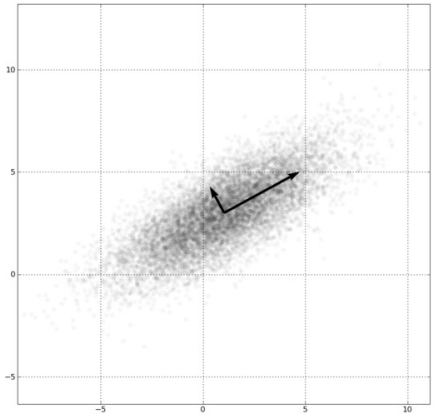
# Machine Learning Problems

| | *Supervised Learning* | *Unsupervised Learning* |
|---|---|---|
| *Discrete* | classification or categorization | clustering |
| *Continuous* | regression | dimensionality reduction |

2

# Dimensionality Reduction

- **PCA**, ICA, LLE, Isomap, *Autoencoder*
- PCA is the most important technique to know. It takes advantage of correlations in data dimensions to produce the best possible lower dimensional representation based on linear projections (minimizes reconstruction error).
- PCA should be used for dimensionality reduction, not for discovering patterns or making predictions. Don't try to assign semantic meaning to the bases.



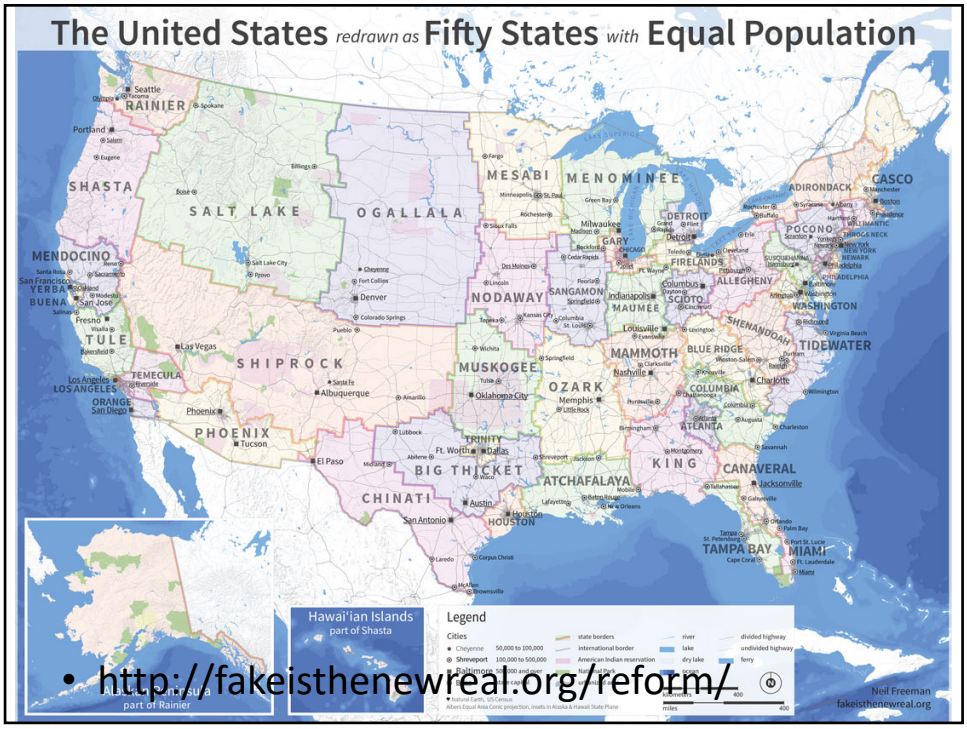

3

# Machine Learning Problems

| | *Supervised Learning* | *Unsupervised Learning* |
|---|---|---|
| *Discrete* | classification or categorization | clustering |
| *Continuous* | regression | dimensionality reduction |

4


5




- http://fakeisthenewreal.org/reform/

6

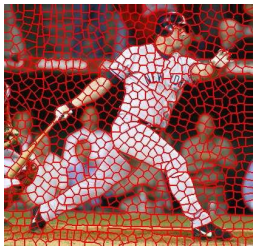
## Clustering example: image segmentation

Goal: Break up the image into meaningful or perceptually similar regions

7

## Segmentation for feature support or efficiency


[Felzenszwalb and Huttenlocher 2004]



[Shi and Malik 2001]

[Hoiem et al. 2005, Mori 2005]

Slide: Derek Hoiem

8

## Segmentation as a result

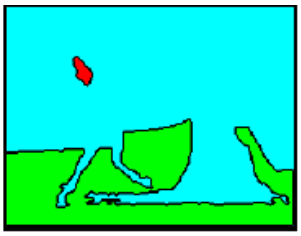



Rother et al. 2004

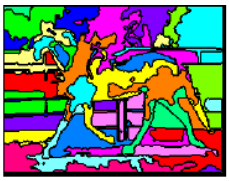
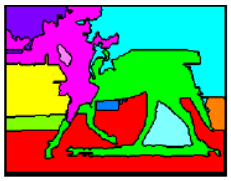
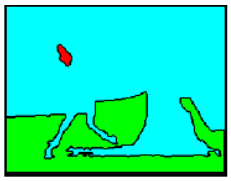
9

## Types of segmentations





Oversegmentation Undersegmentation

Multiple Segmentations

10

Clustering: group together similar points and represent them with a single token

Key Challenges:

1) What makes two points/images/patches similar?

2) How do we compute an overall grouping from pairwise similarities?

Slide: Derek Hoiem

11

## How do we cluster?

- K-means
  - Iteratively re-assign points to the nearest cluster center
- Agglomerative clustering
  - Start with each point as its own cluster and iteratively merge the closest clusters
- Mean-shift clustering
  - Estimate modes of pdf
- Spectral clustering
  - Split the nodes in a graph based on assigned links with similarity weights

12

# Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

– Preserve information

Cluster center

Data

$$\mathbf{c}^*, \boldsymbol{\delta}^* = \underset{\mathbf{c},\boldsymbol{\delta}}{\operatorname{argmin}} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij} \left(\mathbf{c}_i - \mathbf{x}_j\right)^2$$

Whether $x_j$ is assigned to $c_i$

Slide: Derek Hoiem

13

# K-means algorithm

1. Randomly select K centers

2. Assign each point to nearest center

3. Compute new center (mean) for each cluster

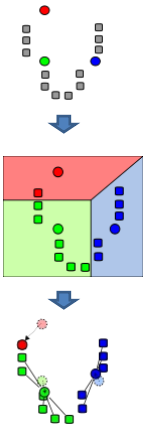


Illustration: http://en.wikipedia.org/wiki/K-means_clustering

14

# K-means algorithm

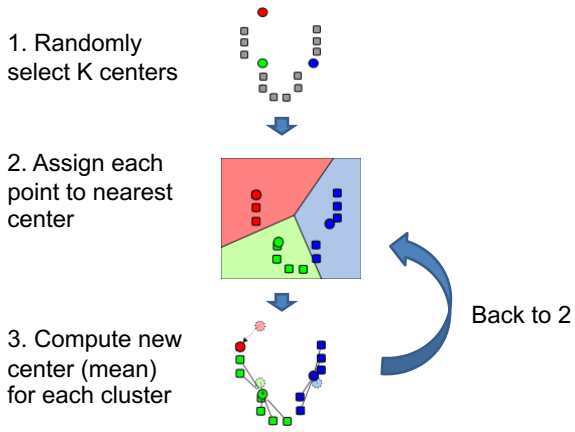



Illustration: http://en.wikipedia.org/wiki/K-means_clustering

15

# K-means

1. Initialize cluster centers: $\mathbf{c}^0$ ; t=0
2. Assign each point to the closest center

$$\boldsymbol{\delta}^t = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij} \left( \mathbf{c}_i^{t-1} - \mathbf{x}_j \right)^2$$

3. Update cluster centers as the mean of the points

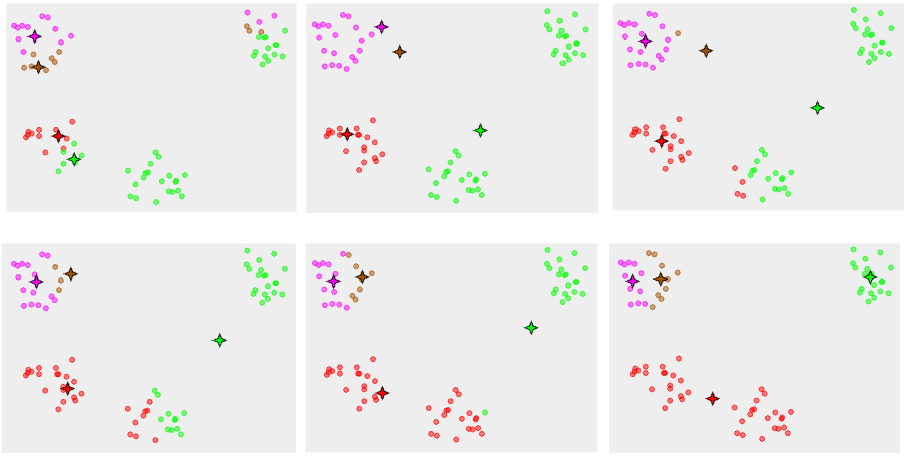
$$\mathbf{c}^t = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^t \left( \mathbf{c}_i - \mathbf{x}_j \right)^2$$

4. Repeat 2-3 until no points are re-assigned (t=t+1)

Slide: Derek Hoiem

16

## K-means converges to a local minimum



17

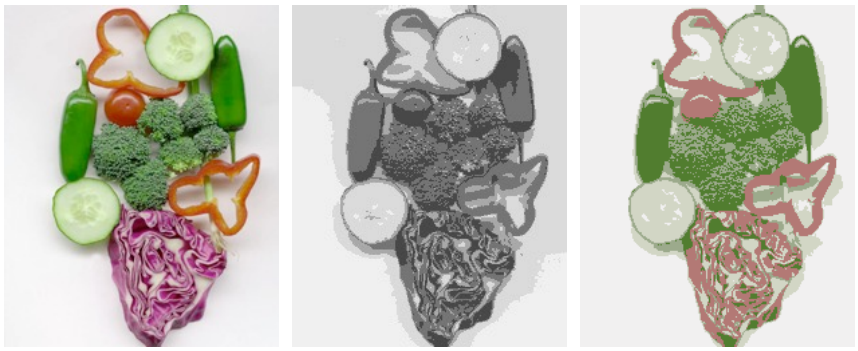
## K-means: design choices

- Initialization
  - Randomly select K points as initial cluster center
  - Or greedily choose K points to minimize residual
- Distance measures
  - Traditionally Euclidean, could be others
- Optimization
  - Will converge to a *local minimum*
  - May want to perform multiple restarts

18

## K-means clustering using intensity or color

Image | Clusters on intensity | Clusters on color



19

## How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?
- Discriminative
  - How well do the clusters correspond to labels?
    - Purity
  - Note: unsupervised clustering does not aim to be discriminative

Slide: Derek Hoiem

20

## How to choose the number of clusters?

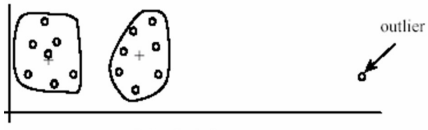
- Validation set
  - Try different numbers of clusters and look at performance
    - When building dictionaries (discussed later), more clusters typically work better

Slide: Derek Hoiem

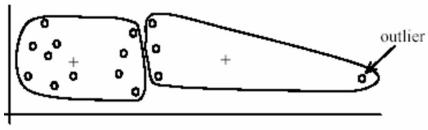
21

## K-Means pros and cons

- Pros
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast*
  - Easy to implement
- Cons
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is O(KNd) for N d-dimensional points
- Usage
  - Rarely used for pixel segmentation




(B): Ideal clusters

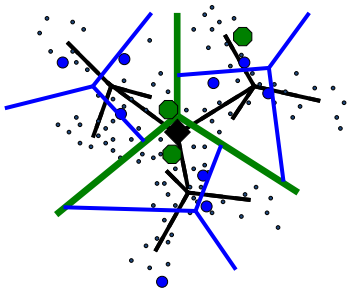



22

# Building Visual Dictionaries

1. Sample patches from a database
   - E.g., 128 dimensional SIFT vectors

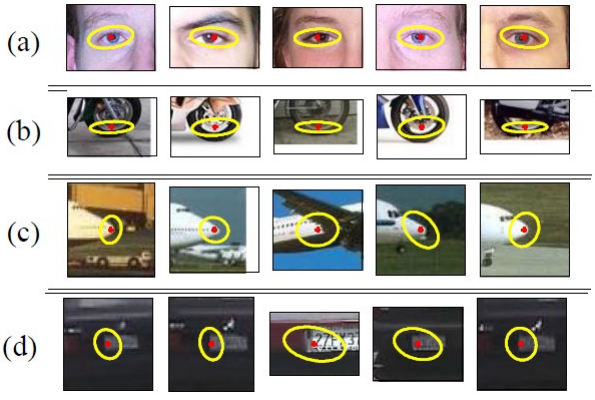
2. Cluster the patches
   - Cluster centers are the dictionary



3. Assign a codeword (number) to each new patch, according to the nearest cluster

23

# Examples of learned codewords



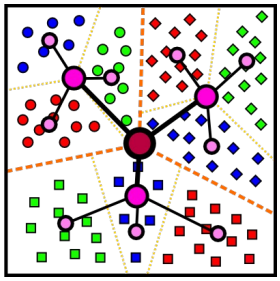

Most likely codewords for 4 learned "topics"
EM with multinomial (problem 3) to get topics

http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic05b.pdf Sivic et al. ICCV 2005

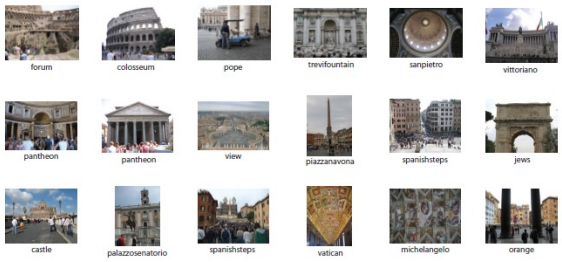
24

# Which algorithm to use?

- Quantization/Summarization: K-means
  - Aims to preserve variance of original data
  - Can easily assign new point to a cluster



Quantization for computing histograms

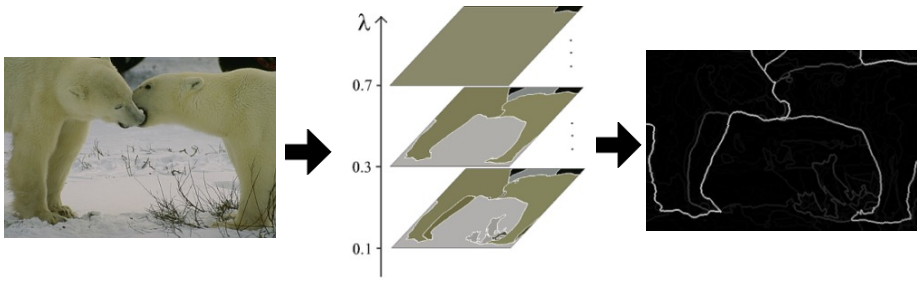



Summary of 20,000 photos of Rome using "greedy k-means"
http://grail.cs.washington.edu/projects/canonview/

25

# Which algorithm to use?

- Image segmentation: agglomerative clustering
  - More flexible with distance measures (e.g., can be based on boundary prediction)
  - Adapts better to specific data
  - Hierarchy can be useful




http://www.cs.berkeley.edu/~arbelaez/UCM.html

26

# Machine Learning Problems

| | Supervised Learning | Unsupervised Learning |
|---|---|---|
| Discrete | classification or categorization | clustering |
| Continuous | regression | dimensionality reduction |

27

# The machine learning framework

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

f(  ) = “apple”

f(  ) = “tomato”

f(  ) = “cow”

Slide credit: L. Lazebnik

28

# The machine learning framework

$$y = f(\mathbf{x})$$

output — prediction function — Image feature

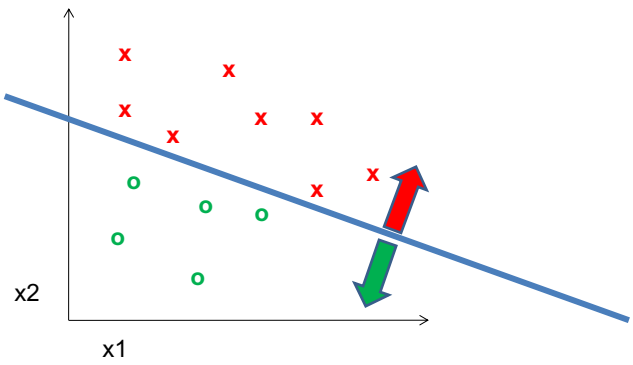
- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function $f$ by minimizing the prediction error on the training set
- **Testing:** apply $f$ to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

29

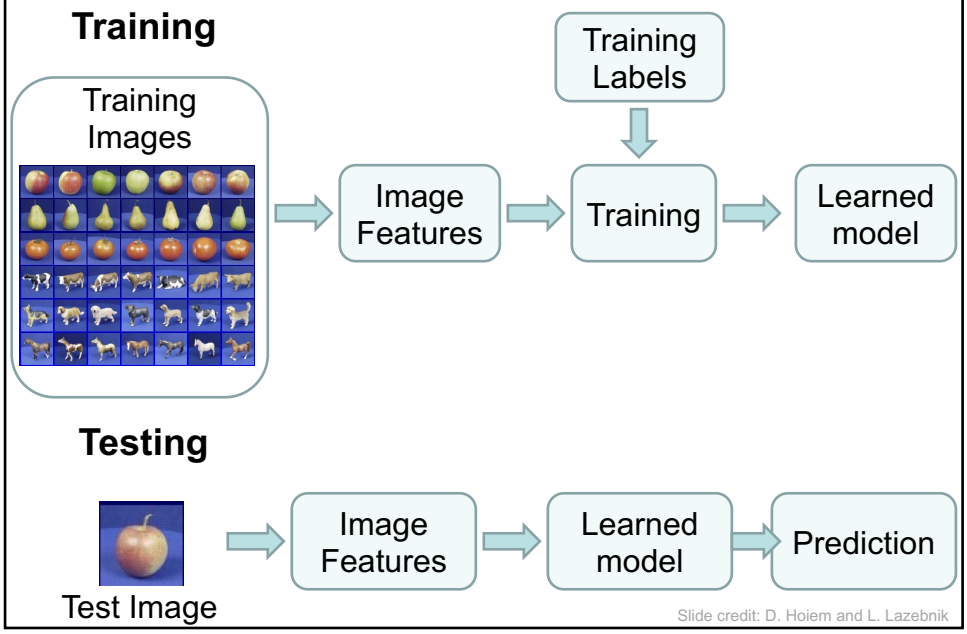
# Learning a classifier

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




30

# Steps

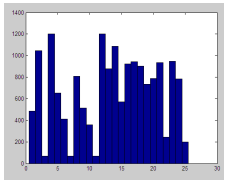


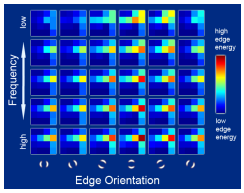

Slide credit: D. Hoiem and L. Lazebnik

31

# Features

- Raw pixels
- Histograms
- GIST descriptors
- …






Slide credit: L. Lazebnik

32

## One way to think about it…

- Training labels dictate that two examples are the same or different, in some sense
- Features and distance measures define visual similarity
- Classifiers try to learn weights or parameters for features and distance measures so that visual similarity predicts label similarity

33

## Many classifiers to choose from

- SVM
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs
- Deep Convolutional Network
- Etc.

Which is the best one?

34

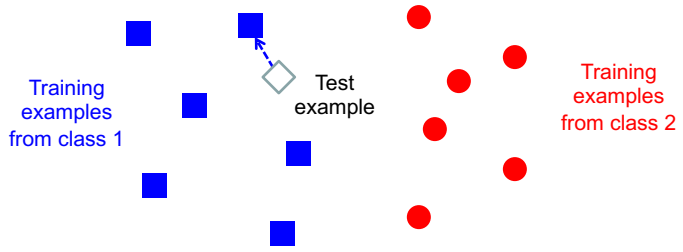
Claim:

The decision to *use* machine learning is more important than the choice of a *particular* learning method.

*Deep learning seems to be an exception to this, at the moment, probably because it is learning the feature representation.

35

# Classifiers: Nearest neighbor



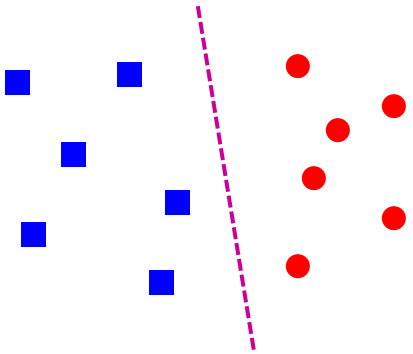

f(**x**) = label of the training example nearest to **x**

- All we need is a distance function for our inputs
- No training required!

Slide credit: L. Lazebnik

36

# Classifiers: Linear



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

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Slide credit: L. Lazebnik

37

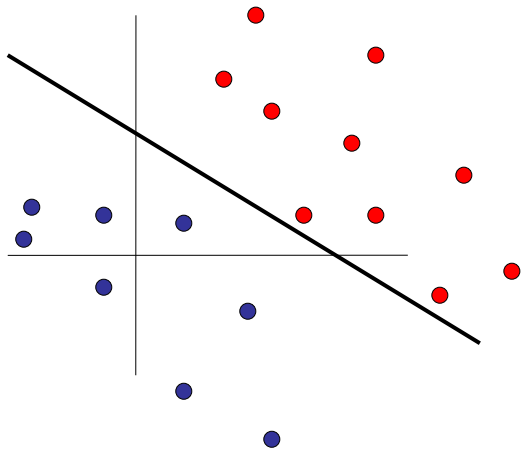
# Outline

- Classifiers
  - Generative classifiers
    - Bayesian
  - Discriminative classifiers
    - Nearest neighbors
    - Support vector machines

38

38

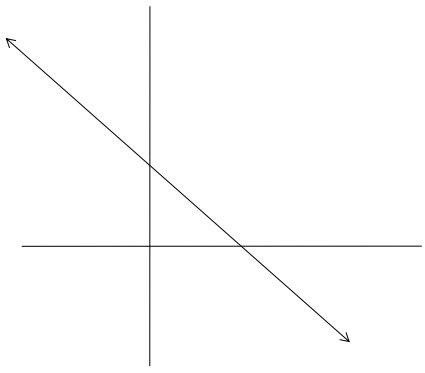
# Linear classifiers



39

39

# Lines in $R^2$



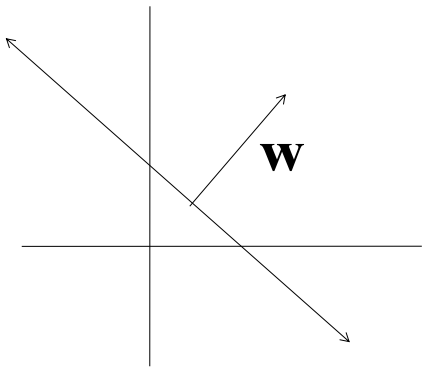
Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

40

40

# Lines in R²




Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

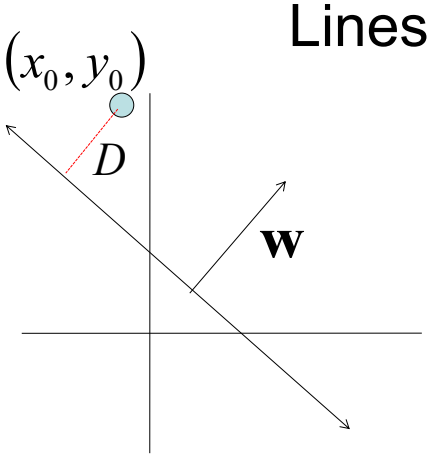
$$\updownarrow$$

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

41

41

# Lines in R²




Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

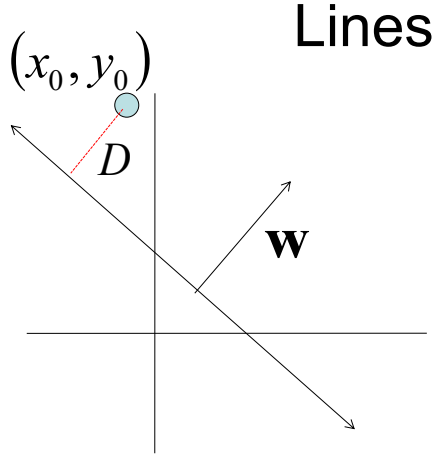
$$\updownarrow$$

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

42

42

## Lines in R²




Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

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

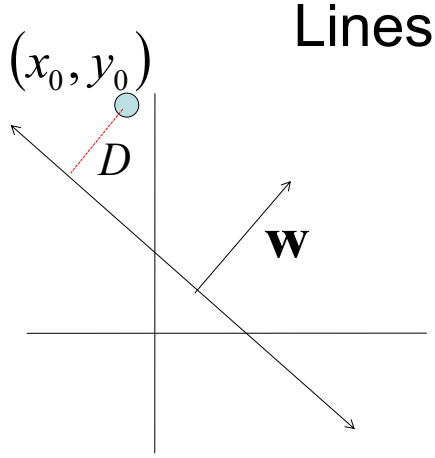
$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}}$$

distance from point to line

43



## Lines in R²




Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

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

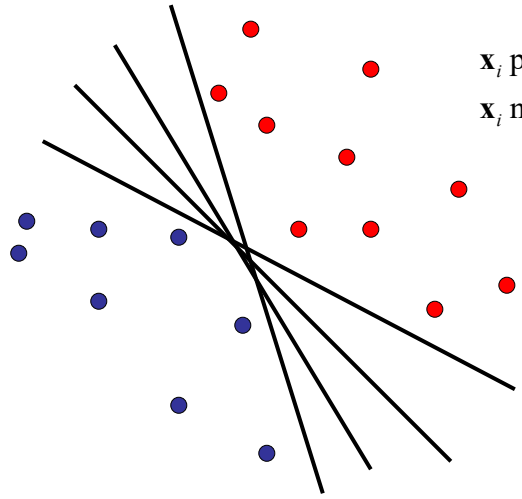
$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}} = \frac{\mathbf{w}^T\mathbf{x}_0 + b}{\|\mathbf{w}\|}$$

distance from point to line

44

# Linear classifiers

- Find linear function to separate positive and negative examples



$\mathbf{x}_i$ positive: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

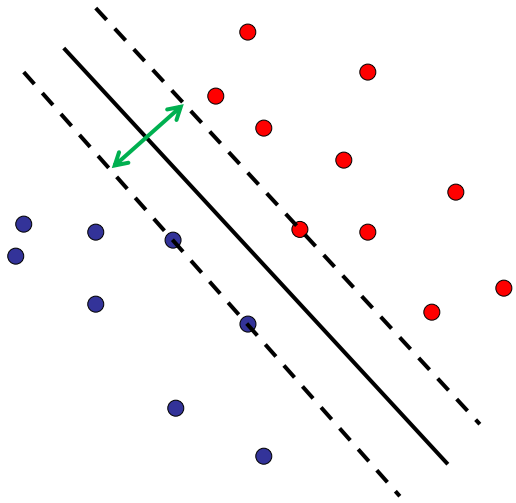
$\mathbf{x}_i$ negative: $\mathbf{x}_i \cdot \mathbf{w} + b < 0$

Which line is best?

45

45

# Support Vector Machines (SVMs)



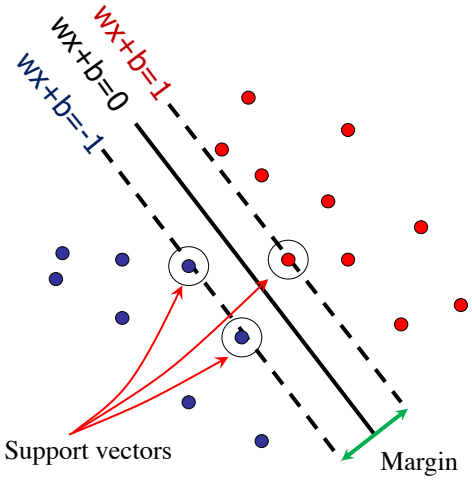
- Discriminative classifier based on *optimal separating line (for 2d case)*
- Maximize the *margin* between the positive and negative training examples

46

46

# Support vector machines

- Want line that maximizes the margin.




$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

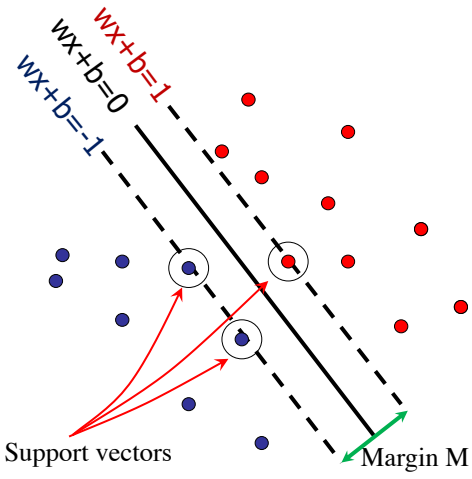
C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 1998

47

47

# Support vector machines

- Want line that maximizes the margin.




$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\frac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

For support vectors:

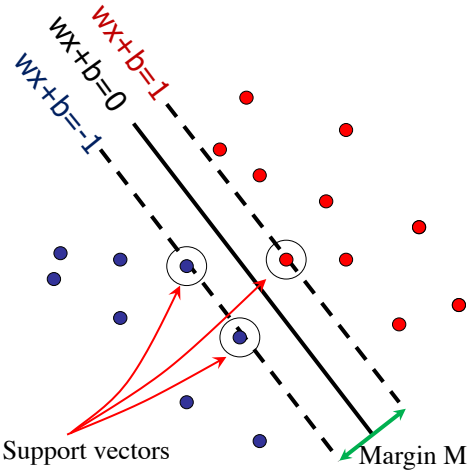
$$\frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|} \qquad M = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$$

48

48

# Support vector machines

- Want line that maximizes the margin.




$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\frac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

49

49

# Finding the maximum margin line

1. Maximize margin $2/\|\mathbf{w}\|$
2. Correctly classify all training data points:

$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

- *Quadratic optimization problem*:
- Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w} \cdot \boldsymbol{x}_i + b) \geq 1$

50

50

# Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

learned weight — Support vector

- Based on the "dual form" of a SVM, which is a Lagrangian formulation

$$L_P \equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \qquad \sum_i \alpha_i y_i = 0.$$

51

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,

51

# Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

$b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ (for any support vector)

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Classification function:

$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

$$= \text{sign}\left(\sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b\right)$$

*If f(x) < 0, classify as negative, if f(x) > 0, classify as positive*

52

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,

52

# Questions

- **What if the features are not 2d?**
- What if the data is not linearly separable?
- What if we have more than just two categories?

53

53

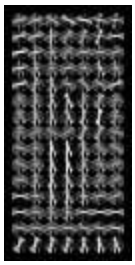
# Questions

- **What if the features are not 2d?**
  - Generalizes to d-dimensions – replace line with "hyperplane"
- What if the data is not linearly separable?
- What if we have more than just two categories?

54

54

# Person detection with HoG's & linear SVM's



- Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Dalal & Triggs, CVPR 2005

Code available:
http://pascal.inrialpes.fr/soft/olt/

55

55

# Person detection with HoG's & linear SVM's


- Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005
- http://lear.inrialpes.fr/pubs/2005/DT05/

56

56

# Questions

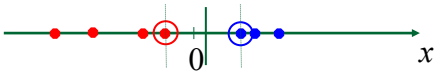
- What if the features are not 2d?
- **What if the data is not linearly separable?**
- What if we have more than just two categories?

57

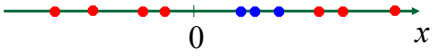
57

# Non-linear SVMs

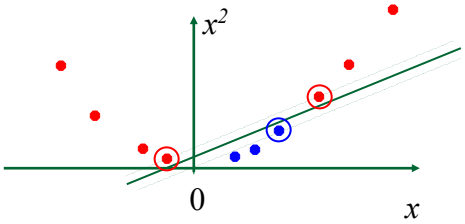
- Datasets that are linearly separable with some noise work out great:




- But what are we going to do if the dataset is just too hard?




- How about… mapping data to a higher-dimensional space:

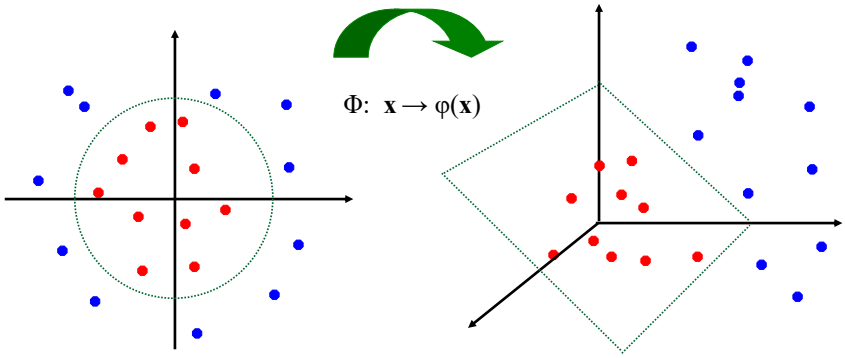



58

58

# Non-linear SVMs: feature spaces

- General idea: the original input space can be mapped to some higher-dimensional feature space where the training set is separable:




59

Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

59

# Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\boldsymbol{\varphi}(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

60

60

# Examples of kernel functions

$$K(x_i, x_j) = x_i^T x_j$$

- Linear:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

- Gaussian RBF:

- Histogram intersection:

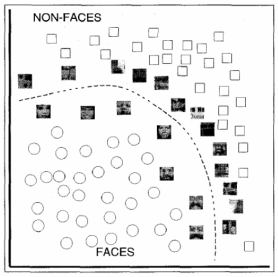
$$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

61

61

# SVMs for recognition

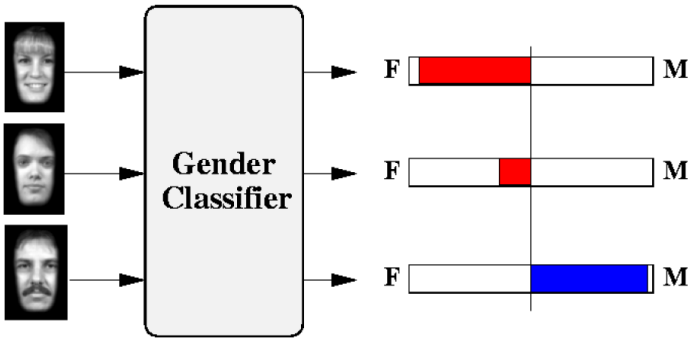
1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Use this "kernel matrix" to solve for SVM support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.




62

62

# Example: learning gender with SVMs

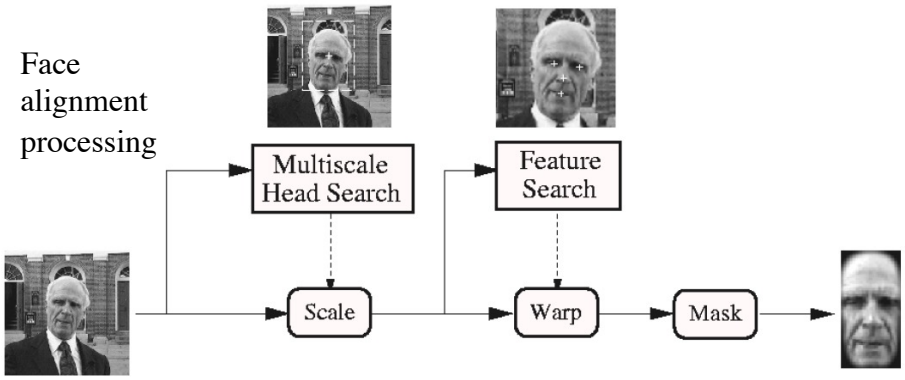



Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Moghaddam and Yang, Face & Gesture 2000.

63

63




Processed faces

64

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

64

# Learning gender with SVMs

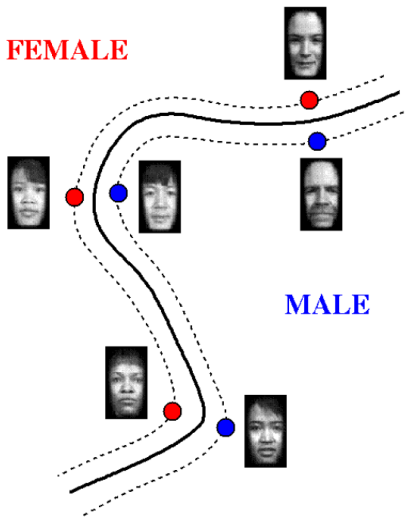
- Training examples:
  - 1044 males
  - 713 females
- Experiment with various kernels, select Gaussian RBF

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sigma^2})$$

65

65

# Support Faces




66

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

66

# Classifier Performance

| Classifier | Error Rate | | |
|---|---|---|---|
| | Overall | Male | Female |
| **SVM with RBF kernel** | **3.38%** | **2.05%** | **4.79%** |
| **SVM with cubic polynomial kernel** | **4.88%** | **4.21%** | **5.59%** |
| Large Ensemble of RBF | 5.54% | 4.59% | 6.55% |
| Classical RBF | 7.79% | 6.89% | 8.75% |
| Quadratic classifier | 10.63% | 9.44% | 11.88% |
| Fisher linear discriminant | 13.03% | 12.31% | 13.78% |
| Nearest neighbor | 27.16% | 26.53% | 28.04% |
| Linear classifier | 58.95% | 58.47% | 59.45% |

67

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

67

# Gender perception experiment: How well can humans do?

- Subjects:
  - 30 people (22 male, 8 female)
  - Ages mid-20's to mid-40's
- Test data:
  - 254 face images
  - Low res (6 males, 4 females)
  - High res versions
- Task:
  - Classify as male or female, forced choice
  - No time limit

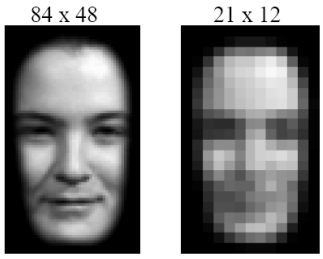
68

Moghaddam and Yang, Face & Gesture 2000.

68

# Gender perception experiment: How well can humans do?

Stimuli →




Results →

| High-Res | Low-Res |
|---|---|
| 6.54% Error | 30.7% Error |

$\sigma$ = 3.7%

69

Moghaddam and Yang, Face & Gesture 2000.

69

# Human vs. Machine

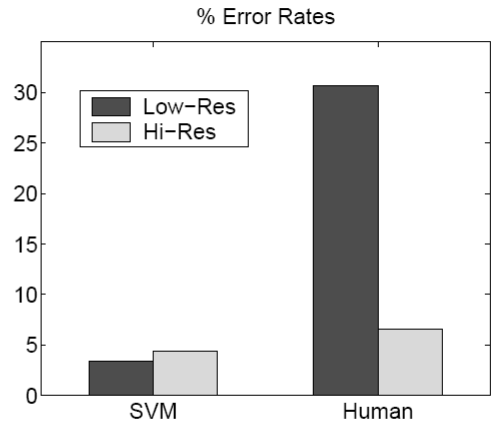


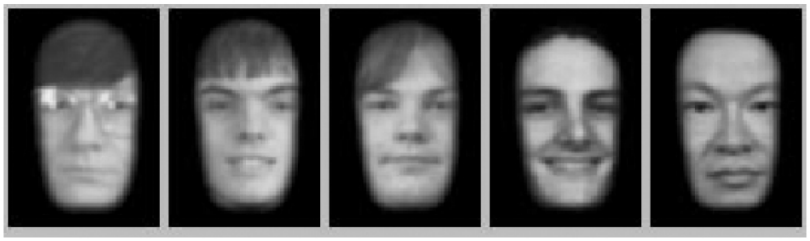

Figure 6. SVM vs. Human performance

- SVMs performed better than any single human test subject, at either resolution

70

70

# Hardest examples for humans



Top five human misclassifications

71

Moghaddam and Yang, Face & Gesture 2000.

71

# Questions

- What if the features are not 2d?
- What if the data is not linearly separable?
- **What if we have more than just two categories?**

72

72

# Multi-class SVMs

- Achieve multi-class classifier by combining a number of binary classifiers
- **One vs. all**
  - Training: learn an SVM for each class vs. the rest
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- **One vs. one**
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

73

73

# SVMs: Pros and cons

- Pros
  - Many publicly available SVM packages: http://www.kernel-machines.org/software
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - Kernel-based framework is very powerful, flexible
  - Often a sparse set of support vectors – compact at test time
  - Work very well in practice, even with very small training sample sizes
- Cons
  - No "direct" multi-class SVM, must combine two-class SVMs
  - Can be tricky to select best kernel function for a problem

74

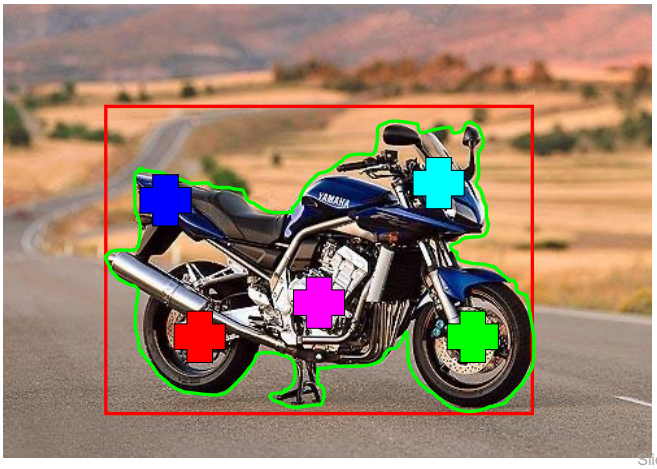
Adapted from Lana Lazebnik

74

# Recognition task and supervision

- Images in the training set must be annotated with the "correct answer" that the model is expected to produce

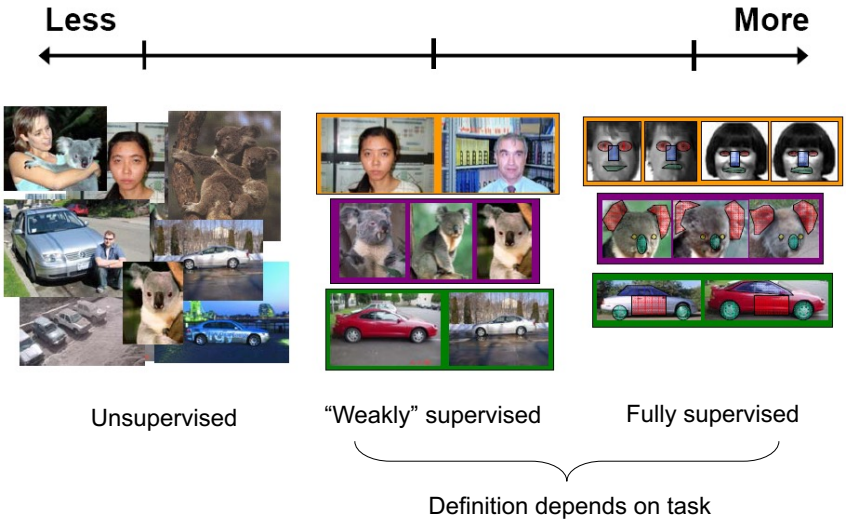
Contains a motorbike



Slide credit: L. Lazebnik

75

# Spectrum of supervision




Slide credit: L. Lazebnik

76

# Generalization

Training set (labels known)

Test set (labels unknown)

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

Slide credit: L. Lazebnik

77

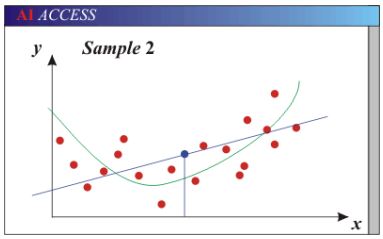
# Generalization

- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model.
  - **Variance:** how much models estimated from different training sets differ from each other.
- **Underfitting:** model is too "simple" to represent all the relevant class characteristics
  - High bias (few degrees of freedom) and low variance
  - High training error and high test error
- **Overfitting:** model is too "complex" and fits irrelevant characteristics (noise) in the data
  - Low bias (many degrees of freedom) and high variance
  - Low training error and high test error

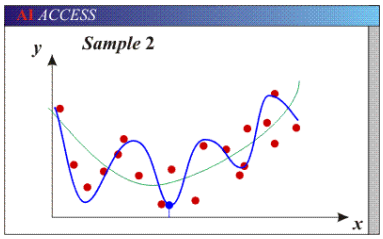
Slide credit: L. Lazebnik

78

# Bias-Variance Trade-off




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




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

Slide credit: D. Hoiem

79

# Bias-Variance Trade-off

$$E(MSE) = noise^2 + bias^2 + variance$$

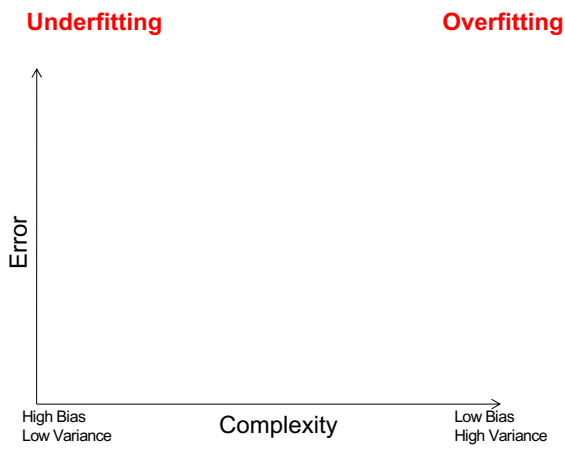
- Unavoidable error
- Error due to incorrect assumptions
- Error due to variance of training samples

See the following for explanations of bias-variance (also Bishop's "Neural Networks" book):
•http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf

Slide credit: D. Hoiem

80

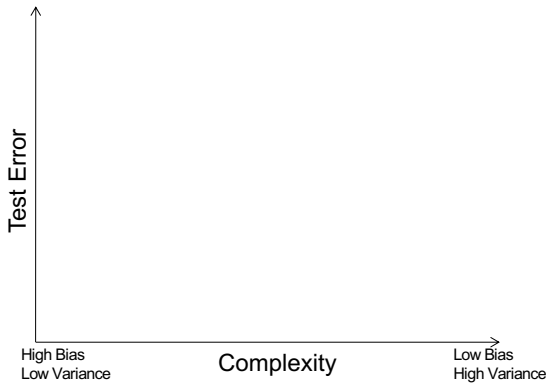
# Bias-variance tradeoff




Slide credit: D. Hoiem

81

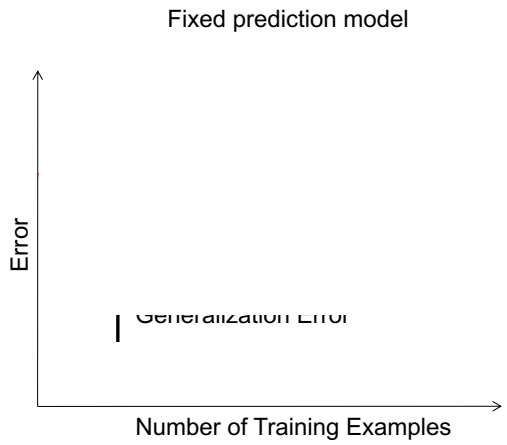
# Bias-variance tradeoff




Slide credit: D. Hoiem

82

# Effect of Training Size




Slide credit: D. Hoiem

83

# Remember…

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


- Three kinds of error
  - Inherent: unavoidable
  - Bias: due to over-simplifications
  - Variance: due to inability to perfectly estimate parameters from limited data

Slide credit: D. Hoiem

84

## How to reduce variance?

- Choose a simpler classifier
- Regularize the parameters
- Get more training data

Slide credit: D. Hoiem

85