All models are wrong, some models are useful.

- George Box, Statistician

CSE455 & CSE 552
Machine Learning

Spring 2025 Semester

SVM

1

Supervised Machine Learning

of experiences | observed outcome | observed input

$$\arg\max_{f \in F} \sum_{i=1}^{n} g(y_i, f(x_i))$$

hypothesis/program space | goodness/performance measure | selected hypothesis

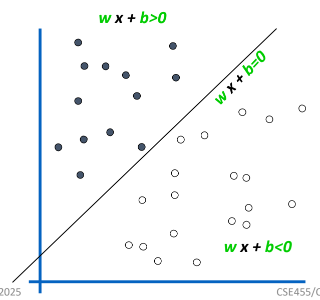
2

Linear Classifiers

- denotes +1
- denotes -1

$\boldsymbol{x} \rightarrow \boxed{f} \rightarrow y^{est}$ (input α)

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



How would you classify this data?

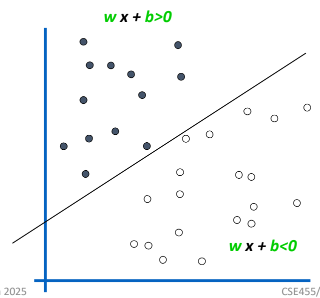
3

Linear Classifiers

- denotes +1
- denotes -1

$\boldsymbol{x} \rightarrow \boxed{f} \rightarrow y^{est}$ (input α)

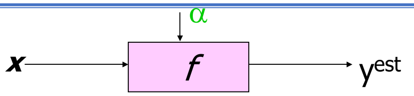
$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}\ \boldsymbol{x} + b)$



How would you classify this data?

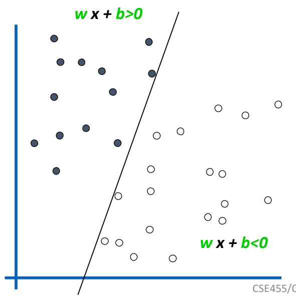
4

Linear Classifiers



● denotes +1
○ denotes -1

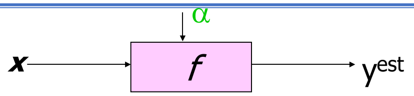
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$



How would you classify this data?

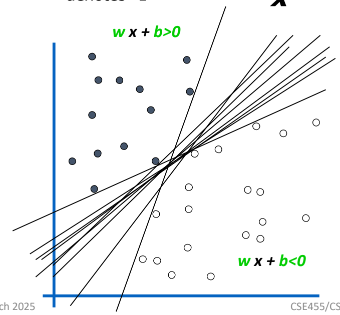
5

Linear Classifiers



● denotes +1
○ denotes -1

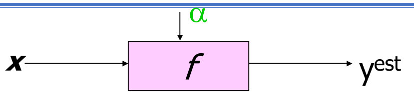
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$



How would you classify this data?

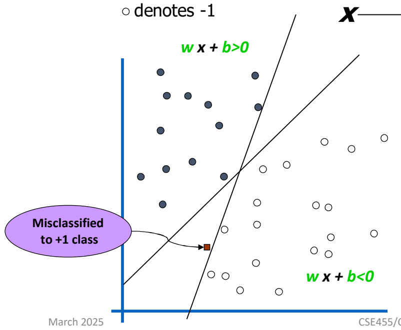
6

Linear Classifiers



● denotes +1
○ denotes -1

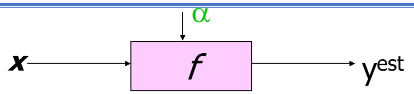
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$



How would you classify this data?

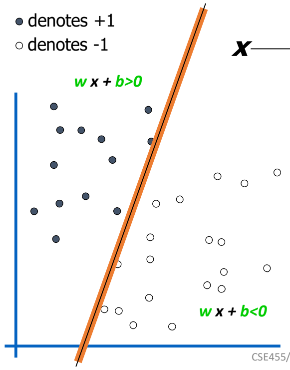
7

Linear Classifiers



● denotes +1
○ denotes -1

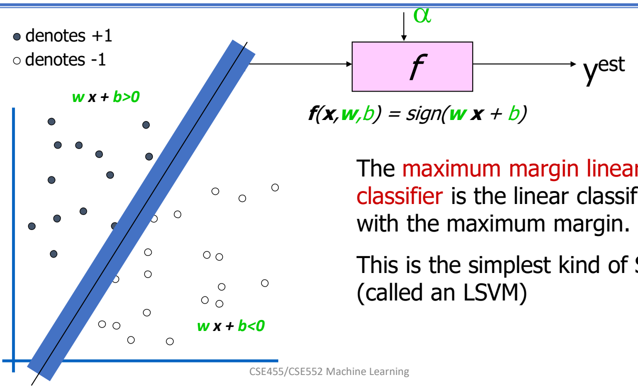
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$



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

8

Linear Classifiers

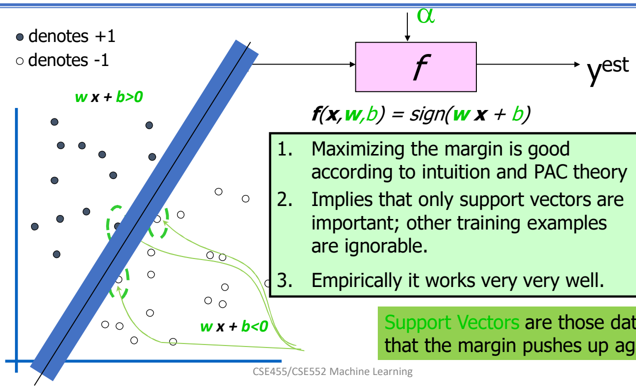


The maximum margin linear classifier is the linear classifier with the maximum margin.

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

9

Linear Classifiers

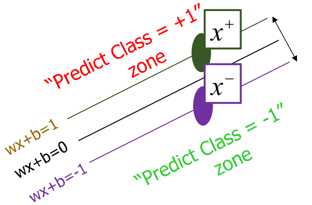


1. Maximizing the margin is good according to intuition and PAC theory
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

Support Vectors are those datapoints that the margin pushes up against

10

Linear SVM Mathematically



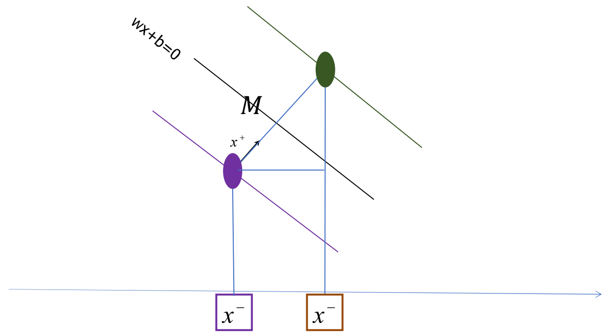
M = Margin Width

$$w \cdot x^+ + b = +1$$
$$w \cdot x^- + b = -1$$
$$w \cdot (x^+ - x^-) = 2$$

$$M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|}$$

11

You Doubt?



12

Linear SVM Mathematically

- Goal: **1) Correctly classify all training data**

$$wx_i + b \geq 1 \quad \textit{if } y_i = +1$$
$$wx_i + b \leq -1 \quad \textit{if } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$

2) Maximize the Margin $M = \frac{2}{|w|}$

same as minimize $\frac{1}{2} w^t w$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**
- Minimize $\Phi(w) = \frac{1}{2} w^t w$ subject to $y_i(wx_i + b) \geq 1 \quad \forall i$

13

Solving the Optimization Problem

Find **w** and b such that
$\mathbf{\Phi(w)} = \frac{1}{2} \mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- Need to optimize a quadratic function subject to linear constraints
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them
- The solution involves constructing a dual problem where a Lagrange multiplier α_i is associated with every constraint in the primary problem:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

14

Optimization Problem Solution

- The solution has the form:

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^T\mathbf{x_k}$ for any $\mathbf{x_k}$ such that $\alpha_k \neq 0$

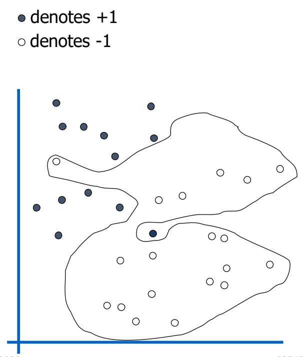
- Each non-zero αi indicates that corresponding xi is a support vector
- Then the classifying function will have the form:

$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$

- Notice that it relies on an inner product between the test point x and the support vectors $\mathbf{x_i}$ – we will return to this later
- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x_i}^T\mathbf{x}$ between all pairs of training points

15

Dataset with Noise



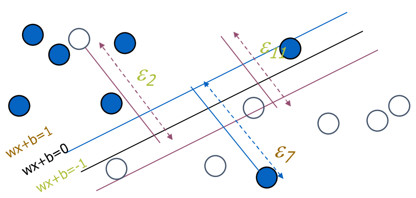
- Hard Margin: So far we require all data points be classified correctly
  - No training error
- What if the training set is noisy?
  - Solution 1: use very powerful kernels

OVERFITTING!

16

Soft Margin Classification

Slack variables ε_i **can be added to allow misclassification of difficult or noisy examples**



What should our quadratic optimization criterion be?

Minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

17

Hard Margin vs Soft Margin

- The old formulation:

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- The new formulation incorporating slack variables:

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

- Parameter C can be viewed as a way to control over fitting

18

Linear SVMs: Overview

- The classifier is a separating hyper-plane
- Most “important” training points are support vectors; defining the hyper-plane
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i
- Both in the dual formulation of the problem and in the solution training points appear only inside dot products:

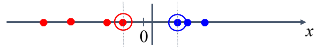
Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x_i}^T\mathbf{x_j}}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

$f(\mathbf{x}) = \Sigma\alpha_i y_i \boxed{\mathbf{x_i}^T\mathbf{x}} + b$

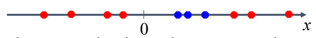
19

Non-linear SVMs

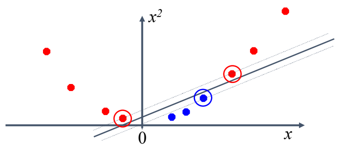
- Datasets that are linearly separable with some noise work out great:



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



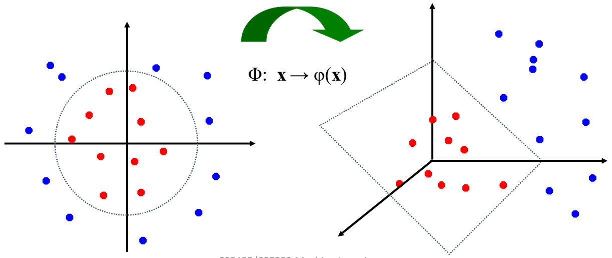
- How about… mapping data to a higher-dimensional space:



20

Non-linear SVMs: Feature spaces

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



21

The "Kernel Trick"

- The linear classifier relies on dot product between vectors

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

- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \phi(x)$, the dot product becomes:

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

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space

22

The "Kernel Trick"

- Example:
  - 2-dimensional vectors x=[x_1 x_2];
  - Let $K(x_i,x_j)=(1 + x_i^T x_j)^2$
  - Need to show that $K(x_i,x_j)= \phi(x_i)^T\phi(x_j)$:

$K(x_i,x_j)$

$= (1 + x_i^T x_j)^2,$

$= 1+ x_{i1}^2 x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$

$= [1\ \ x_{i1}^2\ \ \surd 2\, x_{i1}x_{i2}\ \ x_{i2}^2\ \ \surd 2 x_{i1}\ \ \surd 2 x_{i2}]^T$

$[1\ \ x_{j1}^2\ \ \surd 2\, x_{j1}x_{j2}\ \ x_{j2}^2\ \ \surd 2 x_{j1}\ \ \surd 2 x_{j2}]$

$= \phi(x_i)^T\phi(x_j),$

where $\phi(x) = [1\ \ x_1^2\ \ \surd 2\, x_1 x_2\ \ x_2^2\ \ \surd 2 x_1\ \ \surd 2 x_2]$

23

What Functions are Kernels?

- For some functions $K(x_i,x_j)$ checking that

  $K(x_i,x_j)= \phi(x_i)^T\phi(x_j)$ can be cumbersome.
- **Mercer's theorem**:

  Every semi-positive definite symmetric function is a kernel
- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric **Gram** matrix:

K=

$K(\mathbf{x_1},\mathbf{x_1})$	$K(\mathbf{x_1},\mathbf{x_2})$	$K(\mathbf{x_1},\mathbf{x_3})$	…	$K(\mathbf{x_1},\mathbf{x_N})$
$K(\mathbf{x_2},\mathbf{x_1})$	$K(\mathbf{x_2},\mathbf{x_2})$	$K(\mathbf{x_2},\mathbf{x_3})$		$K(\mathbf{x_2},\mathbf{x_N})$
…	…	…	…	…
$K(\mathbf{x_N},\mathbf{x_1})$	$K(\mathbf{x_N},\mathbf{x_2})$	$K(\mathbf{x_N},\mathbf{x_3})$	…	$K(\mathbf{x_N},\mathbf{x_N})$

24

Examples of Kernel Functions

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
- Polynomial of power p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
- Gaussian (radial-basis function network):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x}_i,\mathbf{x}_j)= \tanh(\beta_0\mathbf{x}_i^T\mathbf{x}_j + \beta_1)$

Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1\ldots\alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$$\boxed{f(\mathbf{x}) = \Sigma\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b}$$

- Optimization techniques for finding α_i's remain the same!

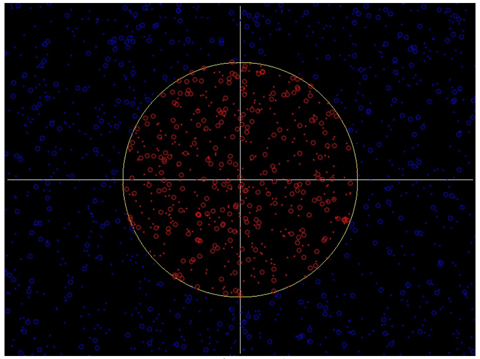
Nonlinear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space
- It does not need to represent the space explicitly, simply by defining a kernel function
- The kernel function plays the role of the dot product in the feature space.

Practical SVM

- OpenCV has a good implementation
- Kernel functions:
  - Polynomial: $(c+d*x)^n$

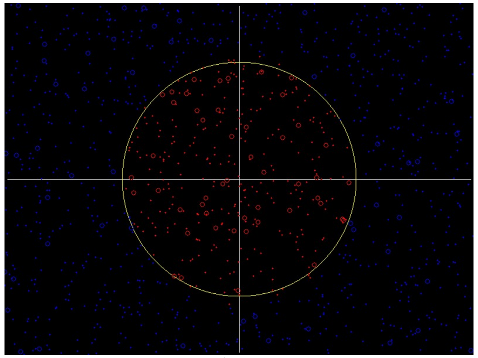
Non-linear SVM and Data



N = 500

29

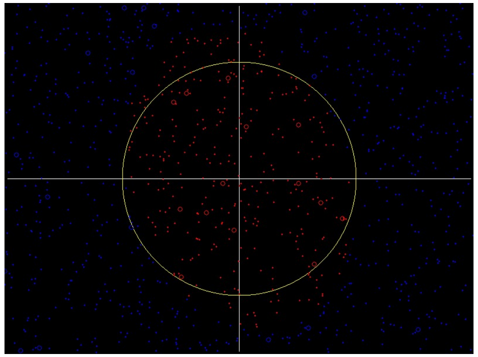
Non-linear SVM and Data



N = 100

30

Non-linear SVM and Data



N = 20

31

Properties of SVM

- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
  - only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
  - complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection

32

SVM Applications

- SVM has been used successfully in many real-world problems
  - text (and hypertext) categorization
  - image classification
  - bioinformatics (Protein classification, Cancer classification)
  - hand-written character recognition

Application 1: Cancer Classification

- High Dimensional
  - **p>1000; n<100**
- Imbalanced
  - **less positive samples**

$$K[x,x] = k(x,x) + \lambda \frac{n^+}{N}$$

- Many irrelevant features
- Noisy

Patients /Genes	**$Gene_1$**	**$Gene_2$**	**…**	**$Gene_p$**
$Patient_1$				
$Patient_2$				
…				
$Patient_n$				

FEATURE SELECTION

In the linear case, w_i^2 gives the ranking of dim i

SVM is sensitive to noisy (mis-labeled) data ☹

Weakness of SVM

- It is sensitive to noise
  - A relatively small number of mislabeled examples can dramatically decrease the performance
- It only considers two classes
  - How to do multi-class classification with SVM?
  - One vs all
    - Step 1: With output arity m, learn m SVM's
      - SVM 1 learns "Output==1" vs "Output != 1"
      - SVM 2 learns "Output==2" vs "Output != 2"
      - :
      - SVM m learns "Output==m" vs "Output != m"
    - Step 2: To predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

Application 2: Text Categorization

- Task: The classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content
  – email filtering, web searching, sorting documents by topic, etc..
- A document can be assigned to more than one category, so this can be viewed as a series of binary classification problems, one for each category

Representation of Text

- IR's vector space model (aka bag-of-words representation)
- A doc is represented by a vector indexed by a pre-fixed set or dictionary of terms
- Values of an entry can be binary or weights

$$\phi_i(x) = \frac{\mathrm{tf}_i \log(\mathrm{idf}_i)}{\kappa},$$

- Normalization, stop words, word stems
- $Doc\ \boldsymbol{x} \rightarrow \Phi(\boldsymbol{x})$

Text Categorization using SVM

- The distance between two documents is $\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{z})$
- $K(\boldsymbol{x}, \boldsymbol{z}) = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{z})$ is a valid kernel, SVM can be used with $K(\boldsymbol{x}, \boldsymbol{z})$ for discrimination
- Why SVM?
  - High dimensional input space
  - Few irrelevant features (dense concept)
  - Sparse document vectors (sparse instances)
  - Text categorization problems are linearly separable

Some Issues

- Choice of kernel
  - Gaussian or polynomial kernel is default
  - If ineffective, more elaborate kernels are needed
  - Domain experts can give assistance in formulating appropriate similarity measures
- Choice of kernel parameters
  - e.g. σ in Gaussian kernel
  - σ is the distance between closest points with different classifications
  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters
- Optimization criterion – Hard margin vs Soft margin
  - a lengthy series of experiments in which various parameters are tested

SVM is Good for Some Problems

Goal: Image classification of tanks. Autofire when an enemy tank is spotted.

Input data: Photos of own and enemy tanks.

Worked really good with the training set used. In reality it failed completely.

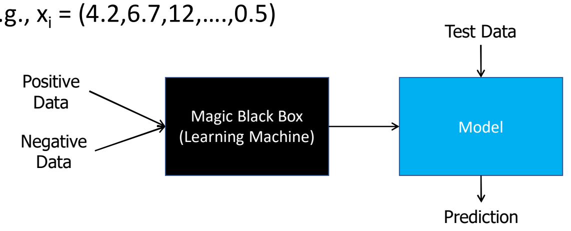
Reason: All enemy tank photos taken in the morning. All own tanks in dawn. The classifier could recognize dusk from dawn!!!!

Assessing and Comparing Classification Algorithms

41

Input Encoding

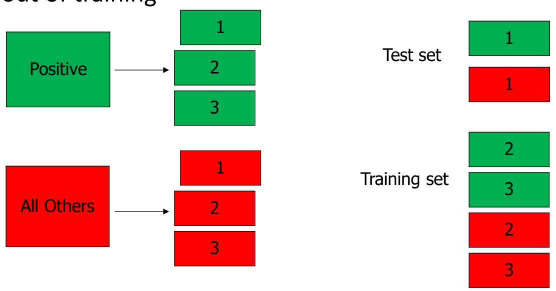
- Prediction / two class classification
  - Label positive data as +1 and all others as –1
  - The input vector x_i represents the input data as a vector of features
- E.g., $x_i = (4.2, 6.7, 12, \ldots, 0.5)$



42

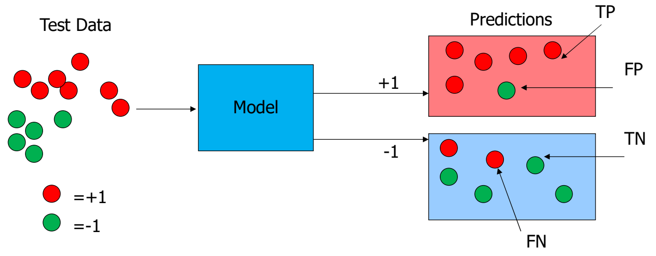
Cross-validation

Cross validation: Split the data into n sets, train on n-1 set, test on the set left out of training



43

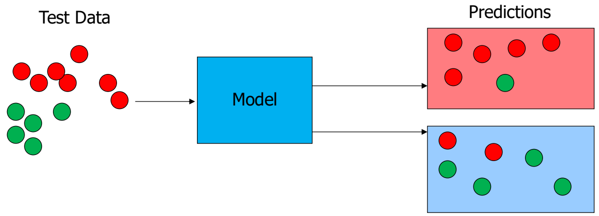
Performance Measurements



Precision = TP /(TP+FP), the fraction of predicted +1 that actually are +1.
Recall = TP /(TP+FN), the fraction of the +1 that actually are predicted as +1.

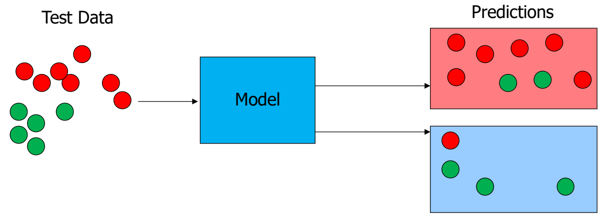
44

Performance Measurements



45

Performance Measurements



46

Precision and Recall

	actual class (observation)	
predicted class (expectation)	tp (true positive) Correct result	fp (false positive) Unexpected result
	fn (false negative) Missing result	tn (true negative) Correct absence of result

Precision = $\frac{tp}{tp+f}$

True negative rate = $\frac{tp}{tp+fp}$

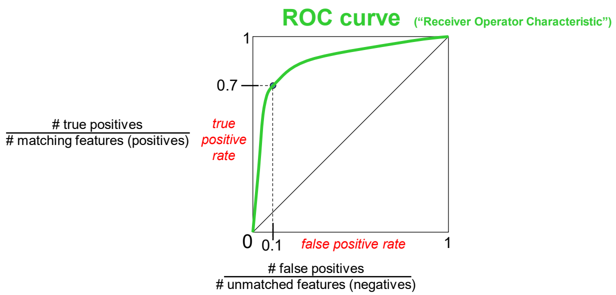
Recall = $\frac{tp}{tp+f}$

Accuracy = $\frac{tp+tn}{tp+tn+fp+f}$

47

Evaluating the Results

As discussed earlier, ROC can be used...



48

ROC Curves

- ROC Curves
  - Generated by counting # current/incorrect matches, for different threholds
  - Want to maximize area under the curve (AUC)
  - Useful for comparing different feature matching methods
  - For more info: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

49

Confusion Matrix

		Predicted Class			
		A	B	C	D
Actual Class	A	.9	.1	.0	.0
	B	.1	.8	.1	.0
	C	.0	.1	.7	.2
	D	.0	.0	.2	.8

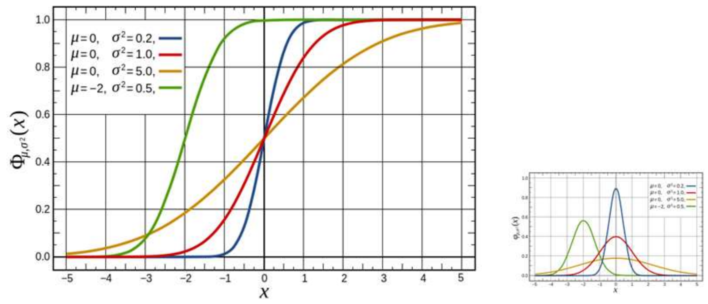
50

Cumulative Distribution Function

- In probability theory and statistics
  - CDF (or distribution function), describes the probability that a real-valued random variable X with a given probability distribution will be found at a value less than or equal to x.
  - Intuitively, it is the "area so far" function of the probability distribution.
- Cumulative distribution functions are also used to specify the distribution of multivariate random variables.

51

CDF



Cumulative distribution function for the normal distributions.

52

Thanks for listening!

53