ISTA 421 + INFO 521
Introduction to Machine Learning

Lecture 21:
Nearest Neighbors, Classifier Evaluation

Clay Morrison
claytonm@email.arizona.edu
Gould-Simpson 819
Phone 621-6609

8 November 2017 1

Non-Probabilistic Classifiers

2

Non-probabilistic Classifier: KNN

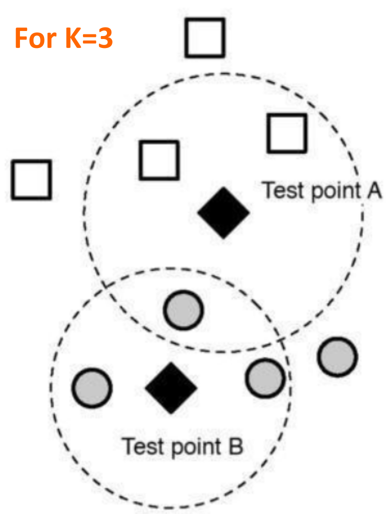
- **K-nearest neighbors** (KNN)
- Very popular because very simple *and* excellent empirical performance
- Handles both binary and multi-class data
- Makes no assumptions about the parametric form of the decision boundary:
  - A **non-parametric** method
- **Does not have a training phase** – just store the training data and do computation when time to classify

3

KNN Classification

- Find the K "training points" that are closest to x_{new} .
- Select the **majority** class amongst these K neighbors

(or for regression: **average**)



4

KNN Classification

- Can use any **distance metric**

$d : X \times X \to \mathbf{R}$

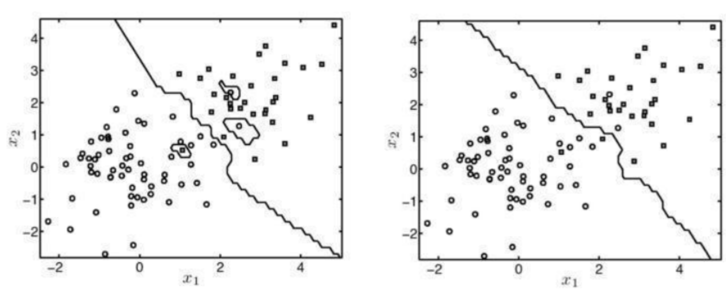
1. $d(x, y) \geq 0$ (non-negativity, or "separation axiom")
2. $d(x, y) = 0$ if and only if $x = y$ (identity of indiscernibles)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality, or "subadditivity")

- Therefore, can be used on any data for which we can define a **distance** between two objects
- KNN has been used successfully for
  - Strings (string edit distance)
  - Graphs (graph edit distance)
  - Images (local feature similarity)

5

KNN Classification

- Three ingredients: Data, Distance Metric, K
- How to choose K ?
  - If K is too **small**, classification may be heavily influenced by noise



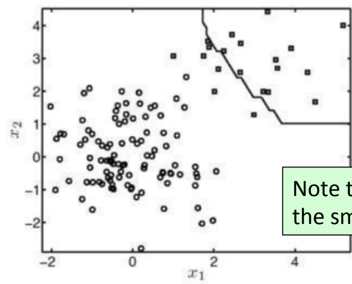
(a) Decision boundary when $K = 1$ (b) Decision boundary when $K = 5$

Including more neighbors has effect of "regularizing" the boundary

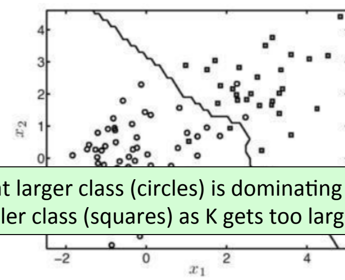
6

KNN Classification

- Three ingredients: Data, Distance Metric, K
- How to choose K ?
  - Increasing K reduces over-fitting, but to a point.
  - If K is too **big**, loose structure (extreme case, N_1=10, K ≥ 21)



(b) Decision boundary when $K = 39$



(b) Decision boundary when $K = 5$

Including more neighbors has effect of "regularizing" the boundary

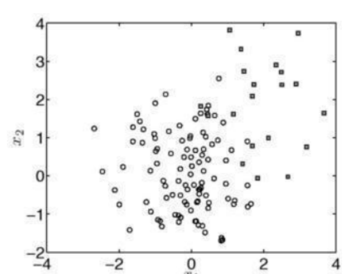
Note that larger class (circles) is dominating the smaller class (squares) as K gets too large

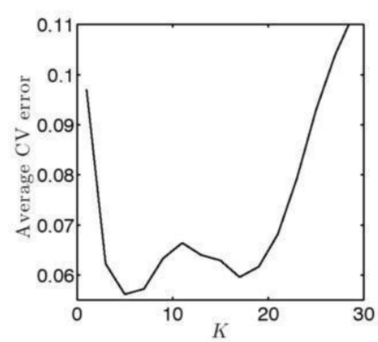
7

KNN Classification

- Three ingredients: Data, Distance Metric, K
- How to choose K ?
  - Most popular way to choose K: cross-validation!
  - Simple performance measure: proportion of mistakes

10-fold CV, repeated 100 times





8

Assessing Classifiers

Assessing Classifiers

- **Consider Binary Classification**
- Decisions can be right or wrong
- How many ways can you be right? Wrong?

	Truly "Yes"	Truly "No"
Say "Yes"	True positive	False Positive
Say "No"	False Negative	True Negative

Lots of functions!

	Truly "Yes"	Truly "No"
Say "Yes"	True positive	False Positive
Say "No"	False Negative	True Negative

true positive (TP)
eqv. with hit
true negative (TN)
eqv. with correct rejection
false positive (FP)
eqv. with false alarm, Type I error
false negative (FN)
eqv. with miss, Type II error
sensitivity or true positive rate (TPR)
eqv. with hit rate, recall
$TPR = TP / P = TP / (TP + FN)$
false positive rate (FPR)
eqv. with fall-out
$FPR = FP / N = FP / (FP + TN)$
accuracy (ACC) ← "classification accuracy"
$ACC = (TP + TN) / (P + N)$
specificity (SPC) or True Negative Rate
$SPC = TN / N = TN / (FP + TN) = 1 - FPR$
positive predictive value (PPV)
eqv. with precision
$PPV = TP / (TP + FP)$
negative predictive value (NPV)
$NPV = TN / (TN + FN)$
false discovery rate (FDR)
$FDR = FP / (FP + TP)$
Matthews correlation coefficient (MCC)
$MCC = (TP * TN - FP * FN)/\sqrt{PNP'N'}$
F1 score
$F1 = 2TP / (P + P')$
Source: Fawcett (2006).

Lots of functions!

	Truly "Yes"	Truly "No"
Say "Yes"	True positive	False Positive
Say "No"	False Negative	True Negative

0/1 Loss
1 = True (positive | negative)
0 = False (positive | negative)

Accuracy: $\frac{\blacksquare}{\blacksquare + \blacksquare}$

E.g.,

Case A:
50% class 1
50% class 2

Imbalanced data
Case B:
80% class 1
20% class 2

How good is 20% loss?

12

Lots of functions!

	Truly "Yes" +, non-Null	Truly "No" –, Null
Say "Yes" +, non-Null	True positive	False Positive
Say "No" –, Null	False Negative	True Negative
Classical statistical hypothesis testing	+: H_a	–: H_0
Epidemiology	+: disease	–: non-disease

True Positive Rate

TP / (TP + FN)

False Positive Rate

FP / (FP + TN)

(Are these rates conditional, joint, or marginal probabilities?)

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

If A = Say "Yes" and B = True "Yes"

$TP = A$ and B

$TP + FN = B$

Type-I error: False Positive Rate

Type-II error: False Negative Rate

= FN / (TP + FN)

13

Related Ideas:

Sensitivity and Specificity

- Common measures of **medical diagnostic tests**
- ***Sensitivity*** is same as the true positive rate
- ***Specificity*** is the number of **detected *negatives*** divided by the total number of negatives:

	Truly "Yes"	Truly "No"
Say "Yes"	True positive	False Positive
Say "No"	False Negative	True Negative

Sensitivity = $\frac{TP}{TP + FN}$

(= True Positive Rate)

Specificity = $\frac{TN}{TN + FP}$

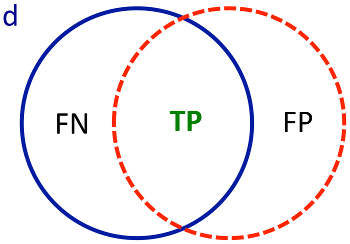
(NOT False Positive Rate! Really: 1 – False Positive Rate)

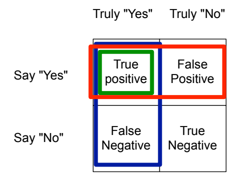
14

Related Ideas:
Information Retrieval

The documents you'd like to retrieve

The documents you actually retrieved





Recall

TP / (TP + FN) (= True Positive Rate)

(How many of the correct docs did I get?)

Precision

TP/ (TP + FP)

(How many of the docs I did get are the ones I wanted?)

F-measure:

$$F = \frac{(2 \times \text{recall} \times \text{precision})}{(\text{recall} + \text{precision})}$$

… an average (harmonic mean) of precision & recall

15

Support Vector Machines

(SVMs)

16

Support Vector Machines (SVMs)

- Considered one of the best "off-the-shelf" classifiers for many problems – state of the art.
- **BUT**, "No free lunch": not guaranteed the best
  - Wolpert & Macready 1997
  - "…any two optimization algorithms are equivalent when their performance is averaged across all possible problems." (from 2005)
- SVMs are particularly useful in applications where the number of attributes is **much larger** than the number of training objects
  - Number of parameters is based on the number of training objects, not the number of attributes!

17

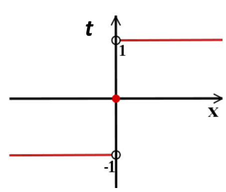
Support Vector Machines (SVMs)

- Standard SVM uses linear decision boundary given by: $\mathbf{w}^\mathsf{T}\mathbf{x}_{\mathsf{new}} + b$
- SVM ***decision function*** for test point:

$$t_{\mathsf{new}} = \mathrm{sign}(\mathbf{w}^\mathsf{T}\mathbf{x}_{\mathsf{new}} + b)$$

labels are $\{1, -1\}$ rather than $\{0, 1\}$

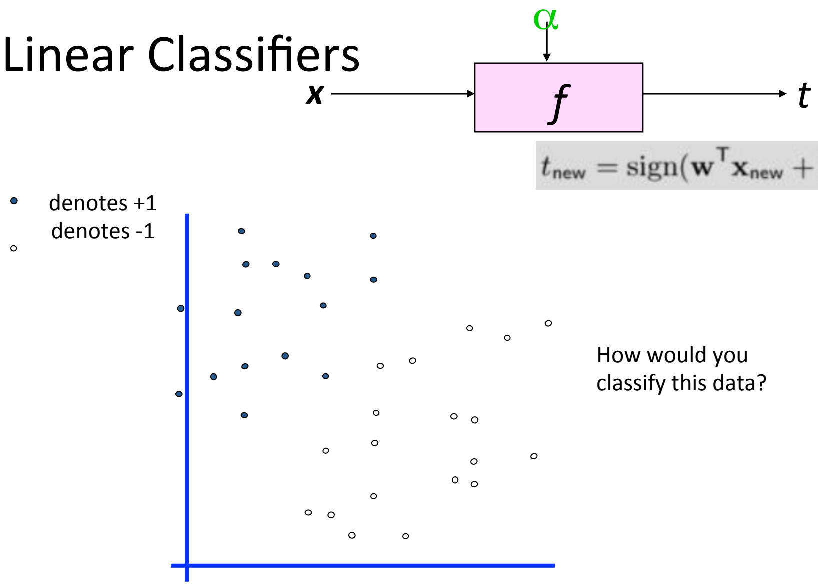
$$\mathrm{sgn}(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$



- **Goal**: find **w** and *b* based on training data
- **Criteria**: Maximize the ***margin***

18

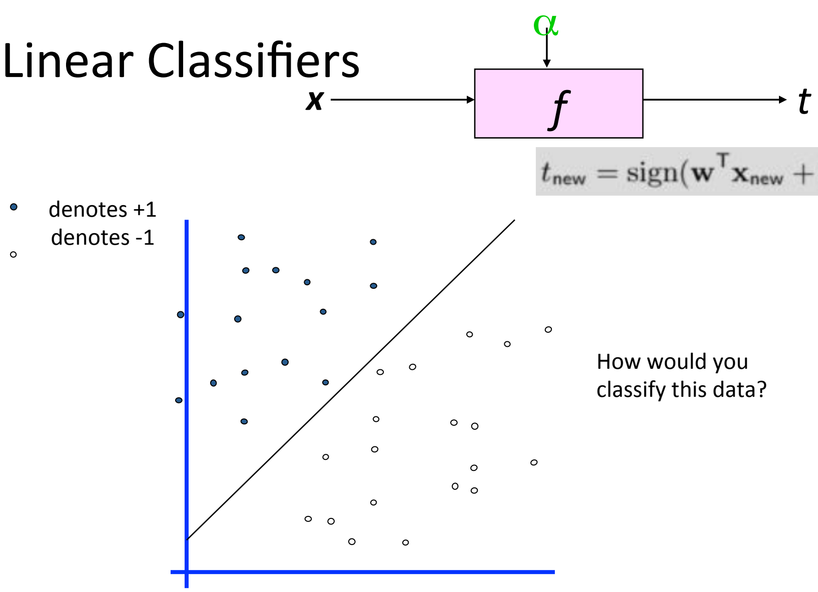
Linear Classifiers



$t_{\text{new}} = \text{sign}(\mathbf{w}^\mathsf{T}\mathbf{x}_{\text{new}} + b)$

How would you classify this data?

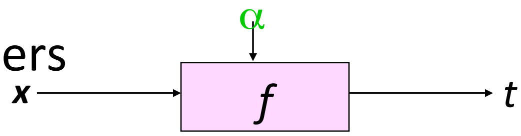
Linear Classifiers



$t_{\text{new}} = \text{sign}(\mathbf{w}^\mathsf{T}\mathbf{x}_{\text{new}} + b)$

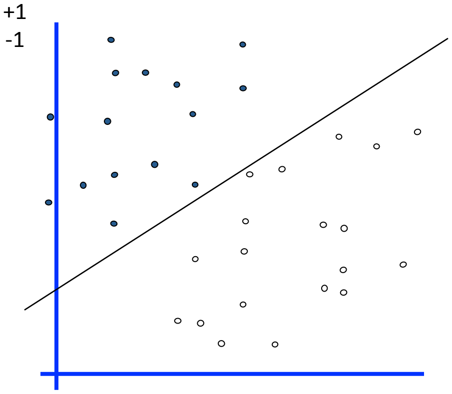
How would you classify this data?

Linear Classifiers



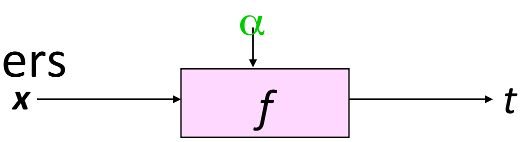
$t_{\text{new}} = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$

- denotes +1
- denotes -1



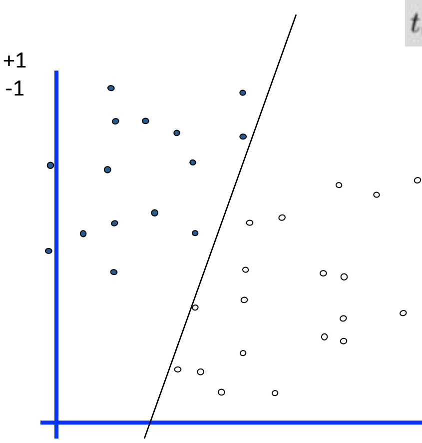
How would you classify this data?

Linear Classifiers



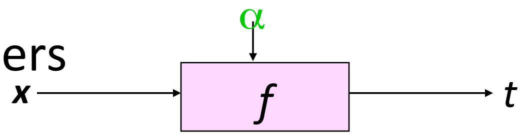
$t_{\text{new}} = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$

- denotes +1
- denotes -1

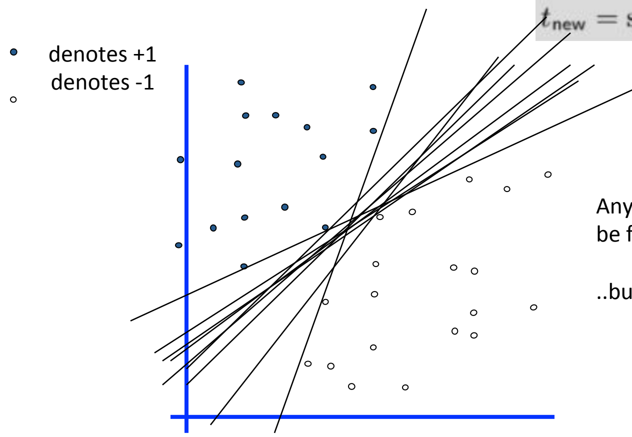


How would you classify this data?

Linear Classifiers



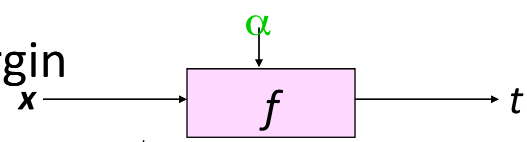
$$t_{\text{new}} = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$$



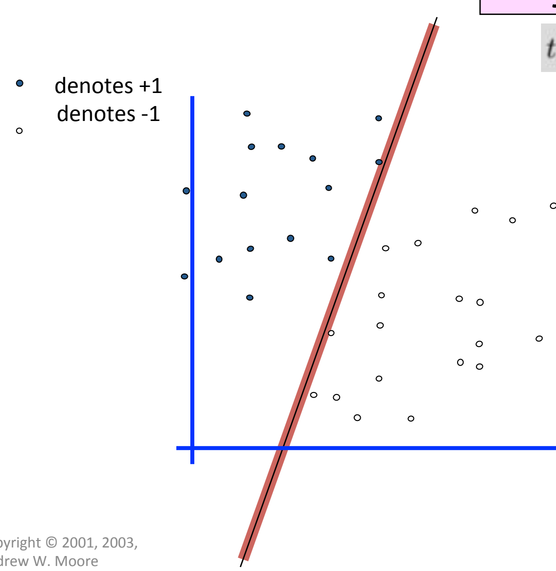
Any of these would be fine..

..but which is best?

Classifier Margin

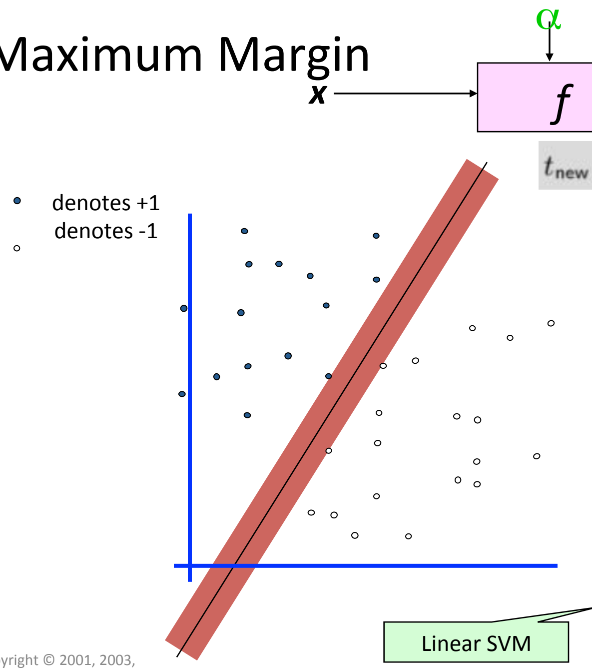


$$t_{\text{new}} = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}} + b)$$



Define the margin of a linear classifier as the width that the boundary (separating the classes) could be increased by before hitting a datapoint.

Maximum Margin

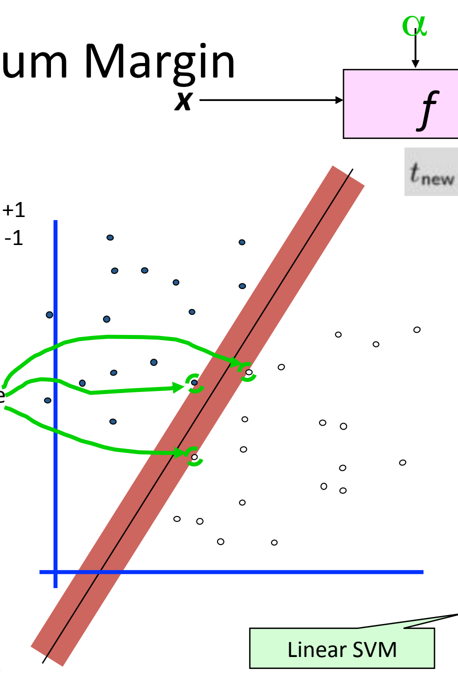


$t_{new} = \text{sign}(\mathbf{w}^T \mathbf{x}_{new} + b)$

The maximum margin linear classifier is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

25

Maximum Margin



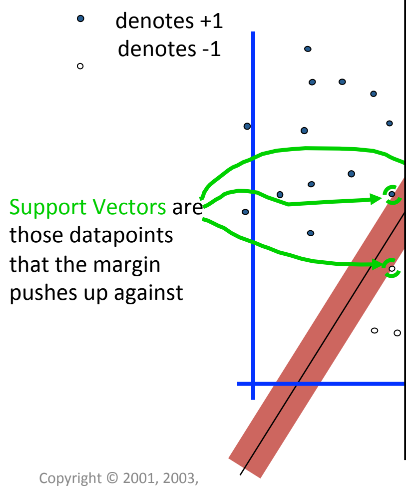
$t_{new} = \text{sign}(\mathbf{w}^T \mathbf{x}_{new} + b)$

Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

26

Why Maximum Margin?



1. Intuitively this feels safest.
2. If we've made a small error in the location of the boundary (it's been jolted in its perpendicular direction) this gives us least chance of causing a misclassification.
3. LOOCV is easy since the model is immune to removal of any non-support-vector datapoints.
4. There's some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
5. Empirically it works very very well.

27