Machine Learning (CE 40477)

Fall 2024

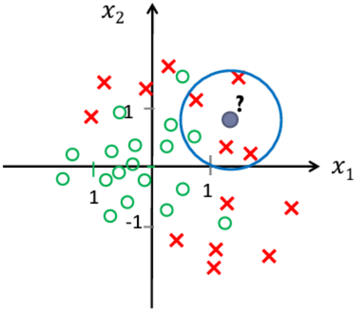
Ali Sharifi-Zarchi

CE Department
Sharif University of Technology

October 4, 2024

kNN

- K-NN classifier: $k \geq 1$ nearest neighbors
  - Label for x predicted by majority voting among its $k - NN$
- $k = 5$, $x = [x_1, x_2]$

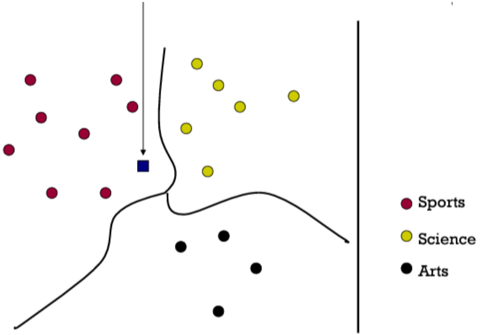


kNN classifier

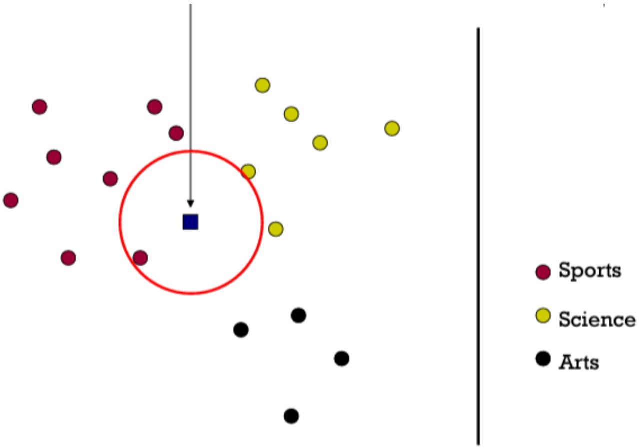
- Given
  - Training data $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ are simply stored.
- To classify x:
  - Find k nearest training samples to x
  - Out of these k samples, identify the number of samples k_j belonging to class C_j ($j = 1, \ldots, C$).
  - Assign x to the class C_{j^*} where $j^* = \underset{j=1,\ldots,c}{\arg\max}\; k_j$
- It can be considered as a **discriminative** method.

kNN example

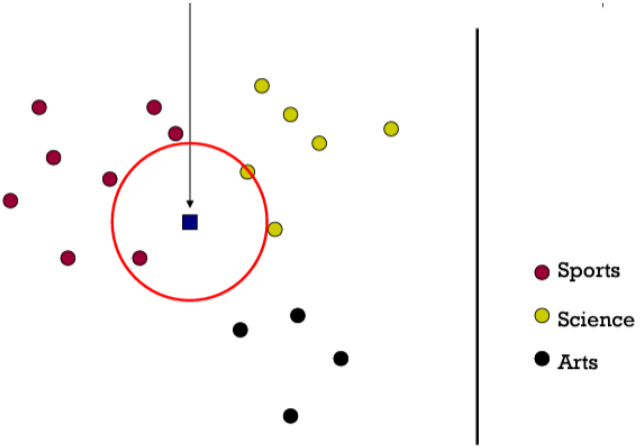
- We want to classify a new document and put it into one of three categories by studying its neighbor samples



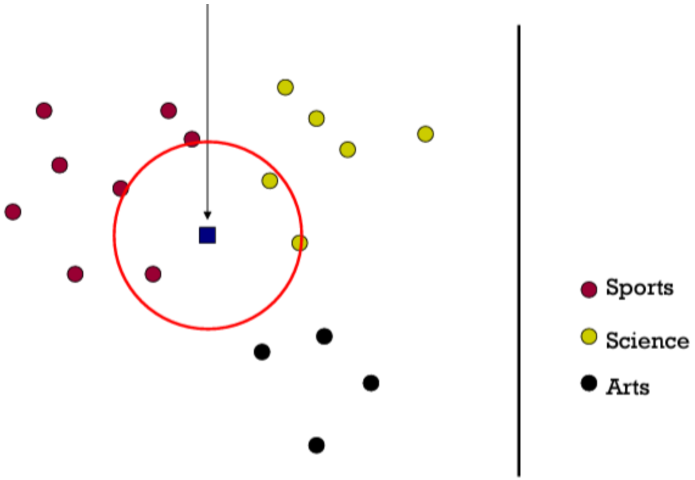
1-Nearest neighbor classifier



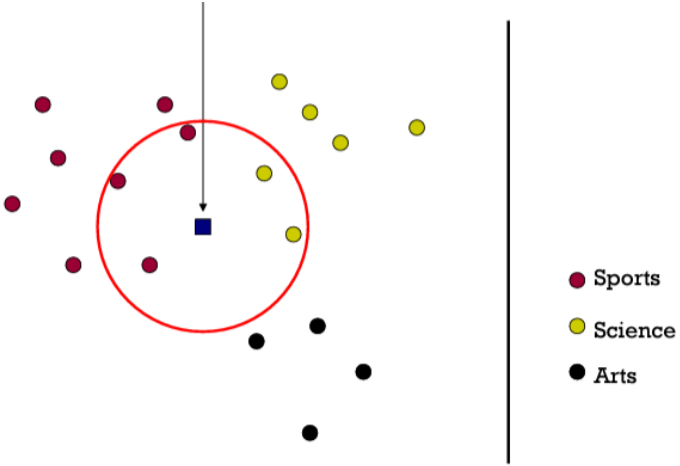
2-Nearest neighbor classifier



3-Nearest neighbor classifier

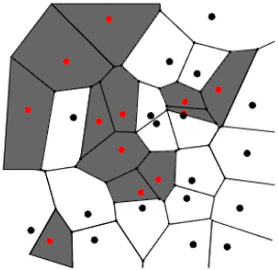


5-Nearest neighbor classifier



Voronoi tessellation

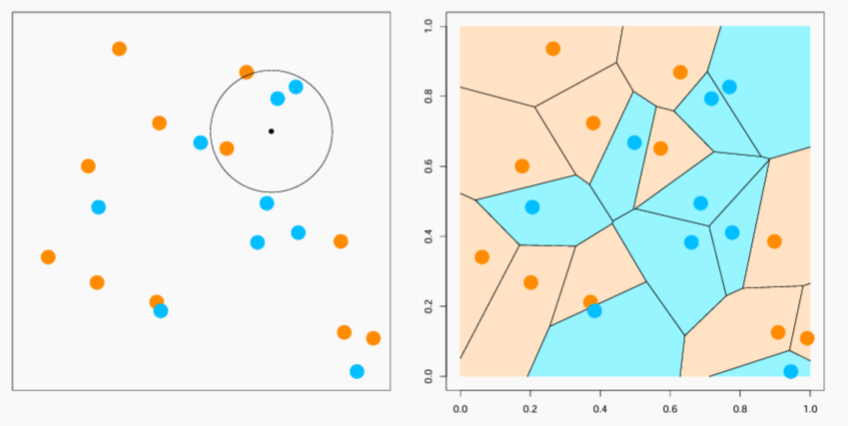
- Voronoi tessellation:
  - Each cell consists of all points closer to a given training point than to any other training points
  - All points in a cell are labeled by the category of the corresponding training point



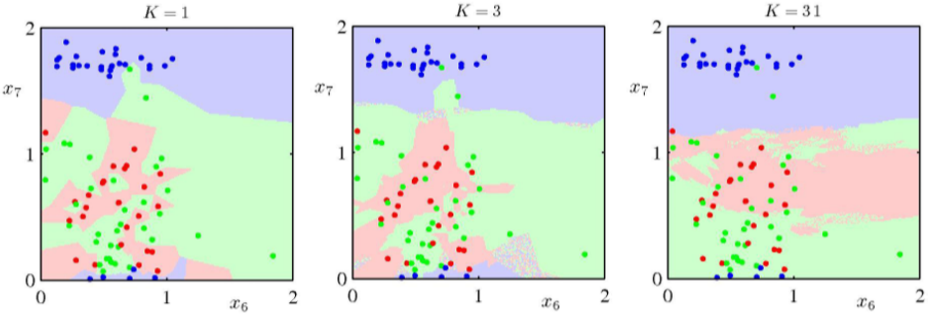
[Duda, Hurt, and Strok's Book]

Voronoi tessellation

- 1NN plot is a Voronoi tessellation



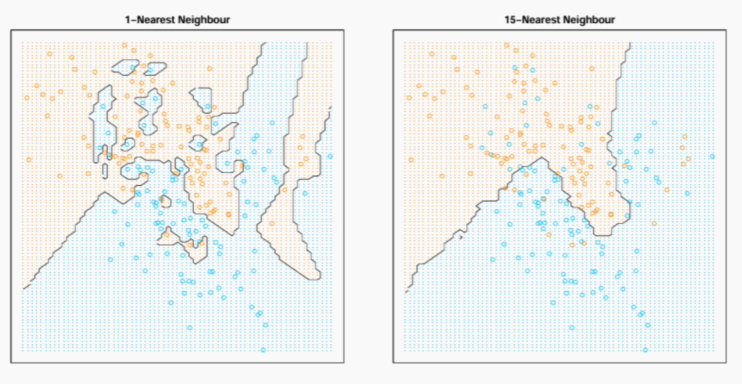
Effect of k



[Bishop]

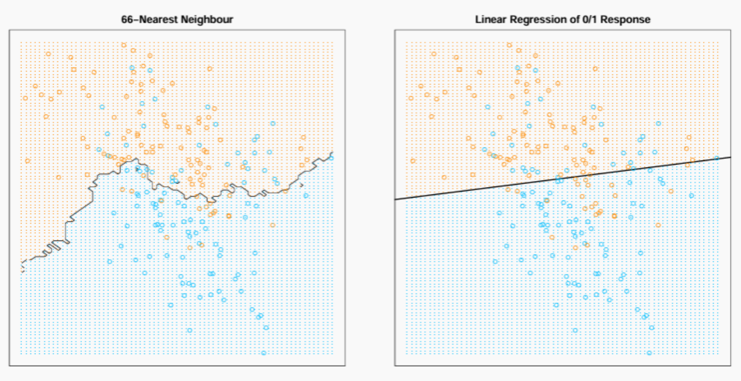
Effect of k cont.

- compare $k = 1$ with $k = 15$



Model complexity

- As we further increase k, the model tends to be less complex.
- Compare $66NN$ with a linear model that uses only 3 parameters:



Parametric vs. non-parametric methods

- Parametric methods need to find parameters from data and then use the inferred parameters to decide on new data points
  - Learning: finding parameters from data
- Non-parametric methods
  - Training examples are explicitly used
  - Training phase is not required
- Both supervised and unsupervised learning can be categorized into parametric and non-parametric methods

Non-parametric learners

- **Memory-based** or **Instance-based** learners
  - lazy learning: (almost) all the work at the test time
- Generic description:
  - Memorize training $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$
  - Given test x predict: $\hat{y} = f(x; x^{(1)}, y^{(1)}, \ldots, x^{(n)}, y^{(n)})$
- f is typically expressed in terms of the similarity of the test samples x to the training samples $x^{(1)}, \ldots, x^{(n)}$

Instance-based learner

- kNN is an instance-based learner
- Main things to construct an instance-based learner:
  - A distance metric
  - Number of nearest neighbors of the test data that we look at
  - A weighting function (optional)
  - How to find the output based on neighbors?

Distance measures

- Euclidean distance

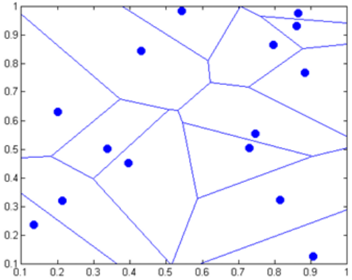
$$d(x, x') = \sqrt{\|x - x'\|_2^2} = \sqrt{(x_1 - x'_1)^2 + \cdots + (x_d - x'_d)^2}$$

- Distance learning methods for this purpose
  - Weighted Euclidean distance

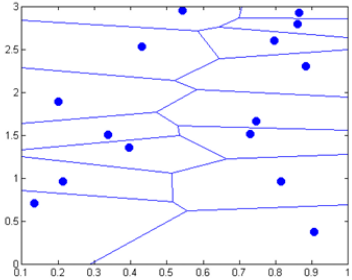
$$d_w(x, x') = \sqrt{w_1(x_1 - x'_1)^2 + \cdots + w_d(x_d - x'_d)^2}$$

- Other distances:
  - Hamming, angle, L-norm, Mahalanobis distance, ...

Effect of distance measure



$$d(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2}$$



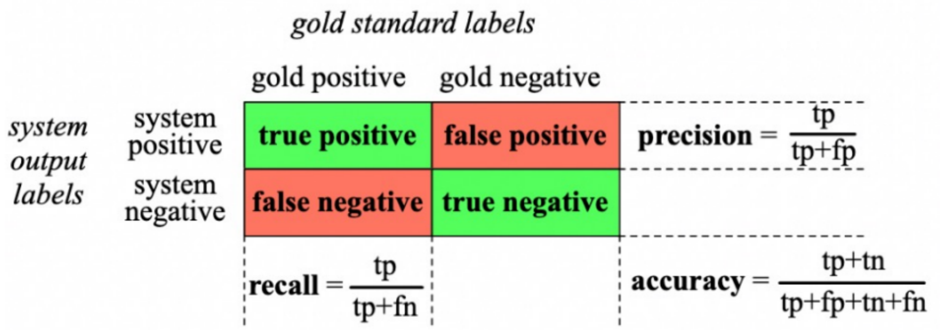
$$d(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{(x_1 - x_1')^2 + 3(x_2 - x_2')^2}$$

Performance metrics

	actually in the class	actually not in the class
predicted to be in the class	tp	fp
predicted not to be in the class	fn	tn

$$\begin{aligned} \text{Precision P} &= \frac{tp}{tp+fp} \\ \text{Recall R} &= \frac{tp}{tp+fn} \\ \text{Accuracy} &= \frac{tp+tn}{tp+tn+fp+fn} \end{aligned}$$

Precision/Recall/Accuracy



A combined measure: F

- Combined measure: **F measure**
  - allows us to trade off precision and recall
  - weighted harmonic mean of P and R

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

You can see: $\beta^2 = \dfrac{1-\alpha}{\alpha}$

A combined measure: F cont.

- People usually use balanced F ($\beta = 1$ or $\alpha = \frac{1}{2}$)

$$F = F_{\beta=1}$$

$$F = \frac{2PR}{P+R}$$

- Harmonic mean of P and R:

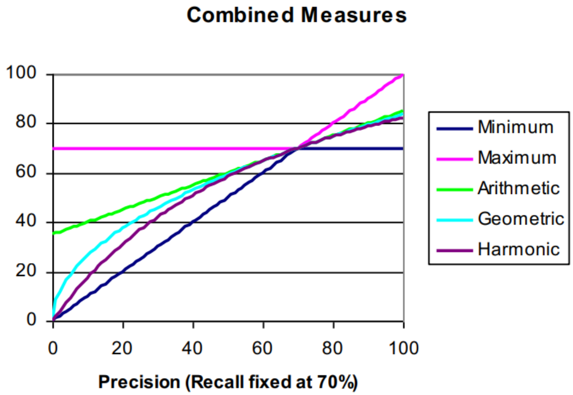
$$\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$$

Why harmonic mean?

- Why don't we use a different mean of P and R as a measure?
  - e.g., the arithmetic mean
- The simple (arithmetic) mean is 50% for "return true for every thing", which is too high.
- Desideratum: Punch really bad performance either on precision or recall
  - Taking the minimum achieves this.
  - F (harmonic mean) is a kind of **smooth minimum**.

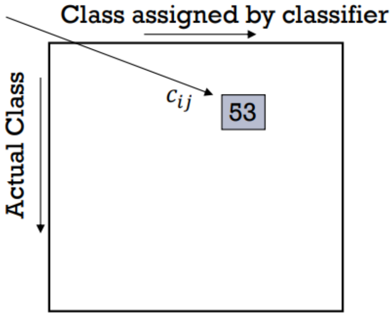
F1 and other averages

- Harmonic mean is a conservative average. We can view the harmonic mean as a kind of soft minimum



Confusion matrix

- This (i, j) entry means 53 of the samples actually in class i were put in class j by the classifier:



- In a perfect classification, only the diagonal has non-zero entries

Per class evaluation measures

- Recall: Fraction of the samples in class i classified correctly:

$$\frac{c_{ii}}{\sum_j c_{ij}}$$

- Precision: Fraction of the samples assigned class i that are actually about class i:

$$\frac{c_{ii}}{\sum_j c_{ji}}$$

- Accuracy: Fraction of the samples classified correctly:

$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

Averaging: macro vs. micro

- We now have an evaluation measure (F1) for one class.
- But we also want a single number that shows aggregate performance over all classes

Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- Macroaveraging: Compute performance for each class, then average
  - Compute F1 for each of the C classes
  - Average these C numbers
- Microaveraging: Collect decisions for all classes, aggregate them and then compute measure.
  - Compute TP, FP, FN for each of the C classes.
  - Sum these C numbers(e.g, all TP to get aggregate TP)
  - Compute F1 for aggregate TP, FP, FN

Micro- vs. Macro-Averaging: example

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- Macroaveraged precision: $(0.5 + 0.9)/2 = 0.7$
- Microaveraged precision: $100/120 = 0.83$
- Microaveraged score is dominated by score on common classes

1. k-Nearest-Neighbor
2. Performance metrics
3. References

Contributions

- **These slides are authored by:**
  - Danial Gharib
  - Mahan Bayhaghi

[1] C. M., *Pattern Recognition and Machine Learning.* Information Science and Statistics, New York, NY: Springer, 1 ed., Aug. 2006.

[2] M. Soleymani Baghshah, "Machine learning." Lecture slides.

[3] M. Soleymani Baghshah, "Modern information retrieval." Lecture slides.

[4] T. Mitchell, *Machine Learning.* McGraw-Hill series in computer science, New York, NY: McGraw-Hill Professional, Mar. 1997.

[5] R. Zhu, "Stat 542: Statistical learning - k-nearest neighbor and the bias-variance trade-off." Lecture notes.

[6] E. Xing, "Theory of classification and nonparametric classifier." Lecture notes.