CS440/ECE448 Lecture 6: Learning

Mark Hasegawa-Johnson, 1/2023

Public domain image: Classes at the University of Bologna. From *Liber ethicorum des Henricus de Alemannia, Laurentius a Voltolina, 14th century, scanned by* The Yorck Project , 2002

Outline

- Biological inspiration
- Parametric learning example: Decision tree
- A mathematical definition of learning
- Non-parametric learning example: K-nearest neighbors
- Training Corpus Error vs. Test Corpus Error

Biological inspiration: Hebbian learning

“Neurons that fire together, wire together.

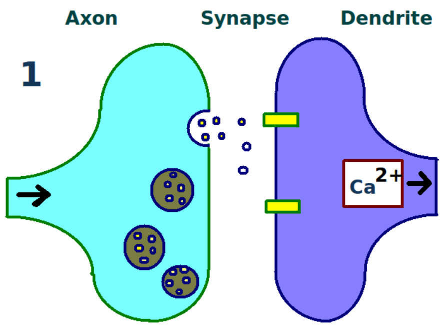
…

The general idea is an old one, that any two cells or systems of cells that are repeatedly active at the same time will tend to become `associated’ so that activity in one facilitates activity in the other.”

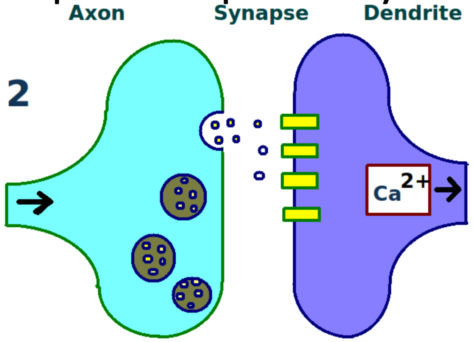
- D.O. Hebb, 1949

Biological inspiration: Long-term potentiation

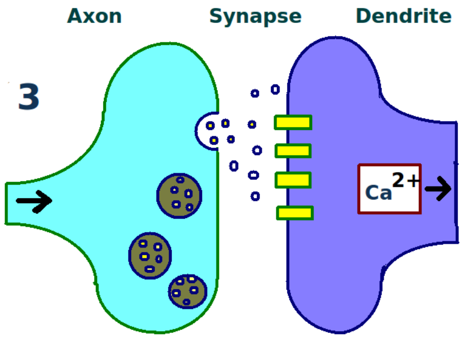
Figures this page are public domain, by Thomas W. Sulcer, 2011



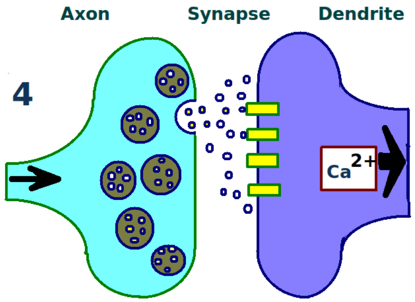
1. A synapse is repeatedly stimulated



2. More dendritic receptors



3. More neurotransmitters

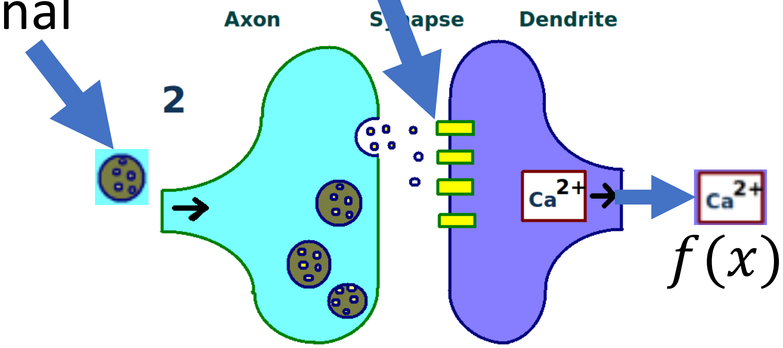


4. A stronger link between neurons

Mathematical model: Learning

Parameters of the learning machine: how many dendritic receptors exist? What types of neurotransmitter do they respond to?

x =input signal

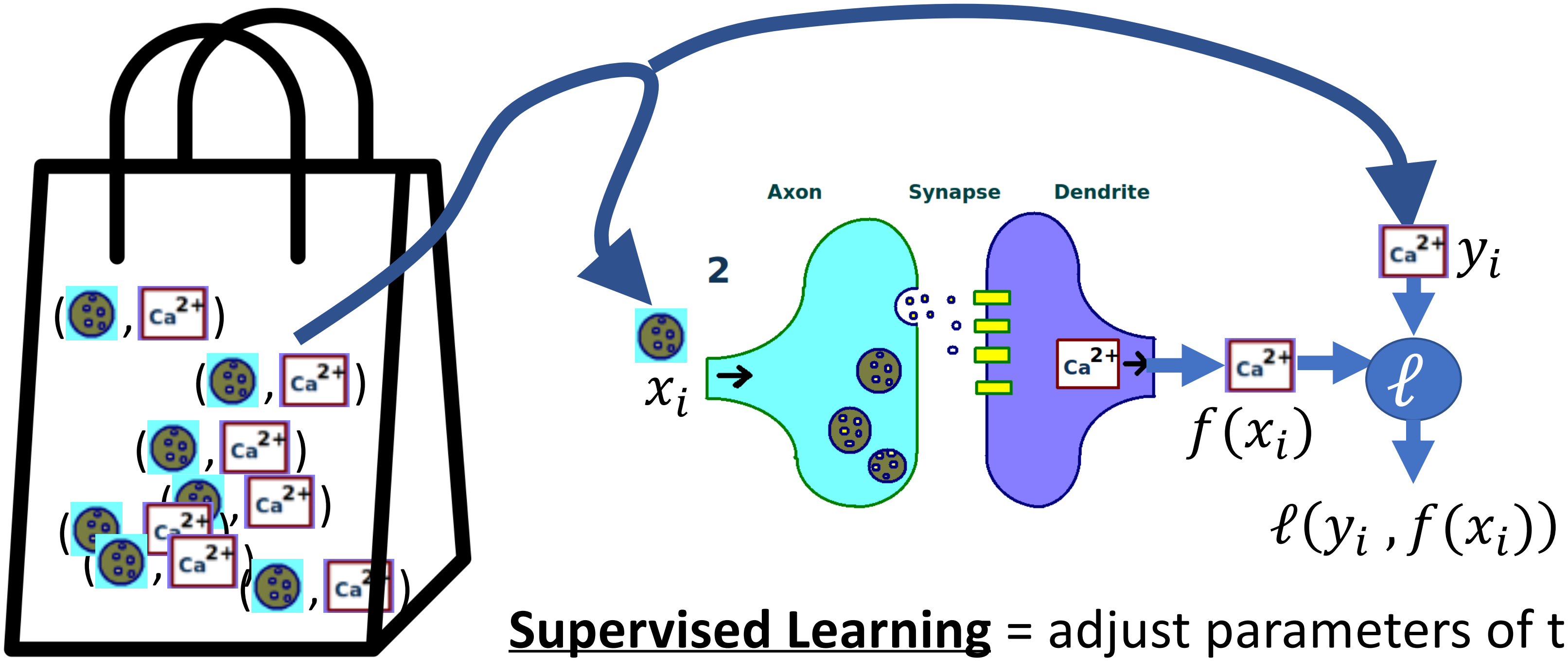


$f(x)$ =output signal

<u>Learning</u> = adjust the parameters of the learning machine so that $f(x)$ becomes the function we want

Mathematical model: Supervised Learning

Supervision: $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ = training dataset containing pairs of (example signal x_i, desired system output y_i)



Supervised Learning = adjust parameters of the learner to minimize $\mathrm{E}[\ell(Y, f(X))]$

Outline

- Biological inspiration
- Parametric learning example: Decision tree
- A mathematical definition of learning
- Non-parametric learning example: K-nearest neighbors
- Training Corpus Error vs. Test Corpus Error

Decision tree learning: An example

- The Titanic sank.
- You were rescued.
- You want to know if your friend was also rescued.
- You can't find them.
- Can you use machine learning methods to estimate the probability that your friend survived?

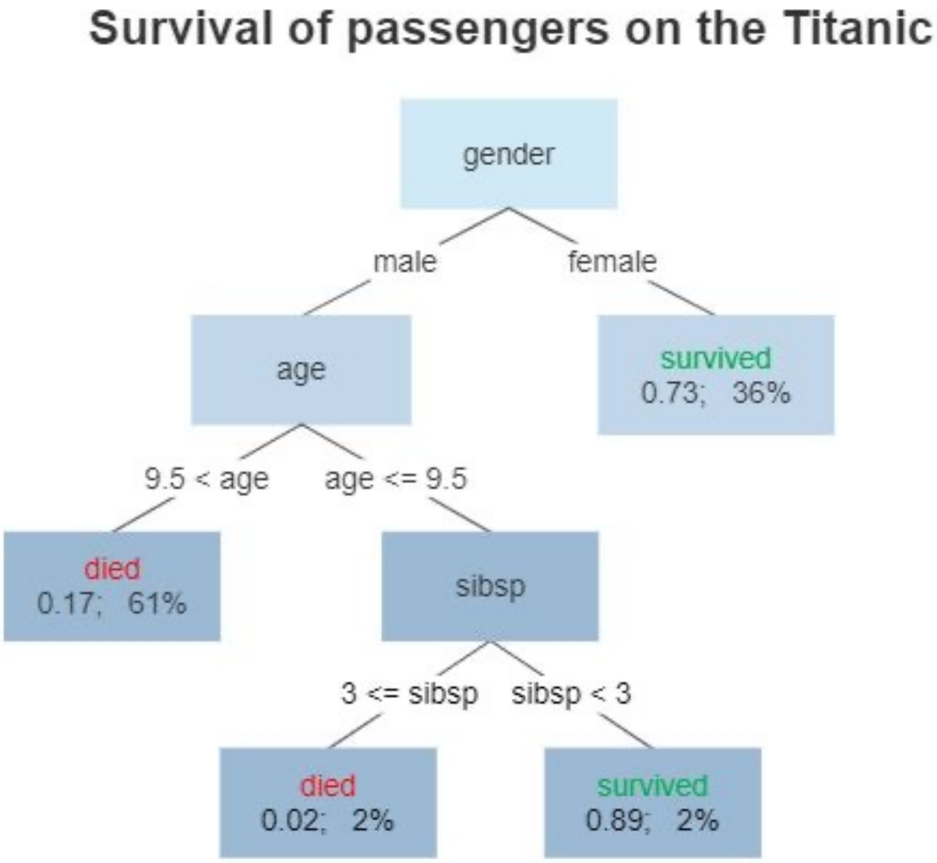
Survival of the Titanic: A machine learning approach

1. Gather data about as many of the passengers as you can.
   - X = variables that describe the passenger, e.g., age, gender, number of siblings on board.
   - Y = 1 if the person is known to have survived
2. Learn a function, f(X), that matches the known data as well as possible
3. Apply f(x) to your friend's facts, to estimate their probability of survival

Survival of the Titanic: A machine learning approach

Decision-tree learning:

- 1st branch = variable that best distinguishes between groups with higher vs. lower survival rates (e.g., gender)
- 2nd branch = variable that best subdivides the remaining group
- Quit when all people in a group have the same outcome, or when the group is too small to be reliably subdivided.

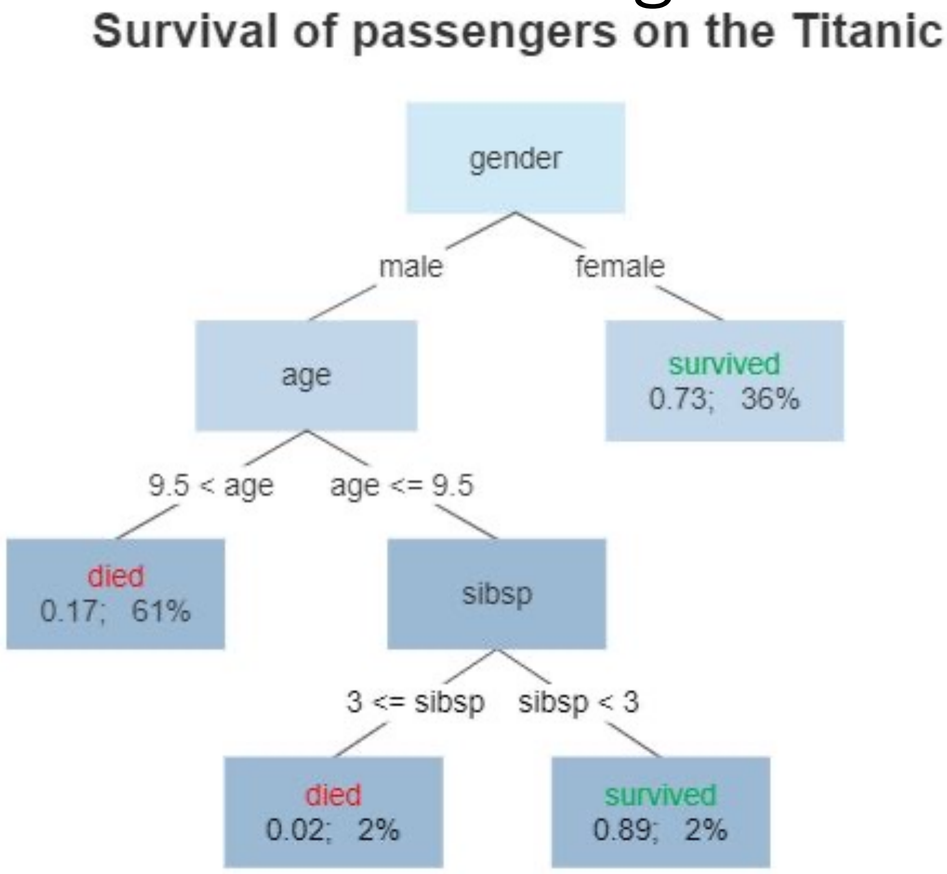


CC-SA 4.0, Gilgoldm, 2020

Survival of the Titanic: A machine learning approach

In each leaf node of this tree:

- Number on the left = probability of survival
- Number on the right = percentage of all known cases that are explained by this node



CC-SA 4.0, Gilgoldm, 2020

Parametric Learner

- A decision tree is an example of a parametric learner
- The function f(x) is determined by some learned parameters
- In this case, the parameters are:
  - Should this node split, or not?
  - If so, which tokens go to the right-hand child?
  - If not, what is $f(x)$ at the current node?
- Titanic shipwreck example:

$$\theta = [Y, \text{female}, Y, \text{age} \leq 9.5, N, f(x) = 0.73, \ldots]$$

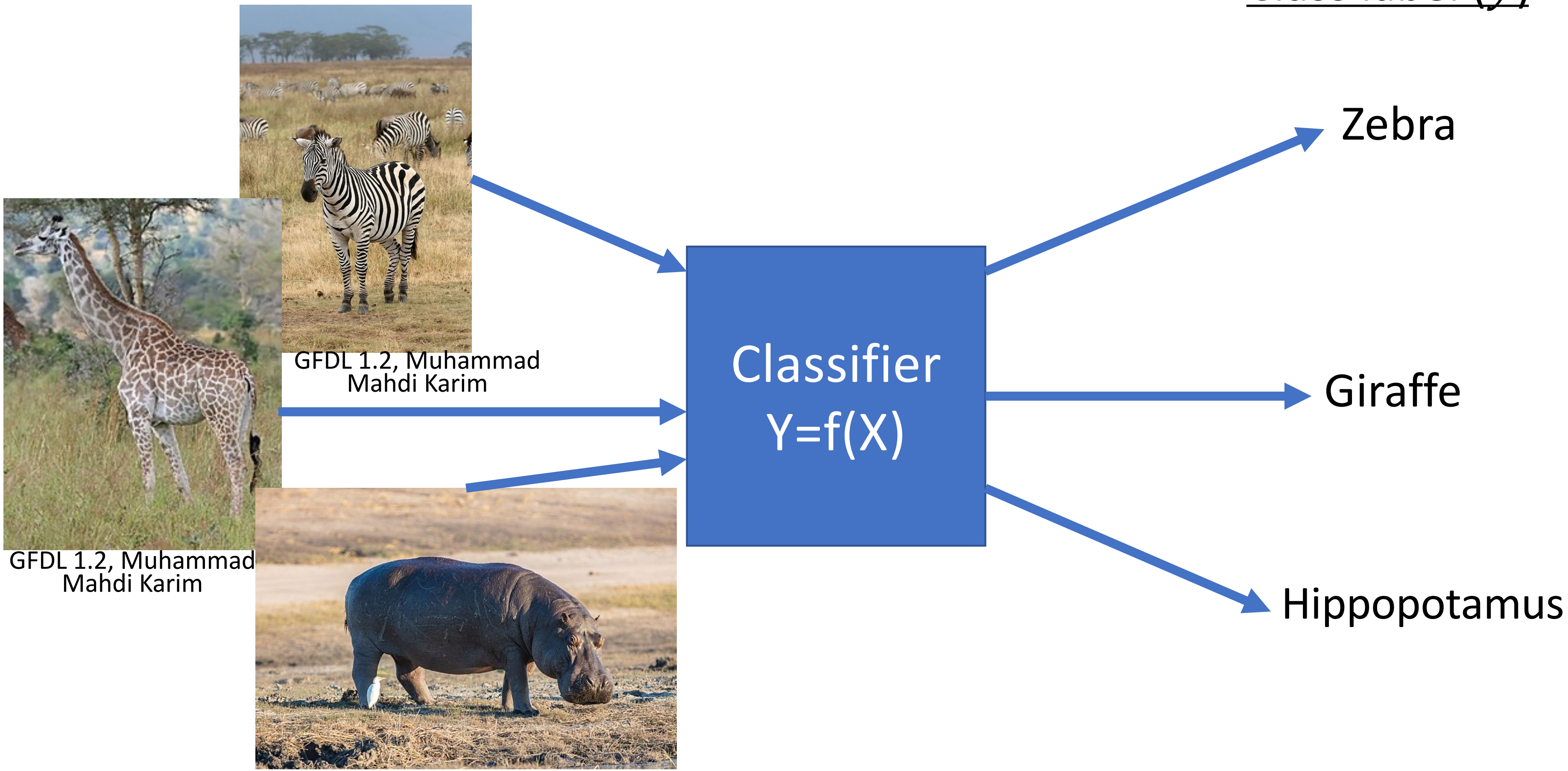
Outline

- Biological inspiration
- Parametric learning example: Decision tree
- A mathematical definition of learning
- Non-parametric learning example: K-nearest neighbors
- Training Corpus Error vs. Test Corpus Error

Learning: learn a function $\hat{y} = f(x)$, where x=features, y=true label, $\hat{y}$=estimated label

<u>Features (x)</u>

<u>Class label (y)</u>



GFDL 1.2, Muhammad Mahdi Karim

GFDL 1.2, Muhammad Mahdi Karim

CC-BY-SA, Diego Delso

Classifier

Y=f(X)

Zebra

Giraffe

Hippopotamus

A mathematical definition of learning

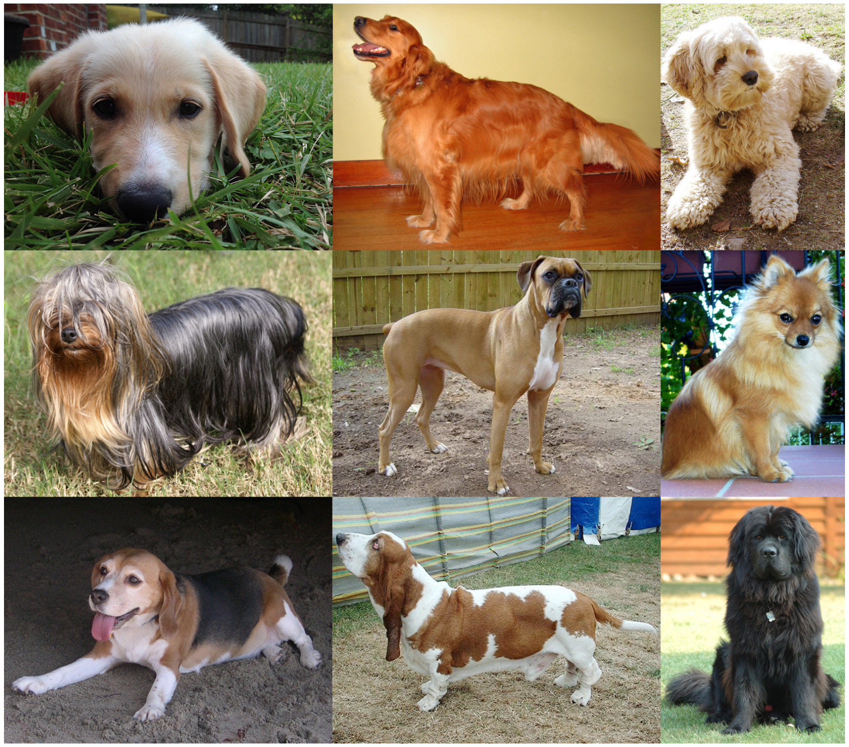
- **Environment:** there are two random variables, $x \sim X$ and $y \sim Y$, that are jointly distributed according to
$$P(X, Y)$$
- **Data:** $P(X, Y)$ is unknown, but we have a sample of training data
$$\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$$
- **Objective:** We would like a function f that minimizes the expected value of some loss function, $\ell(Y\ , f(X))$:
$$\mathcal{R} = \mathrm{E}[\ell(Y, f(X))\ ]$$
- **Definition of learning:** Learning is the task of estimating the function f, given knowledge of $\mathcal{D}$.

Outline

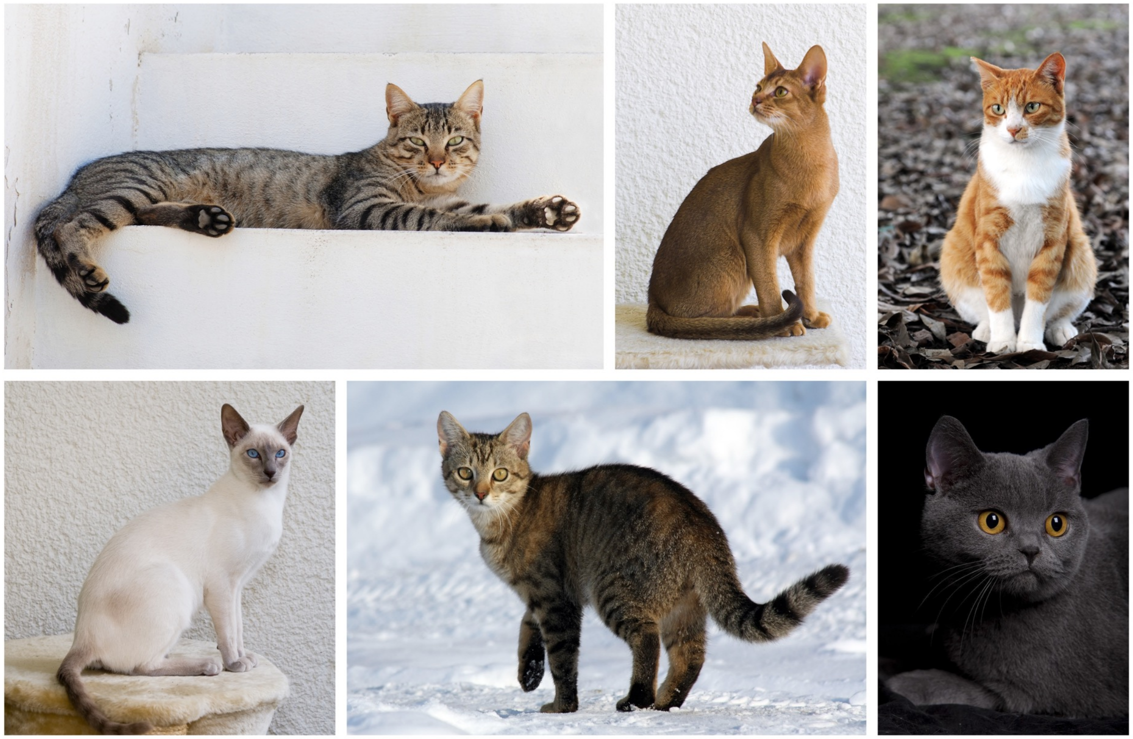
- Biological inspiration
- Parametric learning example: Decision tree
- A mathematical definition of learning
- Non-parametric learning example: K-nearest neighbors
- Training Corpus Error vs. Test Corpus Error

Classifier example: dogs versus cats

Can you write a program that can tell which ones are dogs, and which ones are cats?



By YellowLabradorLooking_new.jpg: *derivative work: Djmirko (talk)YellowLabradorLooking.jpg: User:HabjGolden_Retriever_Sammy.jpg: Pharaoh HoundCockerpoo.jpg: ALMMLonghaired_yorkie.jpg: Ed Garcia from United StatesBoxer_female_brown.jpg: Flickr user boxercabMilù_050.JPG: AleRBeagle1.jpg: TobycatBasset_Hound_600.jpg: ToBNewfoundland_dog_Smoky.jpg: Flickr user DanDee Shotsderivative work: December21st2012Freak (talk) - YellowLabradorLooking_new.jpgGolden_Retriever_Sammy.jpgCockerpoo.jpgLonghaired_yorkie.jpgBoxer_female_brown.jpgMilù_050.JPGBeagle1.jpgBasset_Hound_600.jpgNewfoundland_dog_Smoky.jpg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10793219



By Alvesgaspar - Top left:File:Cat August 2010-4.jpg by AlvesgasparTop middle:File:Gustav chocolate.jpg by Martin BahmannTop right:File:Orange tabby cat sitting on fallen leaves-Hisashi-01A.jpg by HisashiBottom left:File:Siam lilacpoint.jpg by Martin BahmannBottom middle:File:Felis catus-cat on snow.jpg by Von.grzankaBottom right:File:Sheba1.JPG by Dovenetel, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17960205

Nearest Neighbors Classifier

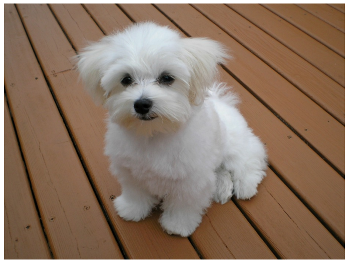
- Given n different **training images**. Each one has a known class label.

- Input to the classifier: a **test image** x whose correct label is unknown.
- Classification function:
  1. Find the training token, x_i, that is most similar to the test token.
  2. Find out the corresponding class label, y_i =correct_label(x_i).
  3. Output y_i as the best guess for the label of test token x.

Example of Nearest-Neighbor Classification

Test Token: Maltese

CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=55084303

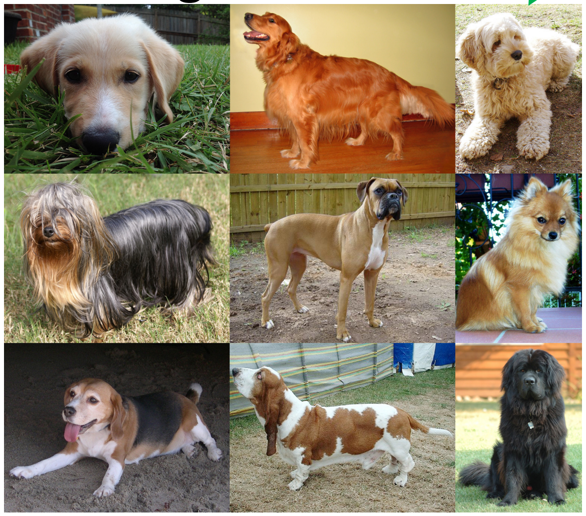


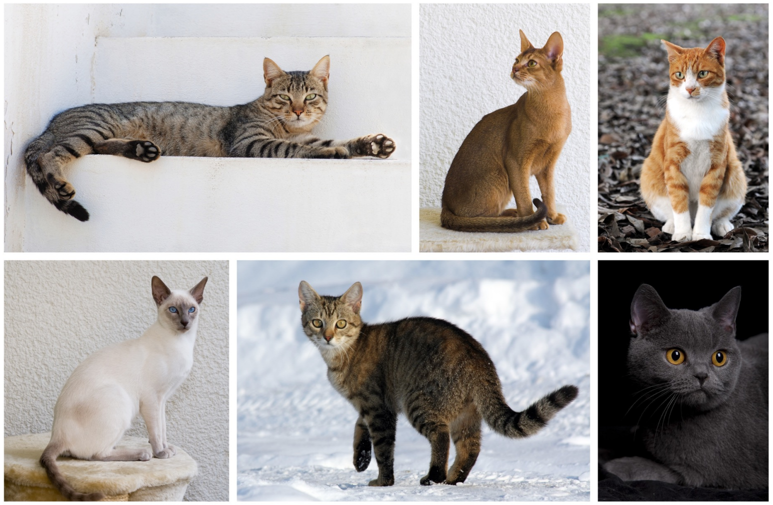
This is the most similar training token…

Therefore the Maltese is classified as a dog.

Training Tokens:

By YellowLabradorLooking_new.jpg: *derivative work: Djmirko (talk)YellowLabradorLooking.jpg: User:HabjGolden_Retriever_Sammy.jpg: Pharaoh HoundCockerpoo.jpg: ALMMLonghaired_yorkie.jpg: Ed Garcia from United StatesBoxer_female_brown.jpg: Flickr user boxercabMilù_050.JPG: AleRBeagle1.jpg: TobycatBasset_Hound_600.jpg: ToBNewfoundland_dog_Smoky.jpg: Flickr user DanDee Shotsderivative work: December21st2012Freak (talk) - YellowLabradorLooking_new.jpgGolden_Retriever_Sammy.jpgCockerpoo.jpgLonghaired_yorkie.jpgBoxer_female_brown.jpgMilù_050.JPGBeagle1.jpgBasset_Hound_600.jpgNewfoundland_dog_Smoky.jpg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10793219





By Alvesgaspar - Top left:File:Cat August 2010-4.jpg by AlvesgasparTop middle:File:Gustav chocolate.jpg by Martin BahmannTop right:File:Orange tabby cat sitting on fallen leaves-Hisashi-01A.jpg by HisashiBottom left:File:Siam lilacpoint.jpg by Martin BahmannBottom middle:File:Felis catus-cat on snow.jpg by Von.grzankaBottom right:File:Sheba1.JPG by Dovenetel, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17960205

K-Nearest Neighbors (KNN) Classifier

The nearest-neighbors classifier sometimes fails if one of the training tokens is unusual. In that case, a test token that is similar to the weird training token might get misclassified. Solution: K-Nearest Neighbors.

Test token:

Mandruss, CC BY-SA 4.0

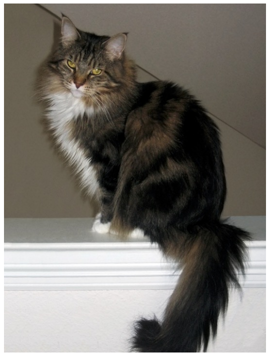
Most similar training token:

DK1k, CC BY-SA 4.0

K-Nearest Neighbors Classification Function

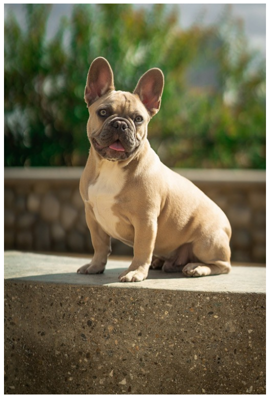
1. Find the K training tokens, x_i, that are most similar to the test token (K is a number chosen in advance by the system designer, e.g., $K = 3$).
2. Find out the corresponding class labels, y_i =correct_label(x_i).
3. Vote! Find the class label that is most frequent among the K-nearest neighbors, and output that as the label of the test token.

Test token:

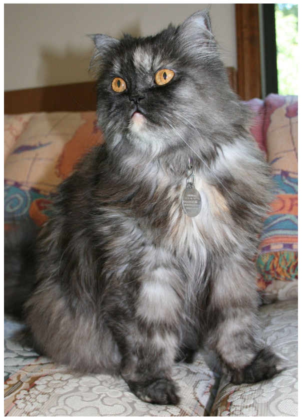


Mandruss, CC BY-SA 4.0

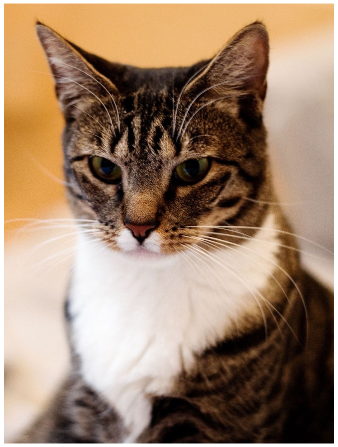
3 most similar training tokens:



DK1k, CC BY-SA 4.0



Mike Powell, CC BY-SA 2.0



Dustin Warrington, CC BY-SA.

Try the quiz!

- Try the quiz at https://us.prairielearn.com/pl/course_instance/129874/assessment/2328563

Non-Parametric Learner

- KNN is an example of a non-parametric learner.
- The function f(x) is determined by a memorized copy of the entire training database.
- There are no numerical parameters that can summarize the behavior of f(x); in order to know what f(x) computes for any given test token, you need to know the entire training database.

Outline

- Biological inspiration
- Parametric learning example: Decision tree
- A mathematical definition of learning
- Non-parametric learning example: K-nearest neighbors
- Training Corpus Error vs. Test Corpus Error

Training corpus error vs. Test corpus error

- **Learning:** Given $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, find the function $f(X)$ that minimizes some measure of risk.
- **Empirical risk**, a.k.a. training corpus error:
$$\mathcal{R}_{\mathrm{emp}} = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$$
- **True risk,** a.k.a. expected test corpus error:
$$\mathcal{R} = \mathrm{E}[\ell(Y, f(X))]$$
- If training and test data are i.i.d.,
$$\lim_{n\to\infty} \mathcal{R}_{\mathrm{emp}} = \mathcal{R}$$

Training vs. Test Corpora

Training Corpus = a set of data that you use in order to optimize the parameters of your classifier (for example, optimize which features you measure, how you use those features to make decisions, and so on).

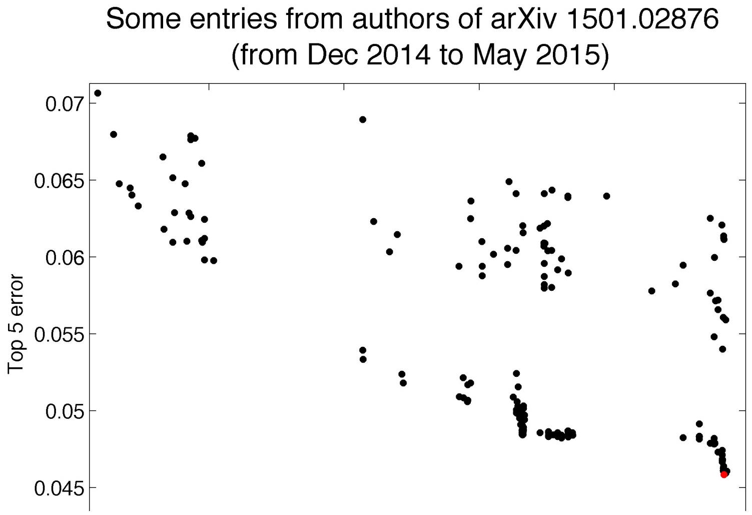
Test Corpus = a set of data that is non-overlapping with the training set (none of the test tokens are also in the training dataset) that you can use to measure the accuracy.

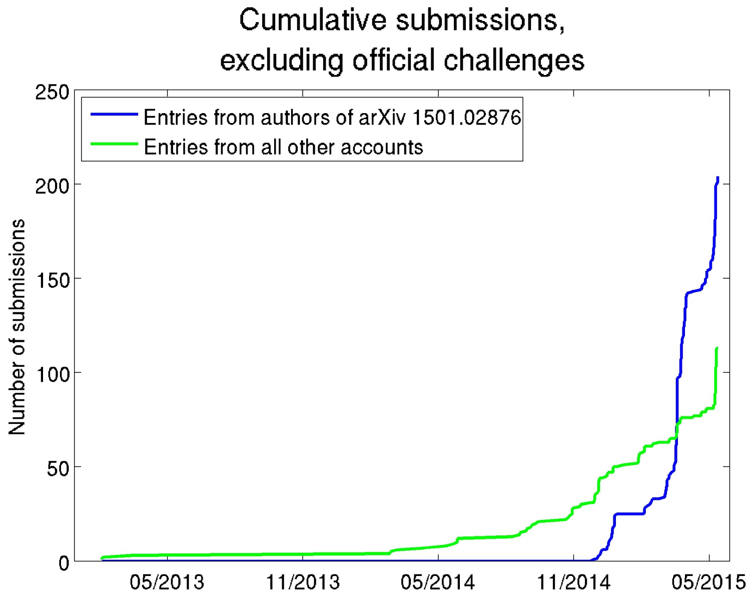
- Measuring the training corpus accuracy is useful for debugging: if your training algorithm is working, then training corpus accuracy should always go up.
- Measuring the test corpus accuracy is the only way to estimate how your classifier will work on new data (data that you’ve never yet seen).

Accuracy on which corpus?

This happened:

- Large Scale Visual Recognition Challenge 2015: Each competing institution was allowed to test up to 2 different fully-trained classifiers per week.
- One institution used 30 different e-mail addresses so that they could test a lot more classifiers (200, total). One of their systems achieved <46% error rate – the competition's best, at that time.
- Is it correct to say that that institution's algorithm was the best?


Some entries from authors of arXiv 1501.02876
(from Dec 2014 to May 2015)


Cumulative submissions,
excluding official challenges

Training vs. development test vs. evaluation test corpora

Training Corpus = a set of data that you use in order to optimize the parameters of your classifier (for example, optimize which features you measure, what are the weights of those features, what are the thresholds, and so on).

Development Test (DevTest or Validation) Corpus = a dataset, separate from the training dataset, on which you test 200 different fully-trained classifiers (trained, e.g., using different training algorithms, or different features) to find the best.

Evaluation Test Corpus = a dataset that is used only to test the ONE classifier that does best on DevTest. From this corpus, you learn how well your classifier will perform in the real world.

Summary

- **Biological inspiration:** Neurons that fire together wire together. Given enough training examples (x_i, y_i), can we learn a desired function so that $f(x) \approx y$?
- **Classification tree:** Learn a sequence of if-then statements that computes $f(x) \approx y$
- **Mathematical definition of supervised learning:** Given a training dataset, $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, find a function f that minimizes the risk, $\mathcal{R} = \mathrm{E}[\ell(Y, f(X))]$.
- **KNN:** Find K training examples that most resemble the test example; let them vote to decide the class label