DS 310 Machine Learning

Vasant G. Honavar
Dorothy Foehr Huck and J. Lloyd Huck Chair in Biomedical Data Sciences and Artificial Intelligence
Professor of Data Sciences, Informatics, Computer Science and Engineering, Bioinformatics & Genomics, Public Health Sciences and Neuroscience
Director, Center for Artificial Intelligence Foundations and Scientific Applications
Associate Director, Institute for Computational and Data Sciences
Pennsylvania State University

vhonavar@psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu

Quick review of Python

- https://colab.research.google.com/?authuser=1

Now, on to some real content …

Classification

- How would you write a program to distinguish a picture of you from a picture of someone else?
  - Provide examples pictures of you and pictures of other people and let a classifier learn to distinguish the two.
- How would you write a program to determine whether a sentence is grammatical or not?
  - Provide examples of grammatical and ungrammatical sentences and let a classifier learn to distinguish the two.
- How would you write a program to distinguish cancerous cells from normal cells?
  - Provide examples of cancerous and normal cells and let a classifier learn to distinguish the two.

Example: To play or not to play tennis

- **Example dataset**

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Three key elements**
  - Class label ("label", denoted by y)
  - Features ("attributes")
  - Feature values ("attribute values", denoted by x)

  Feature values can be binary, nominal or continuous
- **A** *labeled* dataset **is a collection of** (x, y) **pairs**

Example: To play or not to play tennis?

- **Example dataset**

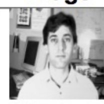
Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

- **Task:**

Class	Outlook	Temperature	Windy?
???	Sunny	Low	No

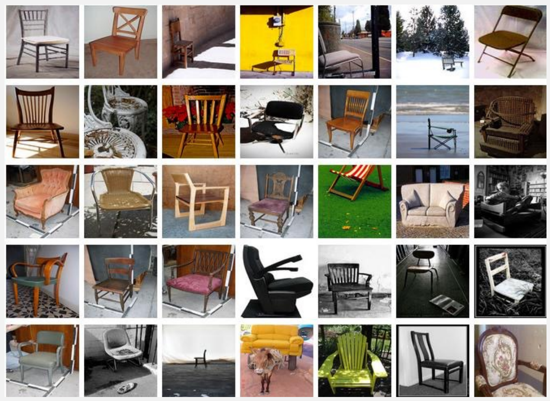
- **Predict the class of this "test" sample**
- Requires us to **generalize** from the training data

Example (face recognition)

Class	Image	Class	Image
Avrim		Tom	
Avrim		Tom	
Avrim		Tom	
Avrim		Tom	

What is a good *representation* for images? Pixel values? Edges?

Example (chair detection)



A deeper dive into data

- What do we mean by data?
  - Digital representation of objects, entities, persons, events, processes, etc. in the real world
    - Employees
    - Genomic sequences
    - Social relationships
    - Images
    - Documents
    - Medical histories
    - …..

Tabular data

- Objects or entities are represented by rows in a table.
- Columns of the table encode properties or characteristics, features, of the objects
- Each object is represented by specifying the values of each attribute
- We call the set of all possible values of an attributes its domain
  - Domain of Refund is {Yes, No}
  - Domain of Taxable Income is $\Re^+$ (positive real numbers)

Attributes

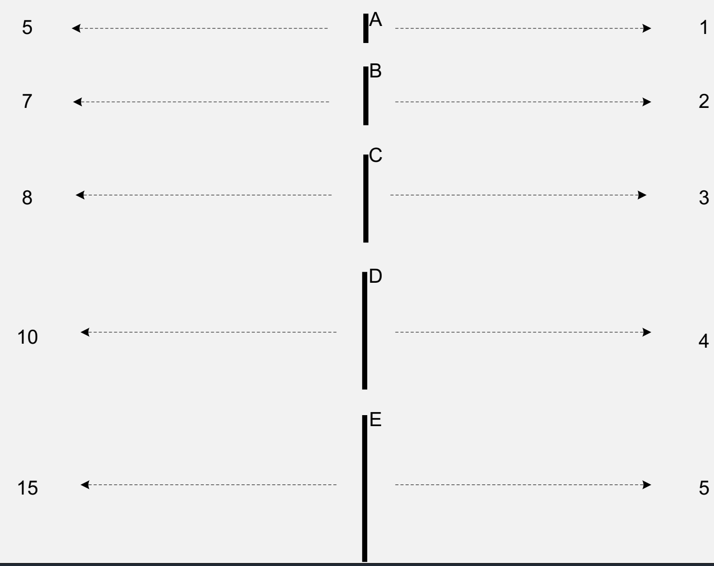
Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

The way you encode an attribute has consequences

- Two different encodings of lengths of objects

This scale preserves only the ordering property of length.



This scale preserves the ordering and additivity property of length.

Attributes come in many flavors

- There are different types of attributes
- Nominal
  - Examples: ID numbers, eye color, zip codes
- Ordinal
  - Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height {tall, medium, short}
- Interval
  - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
- Ratio
  - Examples: temperature in Kelvin, length, counts, elapsed time (e.g., time to run a race)

Properties of Attribute Values

- Different types of attributes possess different properties:
  - Distinctness: $= \neq$
  - Order: $< >$
  - Meaningfulness of differences $+ -$
  - Meaningfulness of ratios $* /$
  - Nominal attribute: distinctness
  - Ordinal attribute: distinctness & order
  - Interval attribute: distinctness, order & meaningfulness of differences
  - Ratio attribute: All 4 properties

Measurement is a tricky subject

- Temperature is measured in Kelvin, degrees Celsius, and degrees Fahrenheit
  - Temp in Kelvin = Temp in degrees Celsius + 273.15
  - Temp in Fahrenheit = (Temp in degrees Celsius)(9/5)+32
- Is it physically meaningful to say that a temperature of 10 ° Celsius is twice as high as 5° Celsius?
  - Not really. Think about the measurement scale. It is relative to the freezing and boiling point of water, not absolute.
- Consider measuring height
  - If Bill's height is three inches above average and Bob's height is six inches above average, then would we say that Bob is twice as tall as Bill?

	Attribute Type	Description	Examples	Operations
Categorical Qualitative	Nominal	Nominal attribute values only distinguish. (=, ≠)	zip codes, employee ID numbers, eye color, sex: {*male, female*}	mode, entropy, contingency correlation, $\chi 2$ test
	Ordinal	Ordinal attribute values also order objects. (<, >)	hardness of minerals, {*good, better, best*}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric Quantitative	Interval	For interval attributes, differences between values are meaningful. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, *t* and *F* tests
	Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, current	geometric mean, harmonic mean, percent variation

	Attribute Type	Transformation	Comments
Categorical Qualitative	Nominal	Any permutation of values	If all employee ID numbers were reassigned, would it make any difference?
	Ordinal	An order preserving change of values, i.e., *new_value = f(old_value)* where *f* is a monotonic function	An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by { 0.5, 1, 10}.
Numeric Quantitative	Interval	*new_value = a * old_value + b* where a and b are constants	Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).
	Ratio	*new_value = a * old_value*	Length can be measured in meters or feet.

Discrete and Continuous Attributes

- Discrete Attribute
  - Has only a finite or countably infinite set of values
  - Examples: zip codes, counts, or the set of words in a collection of documents
  - Often represented as integer variables.
  - Binary attributes are a special case of discrete attributes
- Continuous Attribute
  - Takes real numbers as values
  - Examples: temperature, height, or weight.
  - Practically, real values can only be measured and represented using a finite number of digits.
  - Continuous attributes are typically represented as floating-point numbers.

Asymmetric Attributes

- Only presence (a non-zero attribute value) matters
  - Words present in documents
  - Items present in customer transactions

- If you run into a friend at the grocery store would you ever say the following?

  "We have similar taste because I did not buy almost every item that you also did not buy"

Points to remember about attribute types

- The types of operations you choose should be “meaningful” for the type of data you have
  - Distinctness, order, meaningful intervals, and meaningful ratios are only four (among many possible) properties of data
  - The data type you see – often numbers or strings – may not capture all the properties or may suggest properties that are not present
  - Analysis may depend on these other properties of the data
  - In the end, what is meaningful may be domain-specific

Important Characteristics of Data

- Dimensionality (number of attributes)
- Sparsity
- Resolution
- Size

Types of data

- Tabular data
- Document Data
- Transaction Data
- Graph
  - World Wide Web
  - Molecular Structures
  - Social networks
- Ordered
  - Clinical histories
  - System call sequences
  - Genome Sequences Sequence Data

Tabular data

- Data that consists of a collection of records, each of which encoded by a fixed set of attributes

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tabular data

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute

- Such a data set can be represented by an *m* by *n* matrix, where there are *m* rows, one for each object, and *n* columns, one for each attribute

Document Data

- Each document is encoded using a vector of word frequencies
  - Each term is a component (attribute) of the vector
  - The value of each component is the number of times the corresponding word occurs in the document.

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

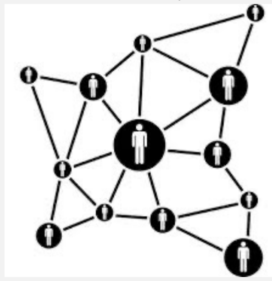
Transaction Data

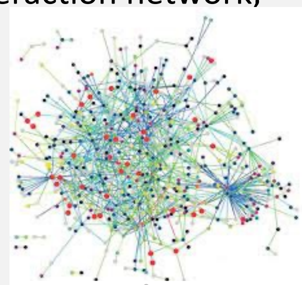
- A special type of data, where
  - Each transaction involves a set of items.
  - For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.
  - Can represent transaction data as record data

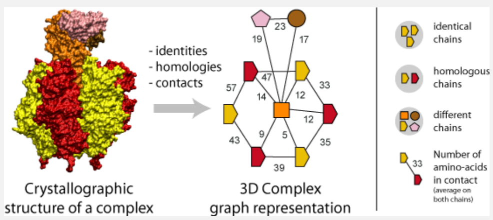
TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Graph Data

- Examples: Social network, protein interaction network, protein structure, criminal network

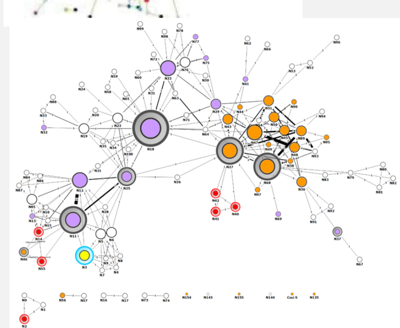






Crystallographic structure of a complex

3D Complex graph representation



Ordered Data

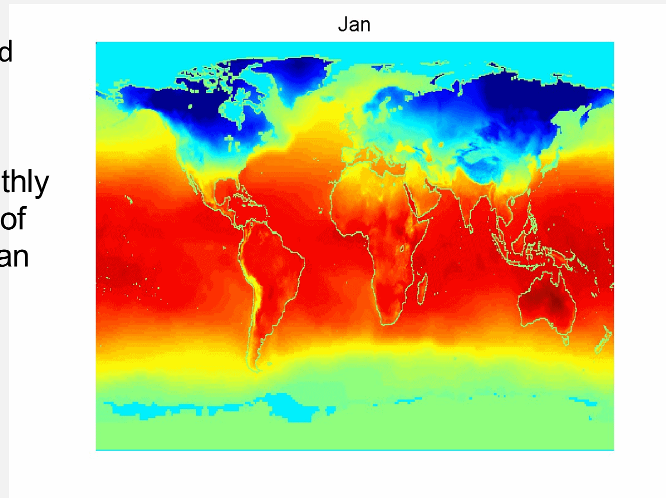
- Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

Ordered Data

- Spatially indexed temporal data

Average Monthly Temperature of land and ocean



Data challenges …

- Noise
- Wrong data
- Fake data
- Missing data
- Duplicate data

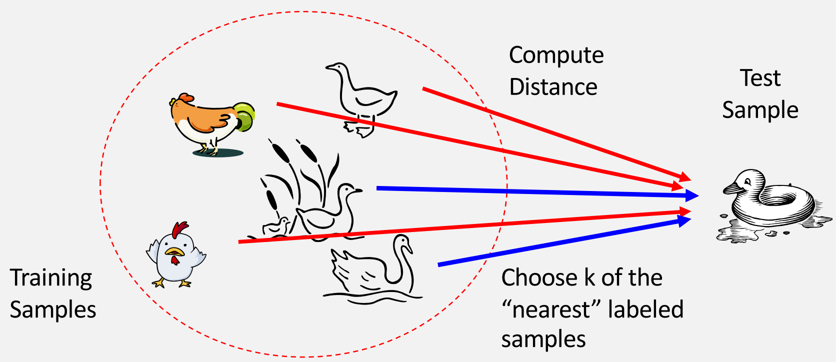
Machine Learning for Classification

Ingredients for classification

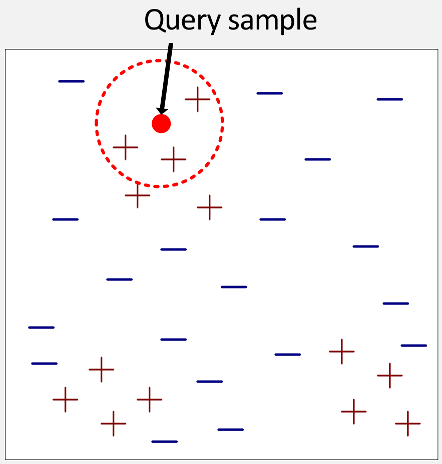
- Idea: Incorporate your knowledge of the problem into a learning system
- Sources of knowledge:
  - Feature representation
    - Crucial for the success of machine learning
    - Can be problem-specific
    - A good representation takes you half way
  - Training data
    - High quality labeled data can be hard to get
    - We may have to get by with the available data
    - Data may be biased for various reasons
  - Model training
    - No single learning algorithm outperforms all others on every task ("no free lunch")
    - Different algorithms have different inductive biases
    - Different algorithms make different assumptions

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



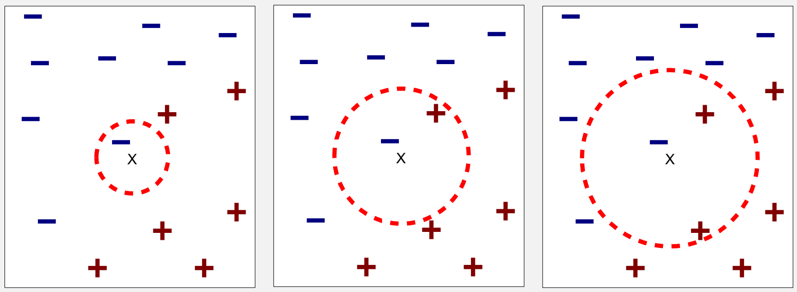
Nearest-Neighbor Classifiers



Require three things
- The set of stored training samples and their labels
- Distance Metric to compute distance between samples
- The value of *K*, the number of nearest neighbors to retrieve

- To classify a query sample:
  - Compute distance to training samples
  - Identify *K* nearest neighbors
  - Use class labels of the K nearest neighbors to determine the class label of the query sample (e.g., by taking majority vote)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a sample x are data points that have the k smallest distance to x

K nearest neighbor classifier

Data samples are assumed to lie in an n-dimensional space – e.g., the Euclidean space

An instance X is described by a feature vector

$$X_p=[x_{1p} \cdots x_{Np}]$$

Where x_{ip} denotes the value of the ith feature in X_p

$$d\left(X_p, X_r\right) = \left(\sum_{i=1}^{N} \left(x_{ip} - x_{ir}\right)^2 \right)$$

Defines the Euclidean distance between two points in the Euclidean space

Standardization

Standardization can be important when the variables are not all measured on the same scale

- 0-1 scaling

    4, 3, 1 2

  e.g. 3 → (3-min)/(max-min)=(3-1)/(4-1)=2/3

- Z-score scaling: subtract out the mean, divide by std. deviation

K nearest neighbor Classifier

<u>Learning Phase</u>

For each training example $(X_i, f(X_i))$, store the example in memory

<u>Classification phase</u>

Given a query instance X_q, identify the k nearest neighbors $X_1 \ldots X_k$ of X_q

Assign X_q the label of the majority class

$$g(X_q) = \underset{\omega}{argmax} \sum_{i=1}^{K} \delta(\omega, f(X_i))$$

where

$\delta(a,b) = 1$ iff $a = b$ and $\delta(a,b) = 1$.

Distance weighted K nearest neighbor Classifier

<u>Learning Phase</u>

For each training example $(X_i, f(X_i))$, store the example in memory

<u>Classification phase</u>

Given a query instance X_q, identify the k nearest neighbors $of\, X_q$ - $KNN(X_q) = \{X_1 \ldots X_k\}$

And obtain a weighted vote, with each nearest neighbor getting a vote in favor of its class label that is weighted by the distance to the query

$$w_i = \frac{1}{d(X_i, X_q)^2}$$

Distance Measures

- Distance
  - Depends on the data representation
  - Distance measure chosen

An Employee DB

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

Word Frequencies for Documents

	w1	w2	w3	w4	w5	w6
Doc1	0	4	0	0	0	2
Doc2	3	1	4	3	1	2
Doc3	3	0	0	0	3	0
Doc4	0	1	0	3	0	0
Doc5	2	2	2	3	1	4

Representation has to be chosen with some care
Distance measure should be chosen to work with the representation

Distance measures

- *d (p, q)* between two points *p* and *q* is a proper distance measure if it satisfies:

  1. Positive definiteness:

  d (p, q) $\geq$ 0 for all *p* and *q* and

  d (p, q) = 0 only if *p* = *q*.

  2. Symmetry: *d (p, q)* = *d (q, p)* for all *p* and *q*.

  3. Triangle Inequality:

  d (p, r) $\leq$ *d (p, q)* + *d (q, r)* for all points *p*, *q*, and *r*.

Cosine Distance

- If d_1 and d_2 are two document vectors, then

$$1\text{-}\cos(d_1, d_2) = 1\text{-}(d_1 \bullet d_2) / ||d_1|| \, ||d_2|| ,$$

where $\bullet$ indicates vector dot product and $|| d ||$ is the length of vector d.

- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**

d_2 = **1 0 0 0 0 0 0 1 0 2**

$d_1 \bullet d_2 = 3x1 + 2x0 + 0x0 + 5x0 + 0x0 + 0x0 + 0x0 + 2x1 + 0x0 + 0x2 = 5$

$||d_1|| = (3x3+2x2+0x0+5x5+0x0+0x0+0x0+2x2+0x0+0x0)^{0.5} = (42)^{0.5} = 6.481$

$||d_2|| = (6)^{0.5} = 2.245$

$\cos(d_1, d_2) = .3150$

Distance Measures

Distances in vector spaces

- Euclidean distance $\sqrt{\sum_{j=1}^{d}\left(p_j - q_j\right)^2}$
- Minkowski distance
  - a generalization of Euclidean distance
  - $\sqrt[n]{\sum_{j=1}^{d}\left|p_j - q_j\right|^n}$

Distance measures in Boolean spaces

- *n*=1 Manhattan distance
- n=2 Euclidean distance

Distance measures for data with nominal attributes

- Nominal attributes can take 2 or more values, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Simple matching – distance between two objects is simply the number of mismatched attributes divided by the total number of attributes
- One hot encoding – Encode each M-valued nominal attribute an M-bit vector

  Red: 1 0 0 0, Yellow: 0 1 0 0; Blue: 0 0 1 0 …
- Use distance measures designed for vectors …

Decision Boundary induced by the 1 nearest neighbor classifier form a Voronoi diagram


Manhattan distance


Euclidian distance

Query points in the polygon surrounding the training data point are closer to it than any other training data point

Image source: Wikipedia

P-spectrum similarity for sequences over an alphabet

- The p-spectrum of a string is the histogram – vector of number of occurrences of all possible contiguous substrings – of length p
- We can define a similarity function $K(s, t)$ over $\sum^* \times \sum^*$ as the inner product of the p-spectra of s and t.

$$s = statistics$$

$$t = computation$$

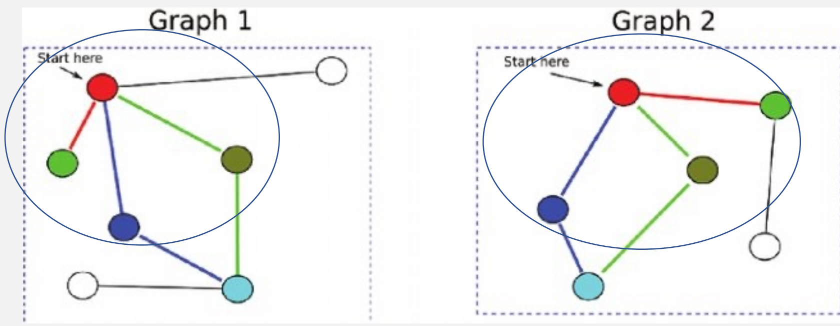
$$p = 3$$

$$\text{Common substrings}: \ tat, ati$$

$$K(s,t) = 2$$

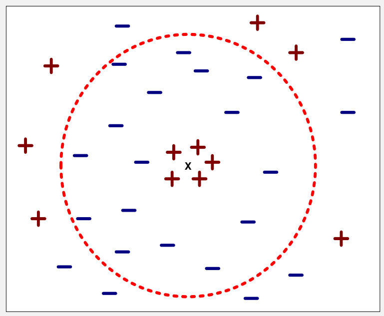
Can you think of a similarity function for graphs?

- Two graphs are similar if their *k*-hop neighborhoods are similar.



Nearest Neighbor Classification…

- Choosing the value of *k*:
  - If *k* is too small, the model can be sensitive to noise
  - If *k* is too large, neighborhood may include too many samples from other classes



Nearest neighbor classifiers

- Nearest neighbor classifiers are conceptually simple
- Learn by simply memorizing the training data
- The computational effort of learning is low
- The storage requirements of learning is high
  – need to memorize the training data
- Cost of classifying new instances can be high
  – Use efficient data structures and algorithms for nearest neighbor search, *k-d* trees, e.g., locality sensitive hashing
- A distance measure needs to be defined over the input space
- Performance degrades when
  - Dimensionality increases
  - The number of irrelevant attributes increases
  - The attributes are highly correlated

  – Need to perform feature selection or dimensionality reduction

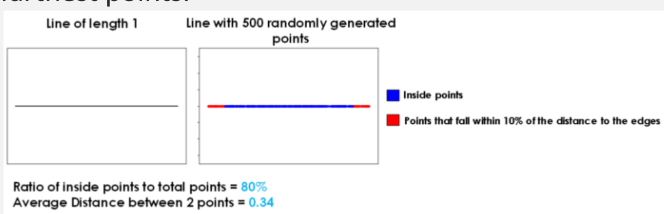
K-d Tree

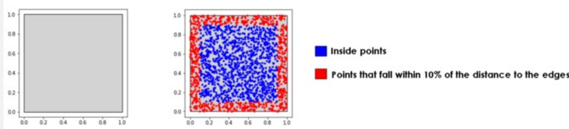
Given a pair $[S, v]$, where $S = x_1, \ldots, x_n$ is a set of points with $x_i \in \mathbb{R}^d$, corresponding to node v in a partially built *kd*-tree:

- if $n = 1$, then store that point in the node v. v will now be a leaf of the tree.
- Otherwise, pick a dimension $i \in \{1, \ldots d\}$ [there are many suggested heuristics: picking uniformly at random, choosing in round-robin order, choosing the dimension with largest variance, etc.]
- Let m be the median of the ith dimension of the points: $m = median[x_1(i), \ldots, x_n(i)]$. Store dimension i and median m at node v. Partition the set S into $S_<$ and $S_>$ according to whether the ith coordinate of each point exceeds m. [Note that in some implementations, "median" might be replaced by "mean" or some other value.]
- Make two children of $v_<$ and $v_>$, and recurse on $[v_<, S_<]$ and $[v_>, S_>]$.

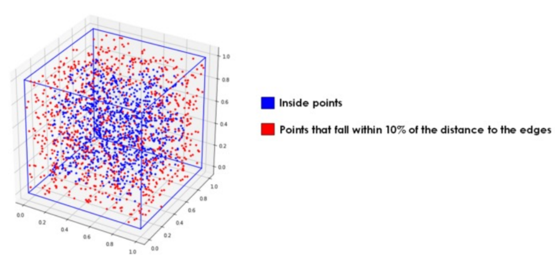
Similarity measures in high dimensions

- **As we increase the number of Dimensions, our data becomes more sparse** (the ”volume” of the space increases exponentially with the number of dimensions)
- As we increase the dimensions of our data, the average similarity between pairs of data points decreases.
- In the limit, the average similarity between the closest points approaches the average similarity between the farthest points.





Ratio of inside points to total points = 63%
Average Distance between 2 points = 0.52



Ratio of inside points to total points = 51%
Average Distance between 2 points = 0.65

Function approximation (Regression)

- Function approximation is like classification except the labels are real valued

Example applications:

Predicting

- Stock value
- Income
- Power consumption

K nearest neighbor Function Approximator

Learning Phase

For each training example $(X_i, f(X_i))$, store the example in memory

Approximation phase

Given a query instance X_q, identify the k nearest neighbors $X_1 \ldots X_k$ of X_q

$$g(X_q) \leftarrow \frac{\sum_{l=1}^{K} f(X_i)}{K}$$

Value of a function (e.g., price of a product) at a query point is simply the average or inverse distance weighted average of the value of the function at the k nearest neighbors of the query point

Lab: Nearest Neighbor Classifier

https://colab.research.google.com/drive/1m71GlCWdEovGlAhVxlwq68qrsIkYda6r?usp=sharing