CSE/ECE-471: Statistical Methods in AI **Spring 2020**

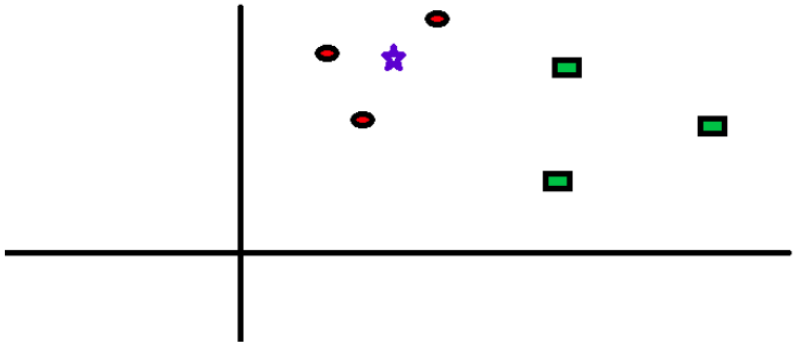
NEAREST NEIGHBOUR CLASSIFICATION

Prepared by: Himani Bhardwaj(2019201081), M.Pradeep Kumar(2019201055), Arshad Mohammed(2018900056)

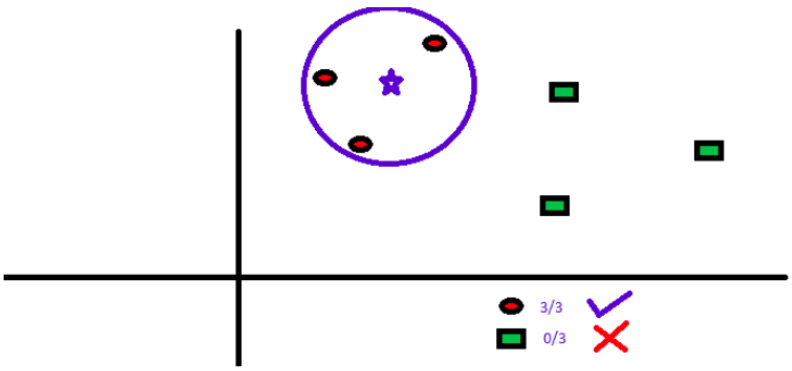
Do NOT just list out the facts and equations like they're given in the slides. Write them in an explanatory way so that others can learn from it (and for your own learning). Add figures where ever necessary. The teams will be scored accordingly.

In this note, we discuss ....

1 Some examples



Following is a spread of red circles (RC) and green squares (GS). You intend to find out the class of the blue star (BS) . BS can either be RC or GS and nothing else. The "K" is KNN algorithm is the nearest neighbors we wish to take vote from. Let's say K = 3. Hence, we will now make a circle with BS as center just as big as to enclose only three datapoints on the plane as shown:.



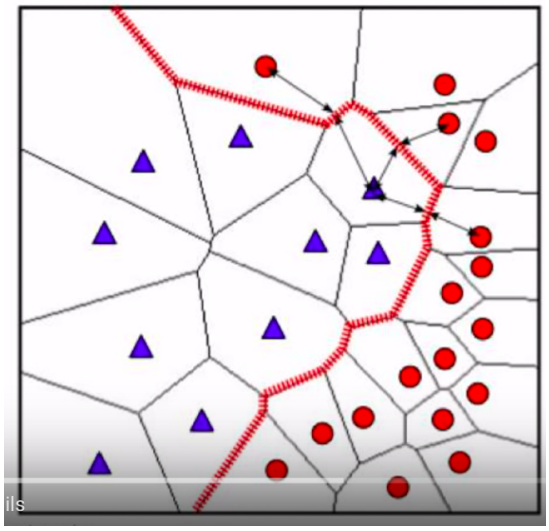
The three closest points to BS is all RC. Hence, with good confidence level we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC.

1.1 ASSUMPTION

1. Similar points have similar labels.
2. Algorithm needs to compute distance to every point.

1.2 Voronoi Tesselation

1. Partition the plane into regions.
2. Boundary consists of points that are at same distance from two different training samples



1.3 Decision Boundary

1. Think of K as controlling the shape of the decision boundary
2. An alternate way of understanding KNN is by thinking about it as calculating a decision boundary (i.e. boundaries for more than 2 classes) which is then used to classify new points.

1.4 Algo For KNN

1. It is defined according to a distance metric between two data points. A popular choice is the Euclidean distance given by

   D(x,x')=$\sqrt{\Sigma_i |x_i - x'_i|^2}$

but other measures can be more suitable for a given setting and include the Manhattan, Chebyshev and Hamming distance.

2. More formally, given a positive integer K, an unseen observation x and a similarity metric d, KNN classifier performs the following two steps:

   (a) It runs through the whole dataset computing d between x and each training observation. We'll call the K points in the training data that are closest to x the set A. Note that K is usually odd to prevent tie situations.

   (b) It then estimates the conditional probability for each class, that is, the fraction of points in A with that given class label. (Note I(x) is the indicator function which evaluates to 1 when the argument x is true and 0 otherwise) P(y=j—X=x)=1KiAI(y(i)=j)

1.5 Impact of value of K

1. When K is small, we are restraining the region of a given prediction and forcing our classifier to be "more blind" to the overall distribution.
2. A small value for K provides the most flexible fit, which will have low bias but high variance. Graphically, our decision boundary will be more jagged.
3. On the other hand, a higher K averages more voters in each prediction and hence is more resilient to outliers. Larger values of K will have smoother decision boundaries which means lower variance but increased bias.
4. Rule says K $< \sqrt{n}$

2 Distance Measures

It is considered as a key component of the kNN algorithm , as it governs the classification of points. We can pick any distance function and can arrive at different solution.

2.1 Types of Distance Measures

2.1.1 Euclidean

Here we take the distance between the training and testing attributes ,squaring them and added up.

$$D(x, x') = \sqrt{\Sigma_i |x_i - x'_i|^2} \tag{1}$$

- Suitable for numeric attributes
- symmetric,spherical and treats all dimensions equally.
- very sensitive to large difference in a single attribute.

Imagine if we take 100-1000 training attributes nearer to the test attribute and there is one point which is far away from the test attribute, then this point distance would effect the classification as it dominates the other distances.

2.1.2 Better version of euclidean

$$D(x, x') = \sqrt{\frac{\Sigma_i |x_i - x'_i|^2}{\sigma_i^2}} \tag{2}$$

2.1.3 Even-Better version of euclidean

$$D(x, x') = (X - X')^T \Sigma^{-1} (X - X') \tag{3}$$

2.1.4 Hamming Distance

Here we count the hamming distance ,which means if $x_i \neq x'_i$, then this will add up to the distance.So, it is the number of the attributes at which the test attribute differs.

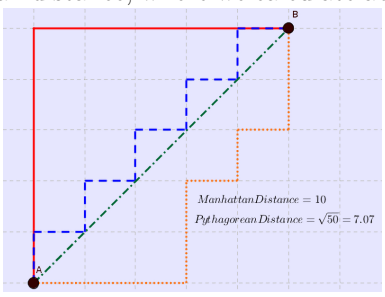
$$D(x, x') = \Sigma_i 1_{x_i \neq x'_i} \tag{4}$$

- number of attributes where x_i and x'_i differ.

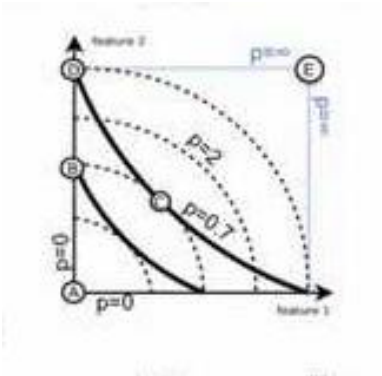
2.1.5 Minkowski Distance(p-norm)

It represents a family of distance measures, depending on the value of 'p'.

$$D(x, x') = \sqrt[p]{\Sigma_i |x_i - x'_i|^p} \tag{5}$$

- At p=2, this equation becomes euclidean distance.
- p=1, Manhattan distance, where we calculate absolute of the difference between the attributes and sum them up. 

- p=0, Hamming distance, cause if we decrease p such that it nears to zero, then $|x_i - x'_i|^p$ is zero iff $|x_i - x'_i|$ is zero, else it is one($0^0 \neq 0, 1^0 = 0$).
- p=∞, max($|x_i - x'_i|$), this can be arrived by analysing the limits of each term in the summation, the term with max value will reach infinite first, so the equation becomes as stated above, also called Chebychev distance.



In the above graph, we can see the effect on the distance with different values of p.X-axis and Y-axis represents the attributes on which our training points are dependent.A is a test point, and the dotted lines represent circles formed by training points, so they are at same distance from A.

- p=2, training points B and C are at the same distance from A and D becomes the far point.
- p=0.7, the dark line represent the points which are at same distance from A, So for this type of distance D and C are at same distance, but B is nearer with p=0.7.
- p=0, A,B,D are at the same distance from A, as they differ in feature2 but they don't differ in feature1.

2.1.6 Kullback-Leibler(KL) divergence

$$D(x, x') = -\Sigma_i x_i \log \frac{x_i}{x'_i} \tag{6}$$

- It is used for histograms ($x_i > 0, \Sigma_i x_i = 1$).
- asymmetric, excess bits to encode x with x'

With these there are some other custom distance measures like BM25 which are used for text training data.

3 Practical Issues of k-NN

3.1 Resolving Ties

If there are equal number of positive and negative nearest neighbours then there could be ties in classification.One naive way to remove these tiesis to pick odd number of nearest neighbours, But this doesn't work for multi-class attributes.

3.1.1 Breaking Ties

- random, flip a coin and decide positive/negative.
- prior, pick a class with greater prior(may be the dominant class).
- nearest, use 1-nn classifier to decide.

3.2 Missing values

- We have to Fill in, otherwise we can't compute the distances.
- This picked value should affect the distance as little as possible.
- reasonable choice would be to use average across entire data set.

4 Applications of k-NN

4.1 MNIST data

- It is a database of handwritten digits, used for pattern recognition, made by yann le cun.
- We can see the change of error rates according to the kNN and different distance measures in their site.

For more details, refer to this website http://yann.lecun.com/exdb/mnist/index.html.

4.2 IM2GPS

- Estimating geographic information from an image.
- Data set made from the image hosting site flickr.
- More info on it can be found here http://graphics.cs.cmu.edu/projects/im2gps/im2gps.pdf.

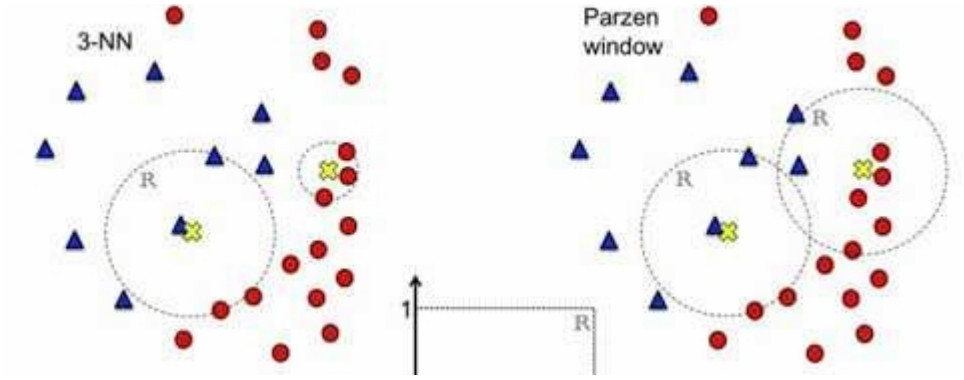
4.3 IM2TEXT

- Describing images using 1 million captioned photographs.
- Performing a huge number of Flickr queries and then filtering the noisy results down to 1 million images with associated visually relevant captions.
- more about it is available here https://papers.nips.cc/paper/4470-im2text-describing-images-using-1-million-captioned-photographs.pdf

5 Parzen windows and Kernels

5.1 Parzen Window

Take a constant window size of radius R ,take the nearest neighbours in that window and use them for classification.

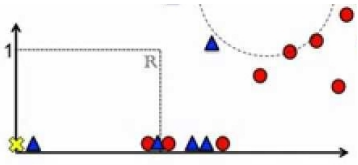


5.2 k-NN

Take a window of size R and check number of nearest neighbours and if not matched the criteria of 'k' then increase the radius the radius to cross the threshold number of neighbours.An example for k=3 is shown in the figure above.

5.3 Approach

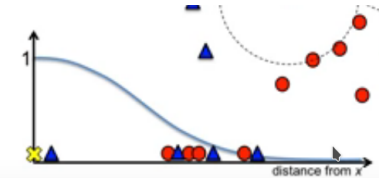
This can be used for both of the above ways.



- Go through all the training points and check whether it is present in the region or not.
- If present, then take y_i as 1 else no.
- This can be written mathematically as sgn($\Sigma_{i:x_i \in R(x)} y_i$).

Problem with the above approach is that when there is tie breaking, or if a point is nearer to the circle but out of it, there could be an error in classification.Here in this approach,the y_i is 1 when in the region and zero outside,to eliminate this we can use weighted y_i so that it depends on the distance from the center.

5.4 Modified Approach-Kernel



- In the above y_i is just either 1 or 0, here it depends on the bell curve shown above.
- So, here the there is no need of radius to pick, as now the point is considered according to weighted distance.
- The equation becomes $sgn(\Sigma_i y_i k(x_i, x))$,so here k($x_i$,x) is a kernel between x_i and x , which represents weighted distance between test point x and training points x_i.

6 Pros and Cons of k-NN

6.1 PROS

- Almost no assumptions about the data, except that nearby regions of space belong same class.
- Nothing to infer from the data, except for k and Distance.
- Easy to update in online setting, just add new item to training set.

6.2 CONS

- Computationally expensive, time- need to compute distance to all examples-O(nd), space- for storing all the attributes, making it expensive at testing.
- Need to handle missing data.
- Sensitive to class-outliers (An outlier will change the decision boundary).
- Sensitive to lots of noise or irrelevant attributes that affects distance.

7 Making k-NN fast

7.1 Ideas That Can Be Used

1. Reduce d- dimensionally reduction
2. Reduce n- don't compare with all training examples
3. Kd trees
4. locally sensitive hashing

7.2 Kd Trees

Say we have given data:
(1,9), (2,3), (4,1), (3,7),(3,4),(6,8),(7,9),(9,6)
Testing for data : (7,4)

1. First we will find median of whole data say along X- axis
2. On second level divide both left and right subarrays by finding median along second axis say Y-axis
3. Continue the procedure further.

- Each decision divides space int two equal partitions.
- Say data is uniformly distributed, we need to take d decisions.
- Kd works well with low dimensions

7.3 Higher Dimensional Data

- For lower dimensions we use k-d trees ,but for higher dimensional data we use Locally sensitive Hashing method(LSH).
- We follow same approach of dividing regions, but into k hyper planes, slicing into 2^k regions.
- Out of all regions we pick one ,and take the points in those region for learning.
- Complexity: O(kd+$\frac{dn}{2^k}$.
- We need to maintain the hash table of the points for each hyper plane.

7.4 Inverted Index

When data is very sparse and high dimensional like text, where very word is an attribute and many attributes are zeros. Example : emails and search engines.

- Maintain the list of sentences in an email and label them whether spam or not.
- prepare an inverted list of the the specific words and search in those sentences which are there in inverted list.

8 Curse of Dimensionality

kNN classifier makes the assumption that similar points share similar labels. Unfortunately, in higher dimensional spaces, points that are drawn froma probability distribution , tend to never be close together. lets us pick a region of shape cube with side length of 'l' and if we need k points out of N, then this length 'l' is approximately proportional to $((\frac{k}{N})^{\frac{1}{d}}$, where d is number of dimensions). So ,as the dimensions increases the number of points to compare will increase. Remedy for this is to use the manifold of the higher dimensional data for k-NN.

References

[1] https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

[2] https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

[3] https://papers.nips.cc/paper/4470-im2text-describing-images-using-1-million-captioned-photographs.pdf

[4] https://papers.nips.cc/paper/4470-im2text-describing-images-using-1-million-captioned-photographs.pdf http://graphics.cs.cmu.edu/projects/im2gps/im2gps.pdf