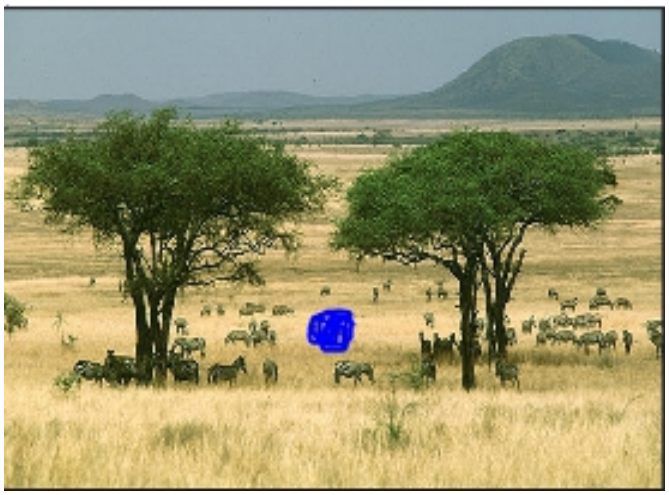
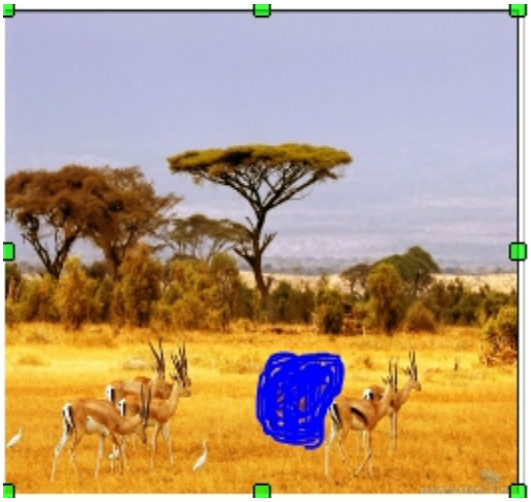
Chapter – 05 Instance based Classification: kNN

Instance based method is one of the non-parametric classification technique. We make numerous decisions each and every day and more often than not they are almost always based on the particular problem and observations we are presented with. Even though we might have an aggregate opinion, that particular decision is based on the data we observe in the immediate without challenging long held beliefs. The immediate percept dominates the decision making process. Thus the instance based process is non parametric because we do not make any assumption about the data, its distribution and hence there are no population parameters that we try to infer using any sample statistic. We do not tend to leverage a model, instead we draw upon past experience under similar situations and try to repeat what we had done before under similar situations. In that sense that model of learning, Jiawei Han et. al. Characterize as ***lazy learners*** in contrast to other models we have studied, where in the learner constructs a model and new scenarios are evaluated with that model, and therefore called ***eager learners***. All parametric models are by design eager learners. But not all instance based (non-parametric) are lazy learners. We will study few eager non-parametric learners later in our journey.

Nearest Neighbor technique is one such non-parametric lazy learner technique. The intuition behind NN is best illustarted using simple heuristics. Imagine yourself in the Yosemite valley. You hear people having animated conversation. You generally avoid trouble and are you likely to assume that those people you heard were trouble makers or other passionate nature lovers, like yourself? Most people would disregard the acrimonious debate and assume them to be fellow hikers. We have heard the old adage "*birds of a feather flock together*" or "*tell me who your friends are and I will tell you who you are.".* Our brain can effortlessly separate like things from unlike things.

Let us take a trip to Africa (photographs from internet)





Animals of a kind, hang out together. This is a natural order. Assume we are on a trek on that distant hill. Someone in the group asks hey what is that animal over there? Look at the above pics from the internet, I have blued out a few, can you guess what that animal might be? If you tell me 2 or 3 animals around that animal, I can guess what animal that is? This is the essential idea behind nearest neighbor algorithms, designated kNN. Even a grade school human brain will score high on this test. Why is it innate? How does the brain know what is similar and what is not? How does it settle on a particular k before making a determination? The k specifies the number of neighbors considered by the algorithm to assign a class to the unknown observation. How does the brain perform this task? Algorithms implement using standard similarity algorithms. Please refer to any basic introduction to Similarity and I would recommend https://aiaspirant.com/distance-similarity-measures-in-machine-learning/ for an easy reading.

How does kNN work?

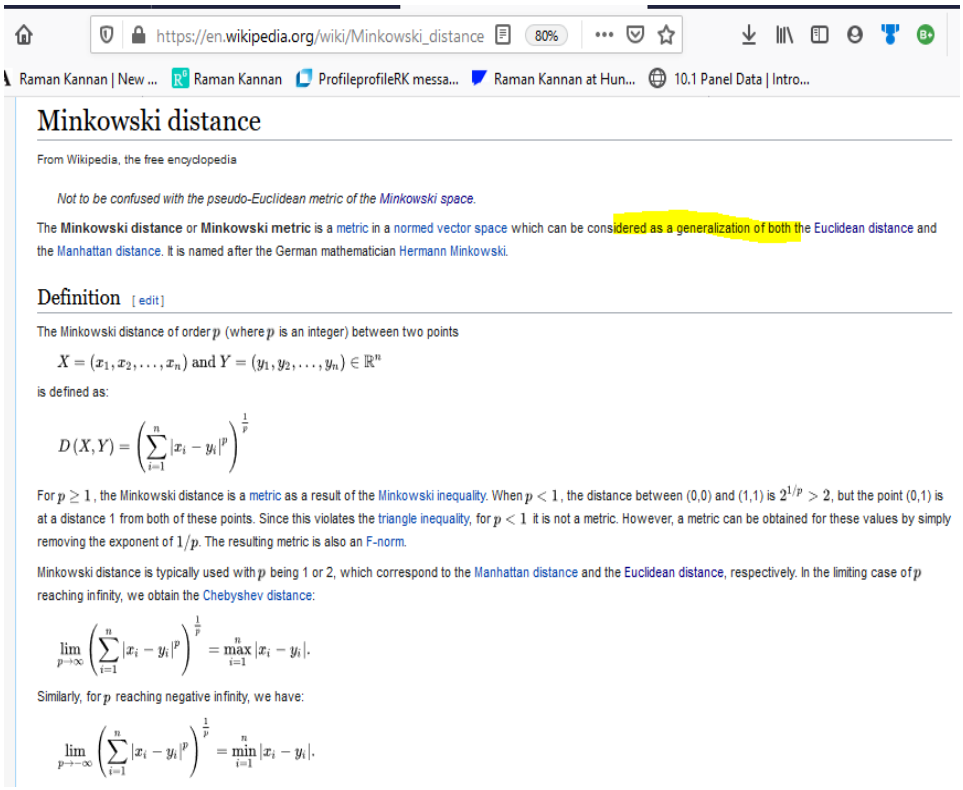
kNN computes the distance between the given instance and every other instance seen in the past, (aka

dataset). kNN then finds the k nearest neighbors, using that distance metric and assigns the majority class (most frequent class of the k nearest neighbors) to the given instance. k is usually 3,5 or some odd number to avoid tie. Thus, by definition there is no training/learning or induction phase in kNN. Given any new instance, kNN iterates over all known instance to compute the distance, finds the k nearest neighbors and assigns the majority class to the new instance. And the keyword here is distance. What is distance between giraffe and elephant. Without any format education how does the human brain compute this distance. All mammalian brains appear to compute if we judge how they tend to remain in groups(aggregation) or pick out a target (discrimination). Our challenge, now then, is to impart this innate technique to a automaton, without knowing how the brain does what it does.

I prefer the concise definition of kNN classifier is from Ethem Alpaydin "*The k-nn classifier assigns the input to the class having most examples among the k neighbors of the input* – all neighbors have equal vote and the class having the maximum number of voters among the k-neighbors is chosen and *the k-nearest neighbor classifier assigns an instance to the class most heavily represented among its neighbors*". The two boundary conditions of k (k=1, the nearest neighbor having the deciding vote) and (k=n, every instance in the known dataset) are interesting. What impact does it have on the variance and bias?

Central to k-NN is the notion of distance? How do we compute two distinct things are similar or dis-similar? Recommendation systems, Clustering (unsupervised learning) besides classification algorithms use this idea of distance. When the variates are orthogonal (change in one dimension has no impact on

any other dimension) and with identical scale an appropriate version of ***Minkowski distance*** may be used. *Euclidean distance in 2-D* and the so called *Manhattan distance in 1-D* are special cases of Mikowski.



https://en.wikipedia.org/wiki/Minkowski_distance 80%

Raman Kannan | New ... Raman Kannan ProfileprofileRK messa... Raman Kannan at Hun... 10.1 Panel Data | Intro...

Minkowski distance

From Wikipedia, the free encyclopedia

Not to be confused with the pseudo-Euclidean metric of the Minkowski space.

The **Minkowski distance** or **Minkowski metric** is a metric in a normed vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance. It is named after the German mathematician Hermann Minkowski.

Definition [edit]

The Minkowski distance of order p (where p is an integer) between two points

$$X = (x_1, x_2, \ldots, x_n) \text{ and } Y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$$

is defined as:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

For $p \geq 1$, the Minkowski distance is a metric as a result of the Minkowski inequality. When $p < 1$, the distance between (0,0) and (1,1) is $2^{1/p} > 2$, but the point (0,1) is at a distance 1 from both of these points. Since this violates the triangle inequality, for $p < 1$ it is not a metric. However, a metric can be obtained for these values by simply removing the exponent of $1/p$. The resulting metric is also an F-norm.

Minkowski distance is typically used with p being 1 or 2, which correspond to the Manhattan distance and the Euclidean distance, respectively. In the limiting case of p reaching infinity, we obtain the Chebyshev distance:

$$\lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = \max_{i=1}^{n} |x_i - y_i|.$$

Similarly, for p reaching negative infinity, we have:

$$\lim_{p \to -\infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = \min_{i=1}^{n} |x_i - y_i|.$$

When orthogonality and scale equivalence cannot be assumed, one must use ***Mahalanobis distance*** a multivariate formulation incorporating the covariance matrix of the features.

Definition and properties [edit]

The Mahalanobis distance of an observation $\vec{x} = (x_1, x_2, x_3, \ldots, x_N)^T$ from a set of observations with mean $\vec{\mu} = (\mu_1, \mu_2, \mu_3, \ldots, \mu_N)^T$ and covariance matrix S is defined as:[2]

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}.$$

Mahalanobis distance (or "generalized squared interpoint distance" for its squared value[3]) can also be defined as a dissimilarity measure between two random vectors $\vec{x}$ and $\vec{y}$ of the same distribution with the covariance matrix S:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}.$$

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. If the covariance matrix is diagonal, then the resulting distance measure is called a *standardized Euclidean distance*:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}},$$

where s_i is the standard deviation of the x_i and y_i over the sample set.

Lazy Learners and Induction Phase

Given a new (never seen before) observation o, kNN,

compute the distance between o and each data d_i in the dataset.

Sorts and selects the first k instances,

determines the majority class in those k nearest neighbors

and assigns the majority class to the new observation o

as described there is no transfer of knowledge from the given dataset or prior distance calculations.

Each new observation is processed exactly as outlined.

In comparison, parametric methods, require the parameters to classify new observations and the parameters are estimated during the learning phase. Whereas the kNN must carry all the known observations to classify new observation. Everything happens at the time of classifying .

In the RMD we present an implementation and comparison with standard implementation of kNN.

There is no need to split the data into training/testing sets because we are not estimating parameters.

In practice I do split it so that I can determine the performance of the algorithm when presented with new data. For kNN, the test set serves this purpose. Before the business generates a new observation we can validate the algorithm for performance.