God, Your Book Is Great !!

Just another WordPress.com weblog

Feeds: Posts Comments

A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm

May 17, 2010 by Saravanan Thirumuruganathan

K Nearest Neighbor (KNN from now on) is one of those algorithms that are very simple to understand but works incredibly well in practice. Also it is surprisingly versatile and its applications range from vision to proteins to computational geometry to graphs and so on . Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple. It also might surprise many to know that KNN is one of the **top 10 data mining algorithms** (http://www.cs.umd.edu/~samir/498/10Algorithms-08.pdf). Lets see why this is the case !

In this post, I will talk about KNN and how to apply it in various scenarios. I will focus primarily on classification even though it can also be used in regression). I also will not discuss much about **Voronoi diagram** (http://en.wikipedia.org/wiki/Voronoi_diagram) or **tessellation** (http://en.wikipedia.org/wiki/Tessellation).

KNN Introduction

KNN is an *non parametric lazy learning* algorithm. That is a pretty concise statement. When you say a technique is non parametric , it means that it does not make any assumptions on the underlying data distribution. This is pretty useful , as in the real world , most of the practical data does not obey the typical theoretical assumptions made (eg gaussian mixtures, linearly separable etc) . Non parametric algorithms like KNN come to the rescue here.

It is also a lazy algorithm. What this means is that it does not use the training data points to do any *generalization*. In other words, there is *no explicit training phase* or it is very minimal. This means the training phase is pretty fast . Lack of generalization means that KNN keeps all the training data. More exactly, all the training data is needed during the testing phase. (Well this is an exaggeration, but not far from truth). This is in contrast to other techniques like SVM where you can discard all non support vectors without any problem. Most of the lazy algorithms – especially KNN – makes decision based on the entire training data set (in the best case a subset of them).

The dichotomy is pretty obvious here – There is a non existent or minimal training phase but a costly testing phase. The cost is in terms of both time and memory. More time might be needed as in the worst case, all data points might take point in decision. More memory is needed as we need to store all training data.

Assumptions in KNN

Before using KNN, let us revisit some of the assumptions in KNN.

KNN assumes that the data is in a *feature space*. More exactly, the data points are in a metric space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of distance – This need not necessarily be Euclidean distance although it is the one commonly used.

Each of the training data consists of a set of vectors and class label associated with each vector. In the simplest case , it will be either + or – (for positive or negative classes). But KNN , can work equally well with arbitrary number of classes.

We are also given a single number "k" . This number decides how many neighbors (where neighbors is defined based on the distance metric) influence the classification. This is usually a odd number if the number of classes is 2. If k=1 , then the algorithm is simply called the nearest neighbor algorithm.

KNN for Density Estimation

Although classification remains the primary application of KNN, we can use it to do density estimation also. Since KNN is non parametric, it can do estimation for arbitrary distributions. The idea is very similar to use of **Parzen window** (http://en.wikipedia.org/wiki/Parzen_window) . Instead of using hypercube and kernel functions, here we do the estimation as follows – For estimating the density at a point x, place a hypercube centered at x and keep increasing its size till k neighbors are captured. Now estimate the density using the formula,

$$p(x) = \frac{k/n}{V}$$

Where n is the total number of V is the volume of the hypercube. Notice that the numerator is essentially a constant and the density is influenced by the volume. The intuition is this : Lets say density at x is very high. Now, we can find k points near x very quickly . These points are also very close to x (by definition of high density). This means the volume of hypercube is small and the resultant density is high. Lets say the density around x is very low. Then the volume of the hypercube needed to encompass k nearest neighbors is large and consequently, the ratio is low.

The volume performs a job similar to the bandwidth parameter in kernel density estimation. In fact , KNN is one of common methods to estimate the bandwidth (eg adaptive mean shift) .

KNN for Classification

Lets see how to use KNN for classification. In this case, we are given some data points for training and also a new unlabelled data for testing. Our aim is to find the class label for the new point. The algorithm has different behavior based on k.

Case 1 : k = 1 or Nearest Neighbor Rule

This is the simplest scenario. Let x be the point to be labeled . Find the point closest to x . Let it be y. Now nearest neighbor rule asks to assign the label of y to x. This seems too simplistic and some times even counter intuitive. If you feel that this procedure will result a huge error , you are right – but there is a catch. This reasoning holds only when the number of data points is not very large.

If the number of data points is very large, then there is a very high chance that label of x and y are same. An example might help – Lets say you have a (potentially) biased coin. You toss it for 1 million time and you have got head 900,000 times. Then most likely your next call will be head. We can use a similar argument here.

Let me try an informal argument here - Assume all points are in a D dimensional plane . The number of points is reasonably large. This means that the density of the plane at any point is fairly high. In other words , within any subspace there is adequate number of points. Consider a point x in the subspace which also has a lot of neighbors. Now let y be the nearest neighbor. If x and y are sufficiently close, then we can assume that probability that x and y belong to same class is fairly same – Then by decision theory, x and y have the same class.

The book "Pattern Classification" by Duda and Hart has an excellent discussion about this Nearest Neighbor rule. One of their striking results is to obtain a fairly tight error bound to the Nearest Neighbor rule. The bound is

$$P^* \le P \le P^*(2 - \frac{c}{c-1}P^*)$$

Where P^* is the Bayes error rate, c is the number of classes and P is the error rate of Nearest Neighbor. The result is indeed very striking (atleast to me) because it says that if the number of points is fairly large then the error rate of Nearest Neighbor is less that twice the Bayes error rate. Pretty cool for a simple algorithm like KNN. Do read the book for all the juicy details.

Case 2 : k = K or k-Nearest Neighbor Rule

This is a straightforward extension of 1NN. Basically what we do is that we try to find the k nearest neighbor and do a majority voting. Typically k is odd when the number of classes is 2. Lets say k = 5 and there are 3 instances of C1 and 2 instances of C2. In this case , KNN says that new point has to labeled as C1 as it forms the majority. We follow a similar argument when there are multiple classes.

One of the straight forward extension is not to give 1 vote to all the neighbors. A very common thing to do is *weighted kNN* where each point has a weight which is typically calculated using its distance. For eg under inverse distance weighting, each point has a weight equal to the inverse of its distance to the point to be classified. This means that neighboring points have a higher vote than the farther points.

It is quite obvious that the accuracy *might* increase when you increase k but the computation cost also increases.

Some Basic Observations

1. If we assume that the points are d-dimensional, then the straight forward implementation of finding k Nearest Neighbor takes O(dn) time.
2. We can think of KNN in two ways – One way is that KNN tries to estimate the posterior probability of the point to be labeled (and apply bayesian decision theory based on the posterior probability). An alternate way is that KNN calculates the decision surface (either implicitly or explicitly) and then uses it to decide on the class of the new points.
3. There are many possible ways to apply weights for KNN – One popular example is the Shephard's method.
4. Even though the naive method takes O(dn) time, it is very hard to do better unless we make some other assumptions. There are some efficient data structures like **KD-Tree** (http://en.wikipedia.org/wiki/Kd_tree) which can reduce the time complexity but they do it at the cost of increased training time and complexity.
5. In KNN, k is usually chosen as an odd number if the number of classes is 2.
6. Choice of k is very critical – A small value of k means that noise will have a higher influence on the result. A large value make it computationally expensive and kinda defeats the basic philosophy behind KNN (that points that are near might have similar densities or classes) .A simple approach to select k is set

$k = \sqrt{n}$

7. There are some interesting data structures and algorithms when you apply KNN on graphs – See **Euclidean minimum spanning tree** (http://en.wikipedia.org/wiki/Euclidean_minimum_spanning_tree) and **Nearest neighbor graph** (http://en.wikipedia.org/wiki/Nearest_neighbor_graph) .

8. There are also some nice techniques like condensing, search tree and partial distance that try to reduce the time taken to find the k nearest neighbor. Duda et al has a discussion of all these techniques.

Applications of KNN

KNN is a versatile algorithm and is used in a huge number of fields. Let us take a look at few uncommon and non trivial applications.

1. Nearest Neighbor based Content Retrieval
This is one the fascinating applications of KNN – Basically we can use it in Computer Vision for many cases – You can consider handwriting detection as a rudimentary nearest neighbor problem. The problem becomes more fascinating if the content is a video – given a video find the video closest to the query from the database – Although this looks abstract, it has lot of practical applications – Eg : Consider **ASL** (http://en.wikipedia.org/wiki/Asl) (American Sign Language) . Here the communication is done using hand gestures.

So lets say if we want to prepare a dictionary for ASL so that user can query it doing a gesture. Now the problem reduces to find the (possibly k) closest gesture(s) stored in the database and show to user. In its heart it is nothing but a KNN problem. One of the professors from my dept , Vassilis Athitsos , does research in this interesting topic – See **Nearest Neighbor Retrieval and Classification** (http://vlm1.uta.edu/~athitsos/nearest_neighbors/) for more details.

2. Gene Expression
This is another cool area where many a time, KNN performs better than other state of the art techniques . In fact a combination of KNN-SVM is one of the most popular techniques there. This is a huge topic on its own and hence I will refrain from talking much more about it.

3. Protein-Protein interaction and 3D structure prediction
Graph based KNN is used in protein interaction prediction. Similarly KNN is used in structure prediction.

References

There are lot of excellent references for KNN. Probably, the finest is the book "Pattern Classification" by Duda and Hart. Most of the other references are application specific. Computational Geometry has some elegant algorithms for KNN range searching. Bioinformatics and Proteomics also has lot of topical references.

After this post, I hope you had a better appreciation of KNN !

(http://del.icio.us/post?v=4&noui&url=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/&title=A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm) (http://digg.com/submit?phase=2&url=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/&title=A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm) (http://www.facebook.com/sharer.php?u=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/&t=A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm) (http://www.google.com/bookmarks/mark?op=add&bkmk=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/&title=A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm) (http://reddit.com/submit?url=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/&title=A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm) (http://www.stumbleupon.com/submit?url=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/&title=A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm) (http://technorati.com/faves/?add=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/) (http://twitter.com/home?status=https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/)

If you liked this post , please subscribe to the ***RSS feed*** (http://feeds.feedburner.com/GodYourBookIsGreat).

Posted in Data Mining | Tagged bioinformatics, classification, genomics, knn | 86 Comments

86 Responses

Kripa K S *on May 18, 2010 at 7:58 am* | Reply

Excellent post!

The Bayes error rate metric is interesting! Does a neighbor's error rate influence the classification of a point and hence its error rate?

Since the set is a feature set, noisy irrelevant features could degrade the kNN's performance. And I read that you could smooth the features using evolutionary algorithms.
Whereas, I have worked on multi-objective EA (NSGA2, SPEA) and kNN is a favorite choice to cluster fit alleles and promote them to a next generation. Something like mutually-beneficial!

Needless to say you are becoming prolific by every post! 🙂

Saravanan Thirumuruganathan *on May 19, 2010 at 12:57 am* | Reply

Thanks Kripa ! Yes KNN is frequently used along with genetic algorithms and this technique is especially common in proteomics

Rama Rao *on August 21, 2013 at 9:31 am*

Hi,

This is excellent information on KNN.

Could u suggest me , how would be the combination of Naive Bayes (at first stage) and KNN (at second stage). This is in the case of application of Hydrid Machine Learning tech.to network traffic to classify trusted and untrusted packets.

What is ur suggestion. KNN should be used in first or second stage.Pls suggest.

Pls also suggest the best hybrid technique in combination with KNN.

myp05 *on October 27, 2010 at 8:49 pm* | Reply

thanks for the information. Actually, i have a question. What will you do if you have 3 classes. It's highly possible you will run into the situation that there does not exist such a dominating type right?

Saravanan Thirumuruganathan *on October 27, 2010 at 10:56 pm* | Reply

Hi ,

That is correct . It is possible that you will run into cases where the Pr(X|class1) is approximately same as the probability of P(X|class2), P(X|class3) etc. There are a few solutions. One of them is to assign different weights to each distance instead of giving them uniform weights. So a point that is closer to test point will get more weight that the point that is farther away. It is still possible to have a degenerate case where the weighted distances are approximately equal in which case we can decide to randomly allocate it to one of the classes.

myp05 *on November 1, 2010 at 9:58 am*

Hi,

The last method by using randomness is exactly what I did. And I think it's also reasonable to use weight, but I am not so comfortable about choosing it wisely. I was thinking about reduce the number of neighbors, and compare it again, which kind of mimic the idea of weighting.

Saravanan Thirumuruganathan *on November 2, 2010 at 12:35 pm*

Hi,

You can use the inverse of the distance as a weight. So this means that the closer points will have higher weight and farther points have lesser weight. This is slightly more principled approach that giving arbitrary weights.

BDO *on November 21, 2010 at 11:58 pm* | *Reply*

Hi,

Thanks for the great introduction to KNN! I'm giving a presentation on the topic for class and this was an excellent reference.

Feng Yi *on February 23, 2011 at 1:47 am* | *Reply*

hi~
"Choice of k is very critical", can u introduce me several papers about how to select k? Thank u

Saravanan Thirumuruganathan *on February 23, 2011 at 8:38 am* | *Reply*

Hello Feng Yi,

Selecting k is a critical and I think is still a research area. There are some basic tips though :

1.If #classes = 2, then select k as odd to avoid ties.
2. Use some technique like cross validation. If you increase the number of folds and compare it with the results, that might give some hint on it.
3. A similar idea is to use elbow method that is used for k-means clustering.

I am have not read much about choice of k so your best bet might be to ask in some forums like meta optimize.

Feng Yi *on February 24, 2011 at 3:08 am*

thanks~

James C. Bennett *on February 28, 2011 at 4:18 am* | *Reply*

Hey Sarvanan,

Great work on the explanation here!

Is it possible for you to provide me with a C++ code that does the K-nearest neighbour algorithm for a 3-D point data cloud?

Any help, including links, would be sincerely appreciated!

Thanks in advance,

JCB.

Saravanan Thirumuruganathan *on March 7, 2011 at 8:37 am* | *Reply*

James,

You can check Antonio Gulli's code at http://codingplayground.blogspot.com/2010/01/nearest-neighbour-on-kd-tree-in-c-and.html . This uses kd-tree internally though.

neha maurya *on March 7, 2011 at 6:13 am* | *Reply*

thanks for explaining the concept of knn so nicely…..

Saravanan Thirumuruganathan *on March 7, 2011 at 8:38 am* | *Reply*

Thanks Neha!

Harini Sridharan *on March 22, 2011 at 11:47 pm* | *Reply*

Hi,
I was trying to do a matlab implementation of K-NN density estimation for 1D data. Since I have a 1-D hypercube, my V is just the distance of the kth point.
I can see that for those points whose kth neighbor is very close the estimated densities exceed 1. What should I be doing?

By the way, it is a great introductory material.

Thank You
Harini

Saravanan Thirumuruganathan *on March 23, 2011 at 1:05 am* | Reply

[Adding to blog for other readers]

Hi Harini,

I am not sure if I understood your question correctly – From my understanding, you have a set of points and given a point , you want to estimate the density at that point.

I am not sure how you will get a value > 1 for this case. Lets say training set is -3, -2, 1, 2 and 3 . Lets also assume you want to find density at point 0.

1) Let k =1. The nearest point here is 1 at a distance of 1. But since this is a half-line centered at 0 and extends to 1 on one side, it becomes a line segment from -1 to 1 with length 2. So the density is (1/5) / 2 which is 0.1 . ie you keep checking for new points between , say, -0.1 to 0.1, -0.2 to 0.2 and so on. At (-1,1), you got a point. But now the length is 2.
2) Let k =2. The nearest point is at 1. There are equally nearby points at -2 and 2. So the density is (3/5) / 4 or 3/20. Or if you just use the formula (k/n) / V, it is (2/5) / 4 which is 0.1 .

I "think", not considering the "entire" length could be your problem.

Again, it is well known that the output of knn for small k and n is not really a valid density as it typically diverges or plain discontinuous. If you need any other info, let me know.

Hope this helps !

Anonymous *on April 30, 2011 at 12:23 pm*

HI. can you tell me how to filter genes by using knn algorithm in R language and how to use different k values in same programe.?

I have 214 genes (positive and negative) and i have to filter those according to their p-values. Which i already did by using t-test. Now i want to select biomarkers using knn.

Saravanan Thirumuruganathan *on May 1, 2011 at 8:02 pm*

Hello,

I do not have much expertise on biomarkers. Your best bet will be to check the net for other publications which can shed some light on it.

Ravi *on May 20, 2011 at 11:36 am* | Reply

Hi this was a great explanation of KNN. However, I got a little lost after the beginning. I was wondering if you could recommend some books to understand the fundamentals with regards to intelligent algorithms.

To be more specific, when you talk about metric space, scalars and vectors, I get a little lost. I do have a mathematical understanding of these scalars/vectors matrices etc… But I fail at understanding how they apply to intelligent algorithms.

So if you do know of a good book or two that teaches the fundamentals of these concepts, it would be really helpful, as I'm very keen on the subject of text classifying.

Saravanan Thirumuruganathan *on May 22, 2011 at 9:14 pm* | Reply

Ravi,

There are lot of excellent books for understanding these data mining algorithms – my favorite is the book by Duda and Hart (Pattern Recognition) and by Christopher Bishop (Pattern Recognition and Machine Learning). Both of them (Especially PRML) have initial chapters that discuss the math preliminaries.

Ravi *on May 22, 2011 at 9:34 pm*

I will definitely get a copy of one of these books come summer time. Thank you so much for your kind help. 🙂

Emily Tabanao *on June 30, 2011 at 4:43 am* | Reply

Nicely done! Do you have any sample datasets for demonstration in a class?

Emily Tabanao

Saravanan Thirumuruganathan *on June 30, 2011 at 7:10 pm* | Reply

Emily,

I used the MovieLens dataset to perform knn based item recommendation. I do not have the code but the dataset can be downloaded from http://www.grouplens.org/node/73 .

Njideka Mbeledogu *on June 30, 2011 at 9:21 am* | Reply

Good day. Thanks for the great work you did. It's been very helpful. Pls, I am thinking of doing a comparative work using k-means model and k-NN model in recognizing and classifying stock patterns. How do you see it?

Njide

Saravanan Thirumuruganathan *on June 30, 2011 at 7:28 pm* | *Reply*

Njideka,

You will need the past time series data of the stock. Then you can use the current stock price to find the k nearest values in the time series. But instead of directly using them you will use their successors in the time series as the estimate. Of course you can use weighted data to smooth the variations because voting directly does not work (unless you just want to know if it goes up or down).

It is not clear to me how to use kmeans for stock prediction though.

gorave *on July 1, 2011 at 12:12 pm* | *Reply*

hey…i need code for video retrieval n classification in matlab or java…can any1 help me..????????

sagee *on July 21, 2011 at 7:24 pm* | *Reply*

Data ID Y X
1 5 3.5
2 7 4.1
3 8 6
4 2 2.1
5 3 0.8
Please assume Y is numerical and please use 5-fold cross-validation to find the optimal K for a K-nearest neighbor regression. Show your cross-validated residual sum of squares vs. each k.

Does anyone have idea how to solve this by using KNN ?

Saravanan Thirumuruganathan *on July 25, 2011 at 6:43 pm* | *Reply*

@sagee,

Sounds like an assignment problem to me. Here is one possible way : Given any point , k can vary between 1 to 4. If you take one out, it varies between 1 and 3. So , remove first point. For the remaining 4 points, vary K between 1 and 3 and check which gives the best result in terms of RMSE or any other error metric. Now remove second point and repeat same experiment for points 1,3,4,5 . Rinse and repeat !

said *on July 27, 2011 at 9:13 am* | *Reply*

Hi
can anybody help me in classification in 'R' using knn algorithm

Saravanan Thirumuruganathan *on August 8, 2011 at 6:36 pm* | *Reply*

Said,

Sorry for the delay 🙂 Takes a look at http://stat.ethz.ch/R-manual/R-devel/library/class/html/knn.html .

said *on September 28, 2011 at 6:30 am* | *Reply*

thank you very much for help

sagee *on July 27, 2011 at 5:47 pm* | *Reply*

HI, THanks for your reply, very helped full. I am doing a Data mining project so for that I am reviewing classification . if I have more questions, I will post in your blog :D.

https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/#comment-form-guest

Anonymous *on August 5, 2011 at 3:41 am* | *Reply*

Thanks sir,

Can we use KNN algorithm for finding a closest node in MANET or mobile network

Murali *on August 5, 2011 at 3:42 am* | *Reply*

Thanks for ur explanation. Can we use KNN algo. for finding a closest node in MANET

Saravanan Thirumuruganathan *on August 8, 2011 at 5:48 pm* | *Reply*

Murali,

I have no idea about networks …But searching for "nearest neighbor manet" in Google does give a few papers . Hopefully they might be useful to you !

james on August 26, 2011 at 3:15 pm | Reply

hi,

i am a student and i would like to use kNN for my thesis. i have some problems in visualizing on what happens during the process. can you provide me some graphs, figures, or any picture about what would happen during the process?.. thanks..

btw, your explanations was great though i get lost sometimes 🙂

Saravanan Thirumuruganathan on August 28, 2011 at 12:25 am | Reply

James,

I do not have any such images. But the book "Pattern Classification" by Duda et al has some nice pics on their behavior with different data/ diff k values etc. Hope they are useful.

Danny on August 28, 2011 at 10:54 pm | Reply

Hai sir,…do you have some ebook for reference KNN-Algoritm, Now i am more need that ebook, please give me link download that ebook. thank you.

Saravanan Thirumuruganathan on August 29, 2011 at 3:30 pm | Reply

Danny,

Most classic books in Pattern Recognition do not have ebooks. You, unfortunately, have to shell out some money for that. Alternatively, you can check out pdfs/lecture notes from other universities.

andrajula on September 23, 2011 at 1:43 am | Reply

hi saravanan enaku KNN C,C++,JAva code venum
i need code could you send me to my mail gowthamandrajula@gmail.com

Saravanan Thirumuruganathan on September 23, 2011 at 1:50 am | Reply

Hi,

The post does give a link to KNN implementation in C++. There are lots of packages available in java and other languages for KNN. A google search should throw up plenty of hits 🙂

Anonymous on October 4, 2011 at 5:52 pm | Reply

Can you please finish this sentence "Where n is the total number of…"? I am not clear whether you are referring to the number of points or the dimensions of the feature set. I've been experimenting with your approximation of sqrt(n) for k, assuming n is the number of data points, but I get very poor results. I think this is because my classes are not very evenly distributed, and I have many more sample points with one particular (background) classification. Looks like I may need to look into some more complex methods for determining k.

Manas Bapna on October 10, 2011 at 5:18 am | Reply

nice article sir………

Anonymous on November 7, 2011 at 1:41 pm | Reply

Can you suggest me some Articles or Books related to Modified K-Mean(I am working on this Topic for My Thesis)..
Thanks

Saravanan Thirumuruganathan on November 19, 2011 at 2:43 am | Reply

Hi,

I would suggest you to take a look at Suresh's Geomblog (given in the More readings section). He also conducted a entire graduate level seminar on clustering. hope this is useful.

michaael on November 12, 2011 at 3:11 am | Reply

Hello can you help me…. i don't no how to code the k-nn algorithm to robot and the other codes. the concept of my proposal are identified the colors of banana and count in different and save and them send the report through SMS through bluetooth",

Saravanan Thirumuruganathan on November 19, 2011 at 2:36 am | Reply

Michaael,

I do not much of robotics and hence may not be able to help directly. If you have any conceptual question in KNN, I would be glad to be of help.

michaael on November 21, 2011 at 7:33 pm | Reply

sir Saravanan. you know how to code k-nn in microsoft visual c++ or c++, c.?..

my project is about to identified RGB colors. Example they have 2 bananas on the field identified the 2 different colors of banana. thanks for helping me

> **Saravanan Thirumuruganathan** on November 22, 2011 at 8:25 pm | Reply
>
> Michaael,
>
> There are lot of codes available in the internet to do KNN based on different media type (vectors,images,video etc). I would suggest googling it . I personally have no idea about vision or image processing. 🙂

d on November 25, 2011 at 4:17 am | Reply

sir plz send me using C/C++, Java) to implement the k-Nearest Neighbor algorithm. as soon as possible on my mail id deepakjaipurriet@gmail.com

michael on December 11, 2011 at 9:25 pm | Reply

sir, can you leave a internet link or code of knn with algorithm. pls pls.. give me a K-NN code using C language. thank you very much for giving code.

my e-mail: vassili_v28@yahoo.com.ph

Anonymous on January 12, 2012 at 2:02 am | Reply

Could you please tell me how to select k using cross validation or refer some articles/books related to this.

Tanmay on January 12, 2012 at 2:50 am | Reply

Hi, can you please tell me how to select k for knn using cross validation, and please tell me some references if you have.

> **Saravanan Thirumuruganathan** on March 18, 2012 at 10:07 pm | Reply
>
> Tanmay,
>
> I am afraid, I dont have much insight into this issue.

john on January 30, 2012 at 8:02 pm | Reply

Saravanan,

It would be great if you could provide me some link for implementing Knn algorithm using C programming language, Please help me if you can, it's urgent.

Raja Ramasamy on February 7, 2012 at 1:03 pm | Reply

hi anybody can help me in regarding KNN problem for graph using wait and without wait.

Raja Ramasamy on February 7, 2012 at 3:13 pm | Reply

hi anybody can help me in regarding KNN problem for graph using weight and without weight.

> **Saravanan Thirumuruganathan** on March 18, 2012 at 9:56 pm | Reply
>
> Raja,
>
> Nearest neighbor on graphs is a fairly well studied problem with lots of tutorials and survey papers. A google search would help. Let me know if you need any specific help.

sathys on February 21, 2012 at 6:28 pm | Reply

a very clear explanation of knn, i was able to implement knn after going through this article.can you please say about the variations in knn like weighted knn,modified knn(MKNN),fuzzy knn.

erica on April 1, 2012 at 2:03 pm | Reply

hey….i need to implement KNN in matlab….can you help me…

suhit on April 14, 2012 at 11:40 am | Reply

Awesome post!!! Thanks a lot 🙂

lordamit on April 24, 2012 at 2:28 am | Reply

Excellent writing. Thanks!

Hellström *on May 6, 2012 at 8:02 am* | Reply

Excellent post. Just one comment on the idea (not yours maybe 🙂 to use cross-validation to figure out the optimal k: Computing the performance of knn on a fixed data set with N points is already (N-fold) cross validated. So there's no more information to get from repeating this in separate cross validation…

jawi *on July 6, 2012 at 12:31 am* | Reply

hello sir,
Do you have any idea about "K reverse nearest neighbour" algorithm……I want this in c++….please help

anonymous *on September 17, 2012 at 4:07 pm* | Reply

I need java code for implementation of K-NN algorithm,can u provide me or suggest any link from where i can it.I tried but didn't find out proper one…thx in advance..:)

JLK *on September 18, 2012 at 12:28 pm* | Reply

Hi Saravanan,
I am trying to read your blob to see if it is applicable to myuse cass.

I have thousands to ten-thousands of data points (x,y)coming from 5 to 6 different source. I need to uniquely group them based on certain distance criteria in such a way that the formed group should exactly contain only one input from each source and each of them in the group should be within certain distance d. The groups formed should be the best possible match.
1.Is this a combination of clustering and nearest neighbor?
2.What are the recommendation for the algorithms?
3.Are there any open source available for it?

I see many references saying KD tree implementation and k-clustering etc. I am not sure how can I tailor to this specific need.

I posted teh above question in another website and I got reply saying mine is not a clustering since clustering always is a random association of points. My requirement might be a 5 fold Nearest Neighbor Join. I want to understand what it is. Please send any inputs that might have.

Thanks

JLK *on September 18, 2012 at 10:46 pm* | Reply

That is very goos blog, I have one question. Can you please see if you can answer it. I have thousands to ten-thousands of data points (x,y)coming from 5 to 6 different source. I need to uniquely group them based on certain distance criteria in such a way that the formed group should exactly contain only one input from each source and each of them in the group should be within certain distance d. The groups formed should be the best possible match.
1.Is this a combination of clustering and nearest neighbor?
2.What are the recommendation for the algorithms?
3.Are there any open source available for it?

I see many references saying KD tree implementation and k-clustering etc. I am not sure how can I tailor to this specific need. I got some responses saying this is not really a clustering case since clustering clusters a group of objects, need not be based on the source/count and recommendation was to use NN join. What is meant by Nearest Neighbor join?

Deep *on January 21, 2013 at 9:54 am* | Reply

What do you mean by Error Rate of the KNN classification algorithm?

Rajneesh *on January 23, 2013 at 2:47 pm* | Reply

This is Excellent Material..
Thanks a lot.

Harshitha *on March 7, 2013 at 11:50 am* | Reply

Hi,
This was a very neat explanation of K-nearest algorithm. Thank you. I am currently planning to use this algorithm in extracting temporal relation between events within a given textual annotated data but I am not very clear on how to go about it using this algorithm. I am aware of the SVM machine learning algorithm used in this area but using KNN is more simpler and easier and so, I want to give it a shot. Do you know of any source that would provide the details of its application in temporal relation analysis or text mining concepts?
Thank you.

M.Subramoniam *on March 21, 2013 at 3:06 am* | Reply

Respected Sir,
The post explains very clearly about K-NN classifier. I got the answers for many of the questions which i had for a very long days. Kindly post some article regarding the support vector machines (SVM)Classifiers too…..

Nillofer Latheef *on March 31, 2013 at 12:53 pm* | *Reply*
which is more eficient.. K-means or KNN ?

Mayuresha *on April 16, 2013 at 4:19 pm* | *Reply*
[…] Reference: https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neig… […]

sawi *on April 18, 2013 at 1:39 am* | *Reply*
Hi,
I agree with M.Subramoniam that it is a very clear explanation of KNN. It explains many thing about KNN without going into to much of math and gives a vary good idea of the method. Thanks!

aravinda *on April 27, 2013 at 9:58 am* | *Reply*
hello can i get source code of knn classifier for handwritten ocr plz?

Italo Farfán *on May 2, 2013 at 9:28 pm* | *Reply*
Hello,
Excellent post. Thank you so much, your knowledge is very useful for me :). Here there are examples about Knn using Scikit-Learn (python)
http://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html)

ajay kumar *on June 24, 2013 at 4:43 am* | *Reply*
Hi,
In Knn what problem could occur, if k is even?

Anonymous *on August 12, 2013 at 2:58 pm* | *Reply*
Hi,
This is a great post for understanding basics of K-NN. Kudos to Saravanan Thirumuruganathan.

My question is about the number of classes. Say we have data from only one class for a two-class problem (normal-class and abnormal-class). Given a test-point and normal-class data points, is it possible to use 1-NN to say whether the test-point is a member of normal-class? I think this might be called an outlier detection problem as well.

Anonymous *on August 17, 2013 at 1:26 pm* | *Reply*
Can i get this Knn program:mdbasha4u@yahoo.in

lovekesh *on August 22, 2013 at 4:38 am* | *Reply*
hi
I'll appreciate if you can point me to code implementation of KNN algorithm in R. I am pursuing my masters in economics and i don't want to use in-built class in R. I'll greatly understand it if i can take a look at the implementation code.
Thanks

priya *on November 19, 2013 at 8:20 am* | *Reply*
sir,

i am priya. doing my M.E. i want the code of knn algorithm in java for my project. i searched in google. bt i'm nt getting the exact one. sir pls do post me a link or reply the link to priyasaishu@gmail.com

deepak *on December 12, 2013 at 4:40 am* | *Reply*
Sir plz tell me how to use KNN algo to predict weather attributes using historical data.

Anonymous *on January 15, 2014 at 4:21 pm* | *Reply*
Hi Saravanan,

Concrete question: let's say that I have N, n-dimensional data points. I want to find an estimate for my (N+1)st data points having attributes (a1, a2, …, an). Now, I identify my k-nearest neighbors to my (N+1)st and produce an estimate for this data point. How do I produce a confidence interval for this estimate? Is it as simple as computing the sample standard deviation of the k points used and building a confidence interval about the estimate obtained for the (N+1)st point? Or is there an additional penalty that's involved?

Mr N *on March 4, 2014 at 6:20 am* | *Reply*
Thanks A lot for the explanation Mr Saravanan Thirumuruganathan

> **priya** *on March 12, 2014 at 10:32 pm* | *Reply*
> sir thank you so much for your valuable explanation about KNN. and i need to implement KNN code in java. can you please help me with this???

Comments RSS