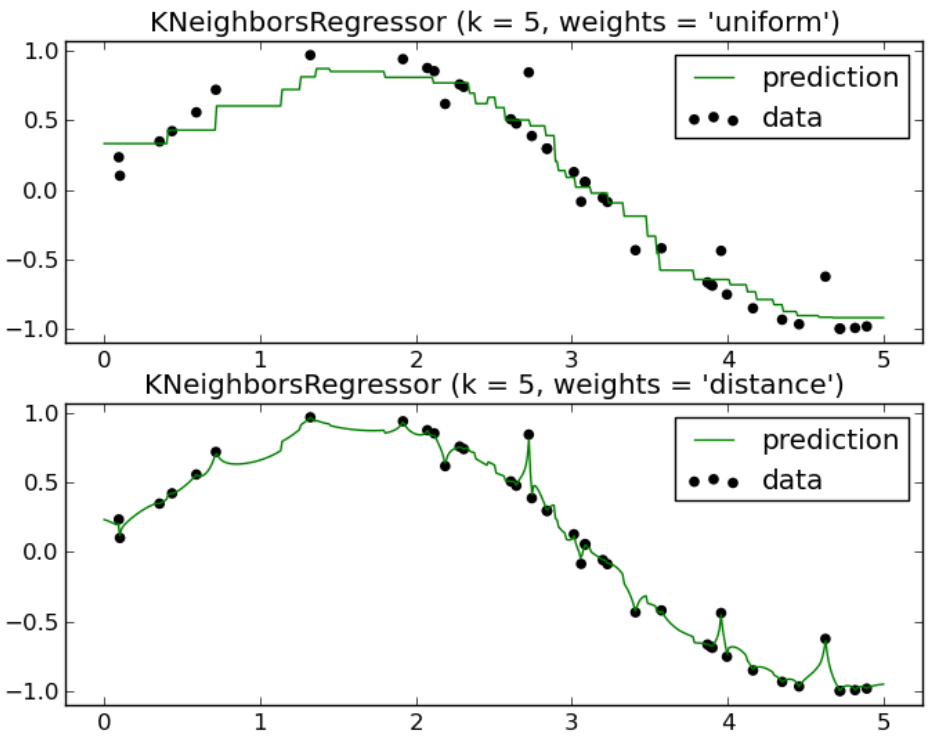
Why would anyone use KNN for regression?

From what I understand, we can only build a regression function that lies within the interval of the training data.

For example (only one of the panels is necessary):



How would I predict into the future using a KNN regressor? Again, it appears to only approximate a function that lies within the interval of the training data.

My question: What are the advantages of using a KNN regressor? I understand that it is a very powerful tool for classification, but it seems that it would perform poorly in a regression scenario.

regression machine-learning k-nearest-neighbour

edited Jun 21 '14 at 20:50
gung ♦
105k 34 255 518

asked Jun 21 '14 at 20:38
user46925

Can you clarify what you mean by "predict into the future"? Do you have time-series & you want to do forecasting, or are you trying to fit a relationship between 2 variables & want to use that in the future to guess a Y value from a known X value? – gung ♦ Jun 21 '14 at 20:51

1 For example, if I wanted to predict the value Y such that X=15 from the image above. A KNN-regressor wouldn't cut it right? – user46925 Jun 22 '14 at 1:38

1 I would agree with you that if you trained on a set with $x \in [0, 5]$ but expected that you may see values of x far beyond what is in your data then non-parametric local methods might not be ideal. Instead you might want to use that domain knowledge and define a parametric model that includes your knowledge of how 'unobserved' x is expected to behave. – Meadowlark Bradsher Jun 22 '14 at 2:45

1 An example of KNN being used successfully for regression is Nate Silver's PECOTA baseball prediction thing. You can read about the pros and cons from the Wikipedia article on PECOTA or newspaper articles like this one: macleans.ca/authors/colby-cosh/… – Flounderer Jun 22 '14 at 2:48

5 Also to make a more general point, as you become knowledgeable in statistics (or data mining/machine learning etc) you'll find that answers to very general questions such as yours will often be a paraphrased version of 'it depends'. Knowing what 'it depends' on and why is the knowledge. – Meadowlark Bradsher Jun 22 '14 at 2:51

Local methods like K-NN make sense in some situations.

One example that I did in school work had to do with predicting the compressive strength of various mixtures of cement ingredients. All of these ingredients were relatively non-volatile with respect to the response or each other and KNN made reliable predictions on it. In other words none of the independent variables had disproportionately large variance to confer to the model either individually or possibly by mutual interaction.

Take this with a grain of salt because I don't know of a data investigation technique that conclusively shows this but intuitively it seems reasonable that if your features have some proportionate degree of variances, I don't know what proportion, you might have a KNN candidate. I'd certainly like to know if there were some studies and resulting techniques developed to this effect.

If you think about it from a generalized domain perspective there is a broad class of applications where similar 'recipes' yield similar outcomes. This certainly seemed to describe the situation of predicting outcomes of mixing cement. I would say if you had data that behaved according to this description and in addition your distance measure was also natural to the domain at hand and lastly that you had sufficient data, I would imagine that you should get useful results from KNN or another local method.

You are also getting the benefit of extremely low bias when you use local methods. Sometimes generalized additive models (GAM) balance bias and variance by fitting each individual variable using KNN such that:

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) + \epsilon$$

The additive portion (the plus symbols) protect against high variance while the use of KNN in place of $f_n(x_n)$ protects against high bias.

I wouldn't write off KNN so quickly. It has its place.

edited Sep 26 '14 at 15:59

answered Jun 21 '14 at 21:50

Meadowlark Bradsher
683 8 22

First an example for "How would I predict into the future using a KNN regressor ?".

Problem: predict hours of sunlight tomorrow sun_{t+1} from $sun_t \,..\, sun_{t-6}$ over the last week.
Training data: sun_t (in one city) over the last 10 years, 3650 numbers.

Denote $week_t \equiv sun_t \,..\, sun_{t-6}$ and $tomorrow(week_t)) \equiv sun_{t+1}$.

Method: put the 3650-odd $week_t$ curves in a k-d tree with k=7.
Given a new $week$, look up its say 10 nearest-neighbor weeks
with their $tomorrow_0 \,..\, tomorrow_9$ and calculate
$predict(week) \equiv$ weighted average of $tomorrow_0 \,..\, tomorrow_9$

Tune the weights, see e.g. inverse-distance-weighted-idw-interpolation-with-python,
and the distance metric for "Nearest neighbor" in 7d.

"What are the advantages of using a KNN regressor ?"
To others' good comments I'd add easy to code and understand, and scales up to big data.
Disadvantages: sensitive to data and tuning, not much *understanding*.

(Longish footnote on terminology:
"regression" is used as a fancy word for "fitting a model to data".
Most common is fitting data X to a target Y with a linear model:
$Y_t = b_0 X_t + b_1 X_{t-1} + \ldots$
Also common is predicting tomorrow's say stock price Y_{t+1} from prices over the last week or year:
$Y_{t+1} = a_0 Y_t + a_1 Y_{t-1} + \ldots$
Forecasters call this an ARMA, Autoregressive moving-average_model or Autoregressive model . See also Regression analysis .

So your first line "we can only build a regression function that lies within the interval of the training data" seems to be about the confusing word "regression".)

I don't like to say it but actually the short answer is, that "predicting into the future" is not really possible not with a knn nor with any other currently existing classifier or regressor.

Sure you can extrapolate the line of a linear regression or the hyper plane of an SVM but in the end you don't know what the future will be, for all we know, the line might just be a small part of a curvy reality. This becomes apparent when you look at Bayesian methods like Gaussian processes for instance, you will notice a big uncertainty as soon as you leave the "known input domain".

Of course you can try to generalize from what happened today to what likely happens tomorrow, which can easily be done with a knn regressor (e.g. last year's customer numbers during Christmas time can give you a good hint about this year's numbers). Sure other methods may incorporate trends and so on but in the end you can see how well that works when it comes to the stock market or long-term weather predictions.

answered Nov 27 at 20:52

meow

123 5