Open in app

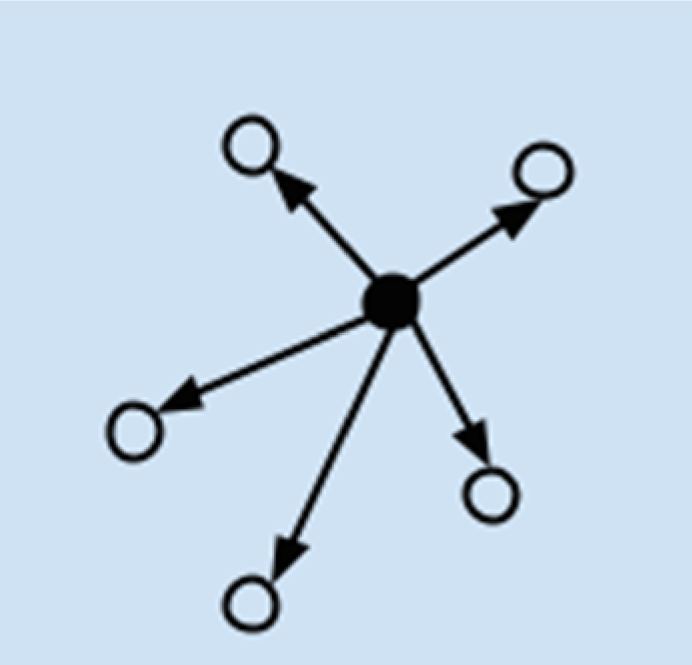
towards
data science

Follow 590K Followers

ML Algorithms: One SD (σ)- Instance-based Algorithms

An intro to machine learning instance-based algorithms

Sagi Shaier Feb 1, 2019 · 5 min read



The obvious questions to ask when facing a wide variety of machine learning algorithms, is "which algorithm is better for a specific task, and which one should I use?"

> *Answering these questions vary depending on several factors, including: (1) The size, quality, and nature of data; (2) The available computational time; (3) The urgency of the task; and (4) What do you want to do with the data.*

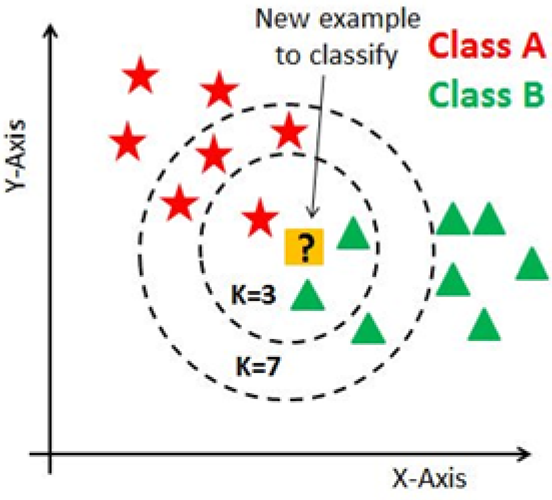
This is one section of the many algorithms I wrote about in a previous article.
In this part I tried to display and briefly explain the main algorithms (though not all of them) that are available for instance-based tasks as simply as possible.

Instance-based Algorithms:

> These algorithms don't perform explicit generalization, instead they compare new problem instances with instances seen in training, which have been stored in memory.

· **K-Nearest Neighbor (KNN)**

Can be used for both classification and regression problems. KNN stores all available cases and classifies new cases by a majority vote of its K neighbors. Predictions are made for a new data point by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances. For instance, if we take K=3 and we want to decide which class does a new example belongs to, we consider the 3 closest (Euclidian distance usually) points to the new example.



For regression problems, this might be the mean output variable:

Some things to consider:

Choosing the optimal value for K is best done by first inspecting the data (you can use the elbow method).

It is a supervised learning algorithm.

· **Learning Vector Quantization (LVQ)**

Developed as a classification algorithm. It is capable of supporting both binary (two-class) and multi-class classification problems. A downside of K-Nearest Neighbors is that you need to hang on to your entire training dataset. The LVQ is an artificial neural network algorithm that allows you to choose how many training instances to hang onto and learns exactly what those instances should look like. The value of the number of instances is optimized during learning process.

Some things to consider:

It is a supervised learning method

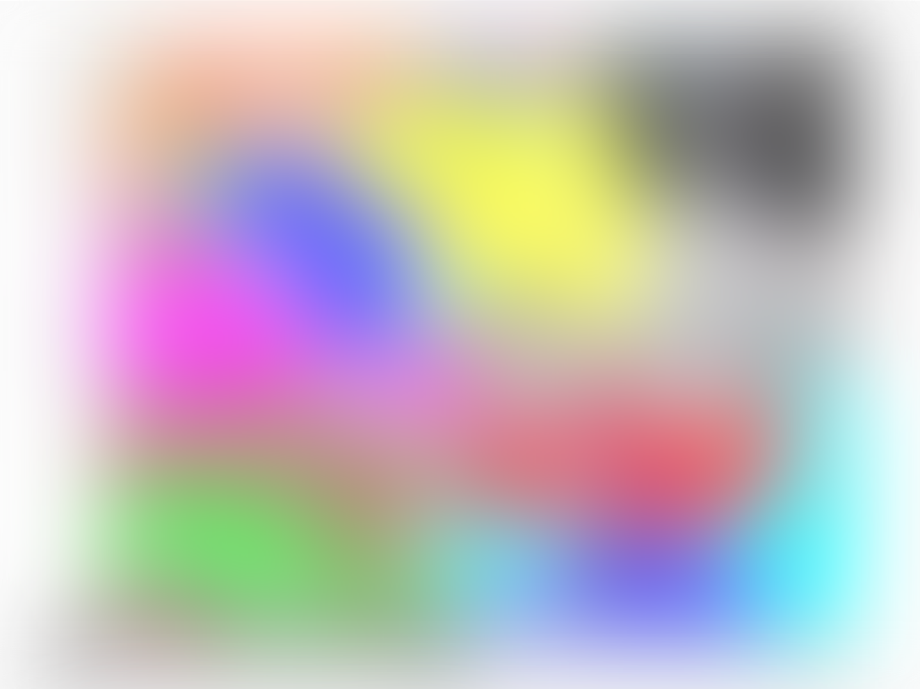
If you discover that KNN gives good results on your dataset try using LVQ to reduce the memory requirements of storing the entire training dataset.

· **Self-Organizing Map (SOM)**

An unsupervised deep learning model, mostly used for feature detection or dimensionality reduction. SOM differ from other artificial neural networks as it apply competitive learning as opposed to error-correction learning (like backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space. SOM performs a topologically ordered mapping from high dimensional space onto two-dimensional space. In other words, it produces a two dimensional representation of the input space of the set of training samples.

For example, let's look at the handwritten digits dataset. The inputs for SOM are high dimensional since each input dimension represents the grayscale value of one pixel on a 28 by 28 image, which makes the inputs 784-dimensional (each dimension is a value between 0 and 255).

If we'll map them to a 20x20 SOM and color them based on their true class (a number from 0 to 9) we'll get the following:



The true classes are labelled according to the colors in the bottom left.

Take a look at the yellow region. That is where the 6s were mapped to, and notice that there is a little overlap with other categories. In comparison, take a look at the bottom left, where the green and brown points overlap. That is where the SOM was “confused” between 4s and 9s.

Another example of SOM is NLP. We can use it for a classification of let’s say 2 million medical papers. SOM will create a cluster of similar meaning words:

The bottom right words are related to brain, and the top right words are related to medical imaging.

Some things to consider:

SOM outputs a 2D map for any number of indicators.

We could use the SOM for clustering data without knowing the class memberships of the input data.

· **Locally Weighted Learning (LWL)**

The basic idea behind LWL is that instead of building a global model for the whole function space, for each point of interest a local model is created based on neighboring data of the query point.

For this purpose, each data point becomes a weighting factor which expresses the influence of the data point for the prediction. In general, data points which are in the close neighborhood to the current query point are receiving a higher weight than data points which are far away. Basically, say you want to predict what is going to happen in the future. You can simply reach into a database of all your previous experiences, you then grab some similar experiences, combine them (perhaps by a weighted average that weights more similar experiences more strongly) and use the combination to make a prediction.

Some things to consider:

LWL methods are non-parametric.

Until next time,

Bobcat.

Machine Learning

Data Science

Algorithms

Pancakes