Chapter 5 Similarity-based Learning

Similarity-based approaches to machine learning come from the idea that the best way to make a predictions is to simply look at what has worked well in the past and predict the same thing. The fundamental concepts required to build a system based on this idea are feature spaces and measures of similarity, and these are covered in the fundamentals section of this chapter. These concepts allow us to understand the standard approach to building similarity-based models: the nearest neighbor algorithm. After covering the standard algorithm, we then look at extensions and variations that allow us to handle noisy data (the k nearest neighbor, or k-NN, algorithm), to make predictions more efficiently (k-d trees), to predict continuous targets, and to handle different kinds of descriptive features with varying measures of similarity. We also take the opportunity to introduce the use of data normalization and feature selection in the context of similarity-based learning. These techniques are generally applicable to all machine learning algorithms but are especially important when similarity-based approaches are used.

5.1 Big Idea

The year is 1798, and you are Lieutenant-Colonel David Collins of HMS Calcutta exploring the region around Hawkesbury River, in New South Wales. One day, after an expedition up the river has returned to the ship, one of the men from the expedition tells you that he saw a strange animal near the river. You ask him to describe the animal to you, and he explains that he didn't see it very well because, as he approached it, the animal growled at him, so he didn't approach too closely. However, he did notice that the animal had webbed feet and a duck-billed snout.

In order to plan the expedition for the next day, you decide that you need to classify the animal so that you can determine whether it is dangerous to approach it or not. You decide to do this by thinking about the animals you can remember coming across before and comparing the features of these animals with the features the sailor described to you. We illustrate this process by listing some of the animals you have encountered before and how they compare with the growling, web-footed, duck-billed animal that the sailor described. For each known animal, you count how many features it has in common with the unknown animal. At the end of this process, you decide that the unknown animal is most similar to a duck, so that is what it must be. A duck, no matter how strange, is not a dangerous animal, so you tell the men to get ready for another expedition up the river the next day.

The process of classifying an unknown animal by matching the features of the animal against the features of animals you have encountered before neatly encapsulated the big idea underpinning similarity-based learning: if you are trying to make a prediction for a current situation then you should search your memory to find situations that are similar to the current one and make a prediction based on what was true for the most similar situation in your memory.

5.2 Fundamentals

As the name similarity-based learning suggests, a key component of this approach to prediction is defining a computational measure of similarity between instances. Often this measure of similarity is actually some form of distance measure. A consequence of this, and a somehow less obvious requirement of similarity-based learning, is that if we are going to compute distances between instances, we need to have a concept of space in the representation of the domain used by our model. In this section we introduce the concept of a feature space as a representation for a training dataset and then illustrate how we can compute measures of similarity between instances in a feature space.

5.2.1 Feature Space

We list an example dataset containing two descriptive features, the SPEED and AGILITY ratings for college athletes (both measures out of 10), and one target feature that list whether the athletes were drafted to a professional team. We can represent this dataset in a feature space by taking each of the descriptive features to the axes of a coordinate system. We can then place each instance within the feature space based on the values of its descriptive features. There is a scatter plot to illustrate the resulting feature space when we do this using the data. In this figure SPEED has been plotted on the horizontal axis, and AGILITY has been plotted on the vertical axis. The value of the DRAFT feature is indicated by the shape representing each instance as a point in the feature space: triangles for no and crosses for yes.

There is always one dimension for every descriptive feature in a dataset. In this example, there are only two descriptive features, so the feature space is two-dimensional. Feature spaces can, however,have many more dimensions-in document classification tasks, for example, it is not uncommon to have thousands of descriptive features and therefore thousands of dimensions in the associated feature space. Although we can't easily draw feature spaces beyond three dimensions, the ideas underpinning them remain the same.

We can formally define a feature space as an abstract m-dimensional space that is created by making each descriptive feature in a dataset an axis of an m-dimensional coordinate system and mapping each instance in the dataset to a point in the coordinate system based on the values of its descriptive features.

For similarity-based learning, the nice thing about the way feature spaces work is that if the values of the descriptive features of two or more instances in the dataset are the same, then these instances will be mapped to some point in the feature space. Also, as the difference between the values of the descriptive features of two instances grows, so too does the distance between the points in the feature space that represent these instances. So the distance between two points in the feature space is a useful measure of the similarity of the descriptive features of the two instances.

5.2.2 Measuring Similarity Using Distance Metrics

The simplest way to measure the similarity between two instances, and , in a dataset is to measure the distance between the instances in a feature space. We can use a distance metric to do this:
$metric(a, b)$
is a function that returns the distance between two instances and . Mathematically, a
$metric$
must conform to the following four criteria:

1. **Non-negativity:**
$metric(a, b) \geqslant 0$

2. **Identity**:
$metric(a, b) = 0 \Leftrightarrow a = b$

3. **Symmetry**:
$metric(a, b) = metric(b, a)$

4. **Triangular Inequality**:

$metric(a, b) \leqslant metric(a, c) + metric(b, c)$

One of the best known distance metrics is **Euclidean distance**, which computes the length of the straight line between two points. Euclidean distance between two instances and in an m-dimensional feature space is defined as

b

$$Euclidean(a, b) = \sqrt{\sum_{i=1}^{m}(a[i] - b[i])^2} \qquad (1)$$

d_{12} The descriptive features in the college athlete dataset are both continuous, which means that the feature space representing this data is technically known as a **Euclidean coordinate space**, and we can compute the distance between instances in it using Euclidean distance. Foe example, the Euclidean distance between instances (SPEED = 5.00, AGILITY = 2.50) and d_5 (SPEED = 2.75, AGILITY = 7.50) from Table (1) is

$$Euclidean(d_{12}, d_5) = \sqrt{(5.00 - 2.75)^2 + (2.50 - 7.50)^2} = 30.0625 = 5.4829$$

Another, less well-known, distance metric is the **Manhattan distance**. The Manhattan distance between two instance a and b in a feature space with m dimensions is defined as

$$Manhattan(a, b) = \sum_{i=1}^{m} abs(a[i] - b[i]) \qquad (2)$$

where the $abs()$ function returns the absolute value. For example, the Manhattan distance between instances d_{12} (SPEED = 5.00, AGILITY = 2.50) and d_5 (SPEED = 2.75, AGILITY = 7.50) in Table (2) is

$$Manhattan(d_{12}, d_5) = abs(5.00 - 2.75) + abs(2.5 - 7.5) = 2.25 + 5 = 7.25$$

We illustrate the differences between the Manhattan and Euclidean distances between two points in a two-dimensional feature space. If we compare Equation(1) and Equation(2), we can see that both distance metrics are essentially functions of the differences between the values of the features. Indeed, the Euclidean and Manhattan distances are special cases of the **Minkowski distance**, which defines a family of distance metrics based on different between features.

The **Minkowski distance** between two distances a and b in a feature space with m dimensions is defined as

$$Minkowski(a, b) = (\sum_{i=1}^{m} abs(a[i] - b[i])^p)^{\frac{1}{p}} \qquad (3)$$

where the parameter p is typically set to a positive value and defines the behavior of the distance metric. Different distance metrics result from adjusting the value of p. For example, the Minkowski distance with $p = 1$ is the Manhattan distance, and with $p = 2$ is the Euclidean distance. Continuing in this manner, we can define an infinite number of distance metrics.

The fact that we can define an infinite number of distance metrics is not merely an academic curiosity. In fact, the predictions produced by a similarity-based model will change depending on the exact Minkowski distance used (i.e., $p = 1, 2, .., \infty$). Larger values of p place more emphasis on large differences between feature values than smaller values of p because all differences are raised to the power of p. Consequently, the Euclidean distance (with $p = 2$) is more strong influenced by a single large difference in one feature than the Manhattan distance (with $p = 1$) .

We can see this if we compare the Euclidean and Manhattan distances between instances d_{12} and d_5 with the Euclidean and Manhattan distances between instances d_{12} and d_{17} (SPEED = 5.25, AGILITY = 9.50).

The Manhattan distances between both pairs of instances are the same: 7.25. It is striking, however, that the Euclidean distance between d_{12} and d_{17} is 8.25, which is greater than the Euclidean distance between d_{12} and d_5, which is just 5.48. This because the maximum difference between d_{12} and d_{17} for any single feature is 7 units (for AGILITY), whereas the maximum difference between d_{12} and d_5 on any single feature is just 5 units (for AGILITY). Because these differences are squared in the Euclidean distance calculation, the larger maximum single difference between d_{12} and d_{17} results in a larger overall distance being calculated for this pair of instances. Overall the Euclidean distance weights features with larger differences in values more than features with smaller differences in values. This means that the Euclidean difference is more influenced by a single large difference in one feature rather than a log of small differences across a set of features, whereas the opposite is true of Manhattan distance.

Although we have an infinite number of Minkowski-based distance metrics to choose from, Euclidean distance and Manhattan distance are the most commonly used of these. The question of which is the best one to use, however, still remains. From a computational perspective, the Manhattan distance has a slight advantage over the Euclidean distance - the computation of the squaring and the square root is saved - and computational considerations can become important when dealing with very large datasets. Computational considerations aside, Euclidean distance is often used as the default.