Machine Learning

Aprendizaje basado en instancias

K vecinos más próximos

Angel Vázquez-Patiño
angel.vazquezp@ucuenca.edu.ec

Departamento de Ciencias de la Computación
Universidad de Cuenca

9 de noviembre de 2017

Objetivo

1) Entender qué es el aprendizaje basado en instancias
2) Entender el algoritmo knn y su versión ponderada por distancia
3) Dejar claro los problemas de aplicación para knn
4) Poner en práctica knn para una base de datos real

Contenido

Aprendizaje basado en instancias

Aprendizaje basado en instancias

- In contrast to learning methods that construct a general, explicit description of the target function when training examples are provided, instance-based learning methods simply **store the training examples**
- **Generalizing** beyond these examples is **postponed** until a new instance must be classified
- Each time a new query instance is encountered, its relationship to the previously stored examples is examined in order to assign a target function value for the new instance

Aprendizaje basado en instancias

- Instance-based learning includes nearest neighbor and locally weighted regression methods that assume instances can be represented as points in a Euclidean space. It also includes case-based reasoning methods that use more complex, symbolic representations for instances
- Instance-based methods are sometimes referred to as "lazy" learning methods because they delay processing until a new instance must be classified
- A key **advantage** of this kind of delayed, or lazy, learning is that instead of estimating the target function once for the entire instance space, these methods can estimate it locally and differently for each new instance to be classified

Aprendizaje basado en instancias

- They are conceptually straightforward approaches to approximating **real**-valued or **discrete**-valued target functions
- Learning in these algorithms consists of simply storing the presented training data. When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query instance

Aprendizaje basado en instancias

- One key **difference** between these approaches and the methods discussed in other chapters is that instance-based approaches can construct a **different approximation to the target function for each distinct query** instance that must be classified
- In fact, many techniques construct only a local approximation to the target function that applies in the neighborhood of the new query instance, and never construct an approximation designed to perform well over the entire instance space
- This has significant **advantages** when the **target function is very complex**, but can still be described by a collection of less complex local approximations

Aprendizaje basado en instancias

- Instance-based methods can also use more complex, symbolic representations for instances. In case-based learning, instances are represented in this fashion and the process for identifying "neighboring" instances is elaborated accordingly
- Case-based reasoning has been applied to tasks such as storing and reusing past experience at a help desk, reasoning about legal cases by referring to previous cases, and solving complex scheduling problems by reusing relevant portions of previously solved problems

Aprendizaje basado en instancias

- One **disadvantage** of instance-based approaches is that the **cost** of classifying new instances can be high. This is due to the fact that nearly **all computation takes place at classification time** rather than when the training examples are first encountered. Therefore, techniques for efficiently indexing training examples are a significant practical issue in reducing the computation required at query time
- A second **disadvantage** to many instance-based approaches, especially nearest-neighbor approaches, is that they typically **consider all attributes of the instances** when attempting to retrieve similar training examples from memory. If the target concept depends on only a few of the many available attributes, then the instances that are truly most "similar" may well be a large distance apart.

k vecinos más próximos

K vecinos más próximos

- The most basic instance-based method
- This algorithm assumes all instances correspond to points in the n-dimensional space
- The nearest neighbors of an instance are defined in terms of the standard Euclidean distance

$$\langle a_1(x), a_2(x), \ldots a_n(x) \rangle$$

where $a_r(x)$ denotes the value of the rth attribute of instance x. Then the distance between two instances x_i and x_j is defined to be $d(x_i, x_j)$, where

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

K vecinos más próximos

- In nearest-neighbor learning the target function may be either **discrete**-valued or **real**-valued

Training algorithm:

- For each training example $\langle x, f(x) \rangle$, add the example to the list *training_examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \ldots x_k$ denote the k instances from *training_examples* that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

K vecinos más próximos

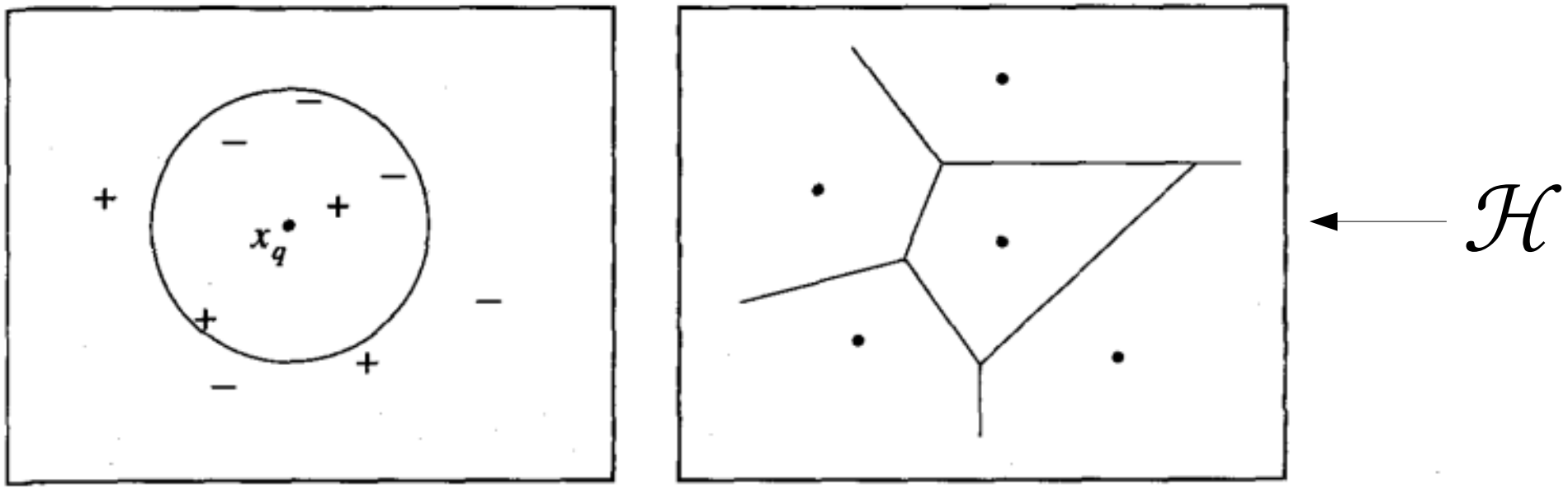


FIGURE 8.1
k-NEAREST NEIGHBOR. A set of positive and negative training examples is shown on the left, along with a query instance x_q to be classified. The 1-NEAREST NEIGHBOR algorithm classifies x_q positive, whereas 5-NEAREST NEIGHBOR classifies it as negative. On the right is the decision surface induced by the 1-NEAREST NEIGHBOR algorithm for a typical set of training examples. The convex polygon surrounding each training example indicates the region of instance space closest to that point (i.e., the instances for which the 1-NEAREST NEIGHBOR algorithm will assign the classification belonging to that training example).

Algoritmo knn ponderado por distancia

knn ponderado por distancia

- One obvious refinement to the k-NEAREST NEIGHBOR algorithm is to weight the contribution of each of the k neighbors according to their distance to the query point xq, giving greater weight to closer neighbors

knn ponderado por distancia

- Inverso del cuadrado de la distancia

Given a query instance x_q to be classified,

- Let $x_1 \ldots x_k$ denote the k instances from *training_examples* that are nearest to x_q
- Return

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

$$w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

knn ponderado por distancia

- Note all of the above variants of the k-NEAREST NEIGHBOR algorithm **consider only the k nearest neighbors** to classify the query point. Once we add distance weighting, there is really no harm in allowing all training examples to have an influence on the classification of the xq, because very distant examples will have very little effect on f(xq). The only **disadvantage** of considering all examples is that our classifier will **run** more **slowly**. If all training examples are considered when classifying a new query instance, we call the algorithm a global method
- If only the nearest training examples are considered, we call it a local method

Observaciones

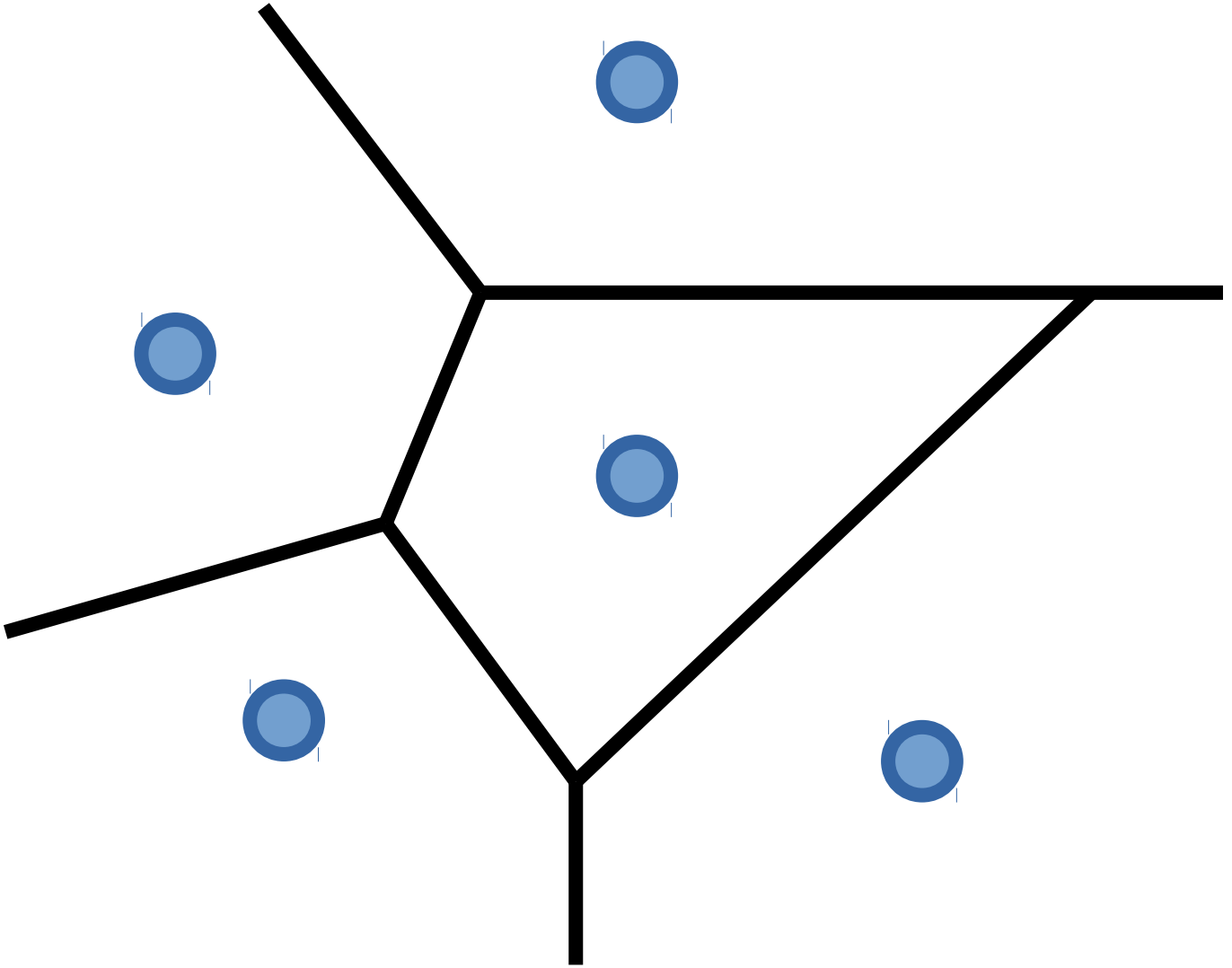
Observaciones

- El algoritmo knn is a highly effective inductive inference method for many practical problems. It is **robust to noisy training data** and quite effective when it is provided a **sufficiently large set of training data**
- Note that by taking the weighted average of the k neighbors nearest to the query point, it can smooth out the impact of isolated noisy training examples

Observaciones

- What is the inductive bias of k-NEAREST NEIGHBOR?
- The basis for classifying new query points is easily understood based on the diagrams in Figure 8.1
- The inductive bias corresponds to an assumption that the classification of an instance x, will be most similar to the classification of other instances that are nearby in Euclidean distance

Observaciones



Observaciones

- One practical issue in applying k-NEAREST NEIGHBOR algorithms is that the **distance** between instances is **calculated based on all attributes** of the instance (i.e., on all axes in the Euclidean space containing the instances).
- This lies in contrast to methods such as rule and **decision tree learning** systems that select only a subset of the instance attributes when forming the hypothesis. To see the effect of this policy, consider applying k-NEAREST NEIGHBOR to a problem in which each instance is described by 20 attributes, but where only 2 of these attributes are relevant to determining the classification for the particular target function. In this case, instances that have identical values for the 2 relevant attributes may nevertheless be distant from one another in the 20-dimensional instance space.
- As a result, the similarity metric used by k-NEAREST NEIGHBOR--depending on all 20 attributes-will be misleading. **The distance between neighbors will be dominated by the large number of irrelevant attributes**. This difficulty, which arises when many irrelevant attributes are present, is sometimes referred to as the curse of dimensionality. **knn approaches are especially sensitive to this problem**

Observaciones

- One interesting approach to **overcoming this problem** is to **weight each attribute differently** when calculating the distance between two instances. This corresponds to stretching the axes in the Euclidean space, shortening the axes that correspond to less relevant attributes, and lengthening the axes that correspond to more relevant attributes. The amount by which each axis should be stretched can be determined automatically using a **cross-validation** approach

Observaciones

- To see how, first note that we wish to stretch (multiply) the jth axis by some factor zj , where the values z1 . . . zn are chosen to minimize the true classification error of the learning algorithm
- Second, note that this true error can be estimated using cross-validation. Hence, one algorithm is to select a random subset of the available data to use as training examples, then determine the values of z1 . . . zn, that lead to the minimum error in classifying the remaining examples. By repeating this process multiple times the estimate for these weighting factors can be made more accurate. This process of stretching the axes in order to optimize the performance of k-NEAREST NEIGHBOR provides a mechanism for suppressing the impact of irrelevant attributes

Observaciones

- An even **more drastic alternative is to completely eliminate the least relevant attributes from the instance space**. This is equivalent to setting some of the zi scaling factors to zero. Moore and Lee (1994) discuss efficient cross-validation methods for selecting relevant subsets of the attributes for k-NEAREST NEIGHBOR algorithms. In particular, they explore methods based on **leave-one-out cross-validation**
- This leave-one-out approach is easily implemented in k-NEAREST NEIGHBOR algorithms because no additional training effort is required each time the training set is redefined.

8.2.3 A Note on Terminology

Much of the literature on nearest-neighbor methods and weighted local regression uses a terminology that has arisen from the field of statistical pattern recognition. In reading that literature, it is useful to know the following terms:

- *Regression* means approximating a real-valued target function.
- *Residual* is the error $\hat{f}(x) - f(x)$ in approximating the target function.
- *Kernel function* is the function of distance that is used to determine the weight of each training example. In other words, the kernel function is the function K such that $w_i = K(d(x_i, x_q))$.

Conceptos y términos importantes

Conceptos y términos importantes

- Observaciones (sumamente útiles en la práctica)

Tarea

1) Leer la subsección K-nearest neighbors – a lazy learning algorithm del libro de Raschka (2016) y reproducir todos los ejemplos que se presenten.

2) Utilizando el FMA data set (https://goo.gl/PhtzjZ) indicar el rendimiento de K-nearest neighbours para el reconocimiento de género, identificación de artista y año de lanzamiento/grabación

- Grupos de tres. 1 de diciembre, 23h55. Evirtual. No más de 10 páginas.

Tarea

Recuerde que lo único con lo que cuento para poder saber qué tan buen trabajo hizo es el documento escrito

Referencias

- Mitchell, T.M., 1997. Machine Learning, McGraw-Hill series in computer science. McGraw-Hill, New York.
- Raschka, S., 2016. Python machine learning, Community experience distilled. Packt Publishing, Birmingham Mumbai.

Preguntas