Appunti di SIGIR 2013 Tutorial

A.A. 2023/2024

Rosso Carlo

Contents

1 Introduction

The chapter explain:

1. basic notion of machine learning
2. Naive Bayes and Decision Tree Classifier
3. PAC learning theory and the Perception Algorithm
4. Support Vector Machine and Kernel Methods

Which means, I am gonna fast forward through the basic concepts (point 1 through 3) and I am gonna focus on the SVM and Kernel Methods.

Decision tree the feature that correctly separates the highest number of training examples should be used before the others. In order to find the most discriminative feature, DTs use the entropy quantity. Let a set of classes $\{C_1, \ldots, C_n\}$ and a set of training examples S with probabilites $P(C_i)$, the entropy H of P is defined as:

$$H(P) = \sum_{i=1}^{m} -P(C_i) \log_2 P(C_i) \tag{1.1}$$

1.1 Naive Bayes

Let us indicate with E the classification example and let $\{C_1, \ldots, C_m\}$ be the set of categories in which we want to classify such example. We are interested to evaluate the probability that E belongs to C_i, i.e. $P(C_i|E)$. E can be represented as a set of features $\{f_1, \ldots, f_n\}$. Thus, we can use the Bayes' rule to derive a more useful probability form:

$$P(C_i|f_1, \ldots, f_n) = \frac{P(f_1, \ldots, f_n|C_i) \times P(C_i)}{P(f_1, \ldots, f_n)} \tag{1.2}$$

where

$$\sum_{i=1}^{m} P(C_i|f_1, \ldots, f_n) = 1$$

To make the Bayesian approach more practical, we naively assume that features are independent (very much not true in reality). Given such assumption, we can rewrite the previous equation as:

$$P(C_i|f_1, \ldots, f_n) = \prod_{k=1}^{n} \frac{P(f_k|C_i) \times P(C_i)}{P(f_1, \ldots, f_n)} \tag{1.3}$$

Note that the denominator is the same for all the classes. And, $P(C_i)$ and $P(f_k|C_i)$ can be estimated from the training set, in the following way:

$$P(C_i) = \frac{|C_i|}{\sum_{j=1}^{m} |C_j|}$$

and

$$P(f_k|C_i) = \frac{|C_i \cap f_k|}{|C_i|}$$

1.2 PAC learnable functions

Hypothesis:

- Let $f : X \to \mathcal{C}$ belongs to the class F;
- the training and the test documents $x \in X$ are generated with probability D;
- Let $h \in H$ be the hypothesis that the learning algorithm learns; where H is the set of all possible hypotheses;
- $error(h)$ be the error of the hypothesis h, defined as $P[f(x) \neq h(x)]$, i.e. the percentage of miss-classified examples;
- Let m be the size of the training set;

Then F is a class of PAC learnable functions if there is a learning algorithm such that: $\forall f \in F, \forall D \in X$ and $\forall \epsilon > 0, \delta < 1$, $\exists m$ such that $P[error(h) > \epsilon] < \delta$.
In other words, a class of functions F is PAC learnable if we can find a learnign algorithm which, given enough number of training examples, produces a function h such that its error is greater than ϵ with a probability less than δ.

1.3 The VC-dimension

We need a property that allows us to determine which hypothesis class is more appropriate to learn a target function $f \in F$. In most cases, we do not know the nature of the target function f. So, our property should be derived only from the training examples and by the function class H that we have available. Intuitively, VC dimension determines the genralization reachable during learning. The definition of VC dimension depends on the concept of shattering a set of points.

Def. 1.1 (Shattered Sets) *Let us consider binary classification functions $f \in F, f : X \to \{0, 1\}$. We say that $S \subseteq X$ is shattered by a function class F if $\forall S' \subseteq S, \exists f \in F$:*

$$f(x) = \begin{cases} 0 & \iff x \in S' \\ 1 & \iff x \in S - S' \end{cases} \tag{1.4}$$

Def. 1.2 (VC dimension) *The VC dimension of a function class F is the maximum number of points that can be shattered by F.*