Relevance Vector Machine

Artem Los, Clement Morin, Kevin Dallatorre, Remi Lacombe

I. Introduction

For supervised learning, support vector machine (SVM) is a state-of-the art way of performing classification and regression. The idea behind SVM is to *minimize the training set error* of a target function (define) while simultaneously maximizing the margin between the two classes [1].

However, SVM's have four problems that can be solved using the Relevance Vector Machines (RVM). First, SVM's do not give us a probabilistic answer, which is useful when we want to apply the notion of uncertainty in the classifier (i.e. similar to error bars). An SVM classification will produce a hard binary decision whereas the SVM regression will produce a point estimate. Secondly, the required number of kernel functions will grow steeply with the size of the training set. Thirdly, a constant C (aka error/margin trade off) has to be estimated. Finally, the kernel functions require Mercer's condition to be satisfied. [1]

The RVM solves all the problems mentioned above. The idea is to associate a prior to each weight (governed by a hyper-parameter for each weight), which are estimated from the data. As a result, fewer kernel functions are required than for SVMs, the error is in general smaller for RVMs than SVMs and the number of relevance vectors needed (similar to support vectors in SVMs) is smaller. On the contrary, it takes longer time to train an RVM than an SVM. [1]

II. Method

RVMs can be thought of as SVMs that use a Bayesian treatment. Instead of providing direct answers, RVMs return probabilistic predictions. We start of by computing the hyperparameters α, σ^2 of the priors for each weight (all of the priors being independent) based on the data. Expectation-maximization (EM) method is used to find optimal values of α, σ^2 (i.e. those values that maximize the *evidence*). This is what constitutes the learning process.

A. Regression

Regression with RVMs consists of three main steps: initialization, learning and inference.

1) Assigning a prior to each weight (initialization): Priors control the importance of a given basis function. The first step is to assign a prior to each weight. Note, the hyperparameters of each weight are independent of each other [1]. The parameter α^{-1} is one of the parameters that we aim to optimize later on. The other parameter is the mean μ, although it is set to zero in this step.

The prior of weight i is of the form $p(w_i|\alpha_i) = \mathcal{N}(0, \alpha_i^{-1})$. When we have multiple data points, the prior $p(\boldsymbol{w}|\boldsymbol{\alpha})$ is simply the product of all the priors of each weight.

[1] In contrast to SVMs, where a single shared hyperparameter is used [2].

2) Optimizing the hyperparameters (learning):

B. Classification

III. Result

IV. Discussion

V. References

FIX REFERENCES

1 Original article (fix ref).

2 http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf