Fixed-Size LS-SVM

The LS-SVM formulation solves a linear system under a least squares cost function. Although the LS-SVM dual formulation is quite advantageous when working with large dimensional input spaces, or when the dimension of the input space is larger than the sample size, it requires the resolution of a linear system with as many unknowns as the number of data points, N. This situation has an obvious drawback when N is too large, and in such case the direct application of this method becomes prohibitive. However, the primal-dual structure of the LS-SVM can be exploited further. It is possible to compute an approximation of the nonlinear mapping *\_* to perform the estimations directly in primal space; furthermore, it is possible to compute a sparse approximation by using only a subsample of selected Support Vectors from the dataset. Explicit expressions for *\_* can be obtained by means of an eigenvalue decomposition of the kernel matrix.

The approximation of the feature space is based on a fixed subset of data-points. One way to select this fixed-size set is to optimize the entropy criterion of the subset. As an example, we run the *fixedsizescript*1 script and obtain the Figure 13. The script chooses an optimal subset of points to be used for fixed size ls-svm according to the quadratic Renyi entropy for a kernel based estimator. The script is run for a specific value of the hyper-parameter of the RBF kernel (*\_*2 = 1). Figure 14 compares the same results for varying *\_*2 (0.01, 0.1, 0.5, 10).

**Sigma2-0.05**

**Sigma2-0.1**

**Sigma2-1.00**

**Sigma2-5.00**

**Sigma2-10.00**

****

****

****

****

We clearly see that the hyper-parameter *\_*2 has an influence on chosen (optimal point). As *\_*2 increases, the optimal support vectors are more spread accross the entire cloud of points and less concentrated in the center. For higher values of *\_*2, the chosen points are more situated at the boundaries of the cloud defined by all points.

The concept of entropy can be broadly defined as the expected value (average) of the information contained in a message (the notion of message is very general, and is here applicable to a set of points). Kernel density estimation is a non-parametric way to estimate the probability density function of a random variable (here, RBF kernel is used).

For fixed size ls-svm, the selection of subsample of size *M* (*M* is here chosen arbitrarly), theinitial support vectors, is done prior to the estimation of the model, and the final performance of the model can depend on the quality of the initial selection. It is possible to take a random selection of *M* datapoints and use them to build the approximation of the nonlinear mapping *\_*, or it is possible to use a more optimal selection. External criteria such as entropy maximization can be applied for an optimal selection of the subsample. In this case, given a fixed-size *M*, the aim is to select the support vectors that maximize the quadratic Renyi entropy.

*HR* = −*log*

Z

*p*(*x*)2*dx*

that can be approximated by

Z

˜*p*(*x*)2*dx* = 1

*N*2 **1***T***1**

where is the kernel matrix.

Starting from a random sample of size *M*, it is possible to replace elements of the selected sample by elements of the remaining sample if the entropy is minimized, and iterate this procedure until convergence. In this way, it is possible to obtain a selection of those *M* points that converge to a minimum value of the quadratic entropy.

Therefore, the algorithm used in the *fixedsizescript*1 script converges to the points that

‘summarizes’ the best the kernel density estimation (i.e. have the highest probability in the KDE). It is important to choose an appropriate value of the hyper-parameter. This can be done using cross-validation based on quadratic Renyi entropy.

Given this optimized subset, the feature space mapping can be reconstructed. Figure 15 shows an example of such reconstruction for a random variable of dimension for which three points have been extracted.



A drawback of LS-SVM models is the lack of sparseness, as nearly all patterns become support vectors. A solution to this problem is To use *L*0-norm. The *L*0-norm counts the number of non-zero elements of a vector. Therefore, minimizing it leads to very sparse models. The Figure 16 compares Fixed-Size LS-SVM with the *L*0 approximation LS-SVM.





The performance (in terms of error) of *L*0 approximation is comparable to the Fixed-Size LS-SVM. This is very good, as the number of support vectors is generally much lower than the Fixed-Size case. The computing time is comparable for both models, but obviously much lower than the LS-SVM case.