DIRAC Training - Report
Finite memory accurate support vector machine

Francesco Orabona
LIRA-Lab, DIST
University of Genoa, Genoa, ITALY 16145
bremen@liralab.it

June 26, 2007

Support Vector Machines (SVMs) are a machine learning method widely employed in, e.g., visual recognition, medical diagnosis and robotic control. One of their most interesting characteristics is that the solution achieved is sparse: a few samples (support vectors) usually account for all the complexity of the classification task. On the other hand the support vectors and the memory requirements are known to grow proportionally with the number of training samples and the difficulty of the problem, without a bound [5]. Clearly this is not acceptable in the framework of continuously learning, because sooner or later the solution would be too big, hence too slow to be used.

To partially solve this problem approximate online training algorithms have been proposed: for each sample they update the solution and forget the sample, not storing the entire training set in memory. But also with these methods there is no guarantee that the number of support vectors, that is the size of the machine, will have an upper bound. In fact typically this is constrained to a prespecified finite size, simply implementing a remove step when this size is exceeded.

The proposed method decouples the concept of "basis" vectors, used to build the classification function, from the samples used to find out the coefficients of the solution itself and in this way to constrain the number of basis vectors without losing much accuracy. Each time a new sample is acquired, a check is done to verify if it is linear independent to the other samples [1]. If not, it is not used as basis vector, but only to evaluate the classification errors. In this way the separating hyperplane is expressed with a small number of (independent) vectors, but at the same time, all the accuracy of the original formulation is retained. Given the ability of SVM to work in a feature space implicitly defined through a kernel function, we have also used a method to check the linear independence through kernel evaluations. Moreover with this method the number of support vectors will always be bounded, regardless of the difficulty of the problem. After each sample the solution is incrementally updated using the method of Keerthi et al. [2], adapted to work in the online setting.

Extensive experimental evaluations have been done in the framework of place recognition on the IDOL2 database [3], with different image features and kernels, demonstrating the efficacy of the method. The number of support vectors in the experiments was up to 3.5 times less with respect to the standard formulation, retaining essentially the same accuracy.

A different interpretation of the selection for the basis vectors can also be stated: each time a new vector is added to the basis, it can be considered to bring new information, that was not present in the past samples. In this sense the independence check can be seen as a sort of "novelty detector".

The result of this work as been accepted in BMVC07 as oral presentation [4].

As future work we plan to extend this work, removing also the need to store all the past samples for the evaluation of the errors.

Acknowledgments

We wish to thank Claudio Castellini and Giorgio Metta for the fruitful discussions.

References

[1] Tom Downs, Kevin E. Gates, and Annette Masters. Exact simplification of support vectors solutions. *Journal of Machine Learning Research*, 2:293–297, 2001.

[2] S. Sathiya Keerthi, Olivier Chapelle, and Dennis DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 8:1–22, 2006.

[3] Jie Luo, Andrzej Pronobis, Barbara Caputo, and Patric Jensfelt. The KTH-IDOL2 database. Technical Report 304, KTH, CAS/CVAP, 2006. Available at `http://cogvis.nada.kth.se/IDOL2/`.

[4] Francesco Orabona, Claudio Castellini, Barbara Caputo, Jie Luo, and Giulio Sandini. Indoor place recognition using online independent support vector machines. In *Accepted in BMVC07*, 2007.

[5] I. Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003.