Hyperplane Folding – a Way to Increase the Margin in Support Vector Machines

Lars Lundberg, Håkan Lennerstad, Eva Garcia-Martin, Niklas Lavesson, and Veselka Boeva

Blekinge Institute of Technology,
SE-371 79 Karlskrona, Sweden
lars.lundberg@bth.se

Hyperplane folding: We present a method, called hyperplane folding, that increases the margin in a linearly separable binary dataset by replacing the support vector machine (SVM) [1] hyperplane with a set of hinging hyperplanes. The proposed approach has two separate phases: dimension reduction and the actual folding.

The first phase of the method reduces the dimension of the problem from n to two dimensions in a sequence of n-2 steps. In each such step a pair of support vectors is selected and the dataset is rotated so that the line that goes through these two support vectors becomes parallel with the axis of the highest dimension, i.e., these two support vectors have the same projection on the hyperplane defined by the lower dimensions. For instance, in a 3-dimensional case, the dataset is rotated so that the line that goes through the two support vectors becomes parallel with the z-axis, and, as result of this, the two support vectors are projected on the same point in the xy-plane. After this we can ignore dimension n and we are now left with an (n-1)-dimensional space with one less support vector. When the 2-dimensional case is reached we can perform the actual folding. Based on the location of the support vectors in the 2-dimensional case, the method splits the dataset into two parts. A SVM is obtained for each of these parts. Each of these two parts has a line that separates the data points. We calculate the intersection and angle between these two lines and, based on this, we rotate one part of the dataset to align the two straight lines and then merges the two parts again. We then expand the data set back to the original n dimensions. This procedure increases the margin for each folding operation as long as the margin is smaller than half of the shortest distance between any pair of data points from the two different classes. We develop an algorithm for the general case with n-dimensional data points. The method can use any standard SVM implementation plus some additional basic manipulation of the data points, i.e., splitting, rotating and merging. An initial evaluation of the algorithm for 2-dimensional case has been conducted. The obtained results have shown that the margin indeed increases and in addition, this improves the classification accuracy of the SVM. For future work, we aim to pursue further evaluation and validation of the proposed hyperplane folding method on richer n-dimensional data sets.

Related work and contribution: There has been work on different types of piecewise linear classifier based on the SVM concept. These methods split the separating hyperplane into a set of hinging hyperplanes [4]. In [5], the authors define an algorithm that uses hinging hyperplanes to separate nonintersecting classes with a multiconlitron. The multiconlitron method cannot benefit directly from implementations of SVMs. Conlitrons and multiconlitrons need to be constructed with new and complex techniques [3]. The hyperplane folding approach presented here is a direct extension of the standard SVM. Soft margin extensions are relatively direct. As a consequence, hyperplane folding can benefit from existing SVM implementations. A piecewise linear SVM classifier is presented in [2]. That method splits the feature space into a number of polyhedra and calculates one hinging hyperplane for each such polyhedron. Some divisions of the feature space will increase the margin in hard margin SVMs. However, unless one has detailed domain knowledge, there is no way to determine which polyhedra to select to improve the margin. The authors recommend basic approaches like equidistantly dividing the input space into hypercubes or using random sizes and shapes for the polyhedra. Based on the support vectors, the hyperplane folding method splits the feature space into two parts (i.e., into two polyhedra) in each iteration. Without any domain knowledge, our method guarantees that the split results in an increase of the margin (except for very special cases). As discussed above, hyperplane folding can directly benefit from existing SVM implementations, which is not the case for the method presented by [2].

References

1. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning 20(3) (1995) 273–297
2. Huang, X., Mehrkanoon, S., Suykens, J.A.: Support vector machines with piecewise linear feature mapping. Neurocomputing 117 (2013) 118–127
3. Li, Y., Leng, Q., Fu, Y., Li, H.: Growing construction of conlitron and multiconlitron. Knowledge-Based Systems 65 (2014) 12–20
4. Wang, S., Sun, X.: Generalization of hinging hyperplanes. IEEE Transactions on Information Theory 51(12) (2005) 4425–4431
5. Yujian, L., Bo, L., Xinwu, Y., Yaozong, F., Houjun, L.: Multiconlitron: A general piecewise linear classifier. IEEE Transactions on Neural Networks 22(2) (2011) 276–289