Perceptron-Like Performance for Intersections of Halfspaces

Adam R. Klivans[1] and Rocco A. Servedio[2]

[1] Harvard University
[2] Columbia University

Given a set of examples on the unit ball in $\mathbf{R}^n$ which are labelled by a halfspace h which has margin ρ (minimum Euclidean distance from any point to the separating hyperplane), the well known Perceptron algorithm finds a separating hyperplane. The Perceptron Convergence Theorem (see e.g. [2]) states that at most $4/\rho^2$ iterations of the Perceptron update rule are required, and thus the algorithm runs in time $O(\frac{n}{\rho^2})$.

Our question is the following: is it possible to give an algorithm which has Perceptron-like performance, i.e. $\text{poly}(n, \frac{1}{\rho})$ runtime, for learning the intersection of two halfspaces with margin ρ? We say that a concept c has margin ρ with respect to a set of points $X \subset \mathbf{R}^n$ if

$$\rho = \min\{\|z - y\| : z \in X, y \in \mathbf{R}^n, c(z) \neq c(y)\}/\|X\|.$$

Here $\|X\|$ denotes $\max_{z \in X} \|z\|$. Note that for the case of a single halfspace where all examples lie on the unit ball, this definition of margin is simply the minimum Euclidean distance from any example to the separating hyperplane as stated above.

The desired learning algorithm need not output an intersection of halfspaces as its hypothesis; any reasonable hypothesis class (which gives an online or PAC algorithm with the stated runtime) is fine.

Motivation: This is a natural restricted version of the more general problem of learning an intersection of two arbitrary halfspaces with no condition on the margin, which is a longstanding open question that seems quite hard (for this more general problem no learning algorithm is known which runs in time less than $2^{O(n)}$). Given the ubiquity of margin-based approaches for learning a single halfspace, it is likely that a solution to the proposed problem would be of significant practical as well as theoretical interest. As described below it seems plausible that a solution may be within reach.

Current status: The first work on this question is by Arriaga and Vempala [1] who gave an algorithm that runs in time $n \cdot \text{poly}\left(\frac{1}{\rho}\right) + \left(\frac{1}{\rho}\right)^{\tilde{O}(1/\rho^2)}$, i.e. polynomial in n but exponential in $1/\rho$. Their algorithm randomly projects the examples to a low-dimensional space and uses brute-force search to find a consistent intersection of halfspaces. Recently we gave an algorithm [3] that runs in time $n\left(\frac{1}{\rho}\right)^{O(\log(1/\rho))}$, i.e. polynomial in n and quasipolynomial in $\frac{1}{\rho}$. Our algorithm also uses random projection as a first step, but then runs the kernel

Perceptron algorithm with the polynomial kernel to find a consistent hypothesis as opposed to using brute-force search. We show that low degree *polynomial threshold functions* can correctly computing intersections of halfspaces with a margin (in a certain technical sense — see [3] for details); this implies that the degree of the polynomial kernel can be taken to be logarithmic in $1/\rho$, which yields our quasipolynomial runtime dependence on ρ. Can this quasipolynomial dependence on the margin ρ be reduced to a polynomial?

References

[1] R. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 616–623, 1999.

[2] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, 2000.

[3] A. Klivans and R. Servedio. Learning intersections of halfspaces with a margin. In Proceedings of COLT 2004.