Threshold Decision Lists

Martin Anthony
Department of Mathematics
LSE

We consider the use of threshold decision lists for classifying data into two classes.

- has a natural geometrical interpretation

- can be appropriate for an iterative approach to data classification, in which some points of the data set are classified, are then removed from consideration, and the procedure iterated until all points are classified.

We apply techniques from probabilistic learning theory to analyse theoretically the generalization properties of data classification techniques based on the use of threshold decision lists.

Extension/explanation of a data set

Suppose given some data points in $\mathbb{R}^n$, each classified as either *positive* (labeled 1) or *negative* (labeled 0). The data points, together with the positive/negative classifications will be denoted bfs.

An extension of $\mathbf{s}$ is a Boolean function f such that f agrees with $\mathbf{s}$.

The aim is to find an extension of f which will be a good 'generalization' of the data.

Many extensions of a given data set. Finding one of a particular type is a natural and central problem in machine learning and data mining.

We analyse the 'generalization accuracy' of *threshold decision list* extensions. In doing so, we employ a probabilistic framework that has been used extensively in the modelling of machine learning.

Decision lists

Suppose K is a set of functions from $\mathbb{R}^n$ to $\{0,1\}$.

$f : \mathbb{R}^n \to \{0,1\}$ is a decision list based on K if there are $f_i \in K$ and $c_i \in \{0,1\}$ such that

`if` $f_1(y) = 1$ `then` $f(y) = c_1$

`else if` $f_2(y) = 1$ `then` $f(y) = c_2$

$\ldots$

$\ldots$

`else if` $f_r(y) = 1$ `then` $f(y) = c_r$

`else` $f(y) = 0$.

We write

$$f = (f_1, c_1),\ (f_2, c_2), \ldots,\ (f_r, c_r),$$

Each f_j is a test (or a query) and the pair (f_j, c_j) is a term of the decision list.

Threshold functions

$t : \mathbb{R}^n \to \{0, 1\}$ is a threshold function if there are $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ such that

$$t(x) = \begin{cases} 1 & \text{if } \langle w, x \rangle \geq \theta \\ 0 & \text{if } \langle w, x \rangle < \theta, \end{cases}$$

where $\langle w, x \rangle = w^T x$ is the inner product of w and x.

Thus,

$$t(x) = \text{sgn}(\langle w, x \rangle - \theta).$$

Threshold decision lists

A decision list in which the tests are threshold functions is a threshold decision lists. Studied by Marchand and colleagues, who called them neural decision lists, and Turan and Vatan, who called them linear decision lists.

Formally, a threshold decision list

$$f = (f_1, c_1), (f_2, c_2), \ldots, (f_r, c_r)$$

has each $f_i : \mathbb{R}^n \to \{0, 1\}$ of the form

$$f_i(x) = \text{sgn}(\langle w, x \rangle - \theta),$$

where $\text{sgn}(x) = 1$ if $x \geq 0$ and $\text{sgn}(x) = 0$ if $x < 0$.

The value of f on $y \in \mathbb{R}^n$ is $f(y) = c_j$ if $j = \min\{i : f_i(y) = 1\}$ exists, or 0 otherwise (that is, if there are no j such that $f_j(y) = 1$).

Geometrical motivation: Given $\mathbf{s}$, it's unlikely that the positive and negative points can be separated by a hyperplane. But we can use a hyperplane to separate off a set of points all having the same classification. These points can then be removed from consideration and the procedure iterated until no points remain.

This procedure is similar in nature to one of Jeroslow, but at each stage in his procedure, only positive examples may be 'chopped off' (not positive *or* negative).

Example: Suppose $\mathbf{s}$ is all of $\{0,1\}^n$, labeled according to parity. (So the classification is 1 precisely when the point has an odd number of ones.)

Best we can do at the first stage is chop off one of the points (since the neighbors of any point have the opposite classification). Suppose we chop off the origin. We may take the first hyperplane to be $x_1 + x_2 + \cdots + x_n = 1/2$.

We then ignore the origin and consider the remaining points.

We can next chop off all neighbors of the origin, all the points which have precisely one entry equal to 1. All of these are positive points and the hyperplane $x_1 + x_2 + \cdots + x_n = 3/2$ will separate them from the other points.

These points are then deleted from consideration. We can continue in this manner.

The procedure iterates n times, and at stage i in the procedure we 'chop off' all data points having precisely $(i-1)$ ones, by using the hyperplane $x_1 + x_2 + \cdots + x_n = i - 1/2$, for example. (These hyperplanes are in fact all parallel, but this is not in general possible.)

What we end up with (assuming n even) is a TDL representing parity, with terms

$(\mathrm{sgn}(\langle -1, x\rangle + 1/2), 0),$

$(\mathrm{sgn}(\langle -1, x\rangle + 3/2), 1),$

$\vdots$

$(\mathrm{sgn}(\langle -1, x\rangle + n - 3/2), 0),$

$(\mathrm{sgn}(\langle -1, x\rangle + n - 1/2), 1).$

As indicated, this 'chopping' procedure constructs a threshold decision list extension of the data set.

The Jeroslow method results in a restricted form of decision list, in which all terms are of the form $(f_i, 1)$. But this is just the disjunction

$$f_1 \vee f_2 \vee \ldots .$$

The problem of decomposing a Boolean function into the disjunction of threshold functions has been considered by Hammer, Ibaraki and Peled, and by Zuev and Lipkin.

Hammer *et al.* defined the threshold number of a Boolean function to be the minimum s such that f is a disjunction of s threshold functions, and they showed that there is an increasing function with threshold number $\binom{n}{n/2}/n$.

Zuev and Lipkin showed that almost all increasing functions have this order of threshold number, and that almost all Boolean functions have a threshold number that is $\Omega(2^n/2)$ and $O(2^n \ln n/n)$.

Note that Jeroslow's method requires 2^{n-1} iterations in the parity-based Example given above, since at each stage it can only 'chop off' one positive point.

The practicalities of the chopping procedure have been investigated by Marchand *et al.*, who derive a greedy heuristic for constructing a sequence of 'chops'. This relies on an incremental heuristic for the NP-hard problem of finding at each stage a hyperplane that chops off as many remaining points as possible.

Multilevel threshold functions

In the parity example, the hyperplanes of the TDL were parallel. By demanding that the hyperplanes are parallel, we obtain a special subclass of TDLs, known as the multilevel threshold functions (or multithreshold functions).

An s-level threshold function f is one representable by a TDL of length at most s with the test hyperplanes parallel to each other.

Equivalently, f is an s-level threshold function if there is a weight-vector $w = (w_1, w_2, \ldots, w_n)$ such that

$$f(x) = F\left(\sum_{i=1}^{n} w_i x_i\right),$$

where the function $F : \mathbb{R} \to \{0, 1\}$ is piecewise constant with at most $s + 1$ pieces. (Without loss, we may suppose that the classifications assigned to points in neighboring regions are different.)

This method of classification is reasonably powerful. For example, Bohossian and Bruck observed that any Boolean function is a 2^n-level threshold function, an appropriate weight-vector being $w = (2^{n-1}, 2^{n-2}, \ldots, 2, 1)$.

Generalization from random data

Recall: an extension of a labeled data set $\mathbf{s}$ is f agreeing with the classifications of the points in $\mathbf{s}$, and a partial extension is one agreeing with at least some proportion of the classification in $\mathbf{s}$.

If a particularly simple type of extension (or a good partial extension) to a fairly large data set can be found we might expect, given the success of this simple function in explaining the large data set, that this extension will perform well on 'most' unseen data.

We use the 'PAC' probabilistic model of learning.

We assume that the (partial) extensions produced all belong to a particular class, H, of functions, known as the *hypothesis space*. The choice of H might reflect either our belief about the mechanism by which the data points are labeled or our intention only to accept simple types of explanation of the data, even if these do not match the data exactly.

Following a form of Valiant's PAC model of computational learning theory, we assume that the labeled data points (x, b) (where $x \in \mathbb{R}^n$ and $b \in \{0, 1\}$) have been generated randomly (perhaps from some larger corpus of data) according to a fixed probability distribution P defined on $Z = \mathbb{R}^n \times \{0, 1\}$. Thus, if $\mathbf{s}$ is of length m, we may regard $\mathbf{s}$ as an element of Z^m, drawn randomly according to the product probability distribution P^m.

Given any function $f \in H$, we measure how well f extends the data set $\mathbf{s}$ through its sample error $\mathrm{er}_{\mathbf{s}}(f) = m^{-1}|\{(x,b) \in \mathbf{s} : f(x) \neq b\}|$ (which is the proportion of points of $\mathbf{s}$ incorrectly classified by f) and we measure how well f performs on further examples by means of its error

$$\mathrm{er}(f) = P\left(\{(x,b) \in Z : f(x) \neq b\}\right),$$

the probability that a further randomly drawn labeled data point would be incorrectly classified by f.

What we want is some guarantee that the sample error $\mathrm{er}_{\mathbf{s}}(f)$ is a good approximation to the error $\mathrm{er}(f)$ for all f, so that an f with small sample error will likely have small error and therefore be a good model of the data labels.

We can do this using results of Vapnik and Chervonenkis, and some combinatorics.

Growth function

To use results from statistical learning theory, we use the growth function. Suppose H is a set of functions from $X = \mathbb{R}^n$ to $\{0, 1\}$. Let $\Pi_H : \mathbb{N} \to \mathbb{N}$ be given by

$$\Pi_H(m) = \max\{|H|_S| : S \subseteq X, |S| = m\},$$

where $H|_S$ denotes H restricted to domain S. Note that $\Pi_H(m) \leq 2^m$ for all m. The function Π_H is known as the growth function of H, and it measures how expressive the hypothesis class H is.

Bounding error

We employ are the following bound, due to Vapnik and Chervonenkis.

For any $\epsilon \in (0, 1)$,

$$P^m \left(\{ \mathbf{s} : \forall\, f \in H,\ \mathrm{er}(f) < \mathrm{er}_{\mathbf{s}}(f) + \epsilon \} \right)$$

is greater than

$$1 - 4\, \Pi_H(2m)\, e^{-m\epsilon^2/8}.$$

Thus, we can obtain (probabilistic) bounds on the error $\mathrm{er}(f)$ of a (partial) extension from a class H when we know something about the growth function of H.

Growth function bounds

We start with general threshold decision lists. We consider the set of threshold decision lists on $\mathbb{R}^n$ with at most some number s of terms. (So, the length of the list is no more than s.)

Theorem Let H be the set of threshold decision lists on $\mathbb{R}^n$ with at most s terms, where $n, s \in \mathbb{N}$. Then, for $m > n$,

$$\Pi_H(m) < 4^s \left(\frac{e(m-1)}{n} \right)^{ns}.$$

Proof: Let $S \subseteq \mathbb{R}^n$, $|S| = m$. Two decision lists $f = (f_1, c_1), \ldots, (f_s, c_s)$, $g = (g_1, d_1), \ldots, (g_s, d_s)$ in H. (We can assume both are of length exactly s by padding.)

If

(i) $c_i = d_i$ for each i and

(ii) $f_i(x) = g_i(x)$ for all $x \in S$,

then f and g are equal on S.

For fixed i, the condition in (ii) is an equivalence relation among functions in K, and the number of equivalence classes is $|K|_S|$ where K is the set of threshold functions. This is bounded by $\Pi_K(m)$, which, it is well known, is bounded above as follows:

$$\Pi_K(m) = 2 \sum_{i=0}^{n} \binom{m-1}{k} < 2 \left(\frac{e(m-1)}{n} \right)^n .$$

So

$$|H|_S| \leq 2^s \left(2\left(\frac{e(m-1)}{n}\right)^n\right)^s .$$

Here, the first 2^s factor corresponds to the number of possible sequences of c_i and the remaining factor bounds the number of ways of choosing an equivalence class (with respect to S) of threshold functions, for each i from 1 to s.

It can be shown that any threshold decision list is a threshold function of threshold functions. But this is nothing more than a two-layer threshold network. So another way of bounding the growth function of threshold decision lists is to use this fact in combination with some known bounds for the growth functions of linear threshold networks. This gives a similar, though slightly looser, upper bound.

Growth of multithreshold functions

Bounding the growth function of the class of s-level threshold functions has been considered in a number of papers.

Takiyama (1985) published an upper bound, but Olafsson and Abu-Mostafa (1988) showed it to be incorrect, and gave the following upper bound:

Theorem Suppose H is class of s-level threshold functions. Then

$$\Pi_H(m) \leq \sum_{l=0}^{s} \binom{m-1}{l} \sum_{i=0}^{n-1} \binom{\binom{m}{2}}{i}.$$

Ngom *et al.* (1999) subsequently claimed to have proved that

$$\Pi_H(m) \leq \binom{m-2}{s-1} \sum_{i=0}^{n} \binom{m-1}{i}.$$

However, this is incorrect.

The following bound improves Olafsson and Abu-Mostafa. It agrees with the known (tight) result for the case $s = 1$; and is, for fixed s and n, tight to within a constant.

Theorem Let H be the set of s-level threshold functions on $\mathbb{R}^n$. Then

$$\Pi_H(m) \leq 2 \sum_{i=0}^{n+s-1} \binom{sm}{i}.$$

Interesting combinatorial question: can we get good bounds on the VC-dimension?

Theorem Let $\delta \in (0,1)$. Then, with probability at least $1-\delta$: If f is a TDL with at most s terms, then, for $m > n$, the error $\mathrm{er}(f)$ of f and its sample error on $\mathbf{s}$, $\mathrm{er}_{\mathbf{s}}(f)$ are such that

$$\mathrm{er}(f) < \mathrm{er}_{\mathbf{s}}(f) + \epsilon_1(m,\delta),$$

where $\epsilon_1(m,\delta)$ is

$$\sqrt{\frac{8}{m}\left(2s \ln 2 + ns \ln\left(\frac{e(2m-1)}{n}\right) + \ln\left(\frac{8}{\delta}\right)\right)}.$$

Theorem Let $\delta \in (0,1)$. Then, with probability at least $1-\delta$: If f is an s-level threshold function, then, for $m \geq n+s$,

$$\mathrm{er}(f) < \mathrm{er_s}(f) + \epsilon_2(m,\delta),$$

where $\epsilon_2(m,\delta)$ is

$$\sqrt{\frac{8}{m}\left((n+s-1)\ln\left(\frac{2emS}{n+s-1}\right)+\ln\left(\frac{8}{\delta}\right)\right)}.$$

If there is f that is an extension of $\mathbf{s}$, with no sample errors—in particular, if the labels correspond to a threshold decision list of length at most s, or to an s-level threshold function—then tighter bounds can be obtained.

But one does not necessarily know *a priori* how many terms a suitable threshold decision list will have. The following results, in which s is not prescribed in advance, are therefore potentially more useful.

Theorem With the same notations,

If f is a threshold decision list, then

$$\mathrm{er}(f) < \mathrm{er_s}(f) + \epsilon_3(m, \delta),$$

where $\epsilon_3(m, \delta)$ is

$$\sqrt{\frac{8}{m}\left(2s \ln 2 + ns \ln\left(\frac{e(2m-1)}{n}\right) + \ln\left(\frac{14s^2}{\delta}\right)\right)},$$

for $m \geq n + s$, where s is the number of terms of f.

If f is a multilevel threshold function, then

$$\mathrm{er}(f) < \mathrm{er_s}(f) + \epsilon_4(m, \delta),$$

where $\epsilon_4(m, \delta)$ is

$$\sqrt{\frac{8}{m}\left((n+s-1)\ln\left(\frac{2emes}{n+s-1}\right) + \ln\left(\frac{14s^2}{\delta}\right)\right)},$$

for $m \geq n + s$, where s is the number of levels (planes) of f.

Large margins

As is often the case, better generalization bounds can be given if we consider 'margins'. Learning with a large margin is central to SVMs, for instance.

Idea is: if a classifier has managed to achieve a 'wide' separation between (most of) the points of different classification, then this indicates that it is a good classifier, and it is possible that a better (that is, smaller) generalization error bound can be obtained.

Classical example:linear separation. If we have found a linear threshold function that classifies the points of a sample correctly *and* the points of opposite classifications are separated by a wide margin (so that the hyperplane gives a 'definitely' correct classification), then we might have a better classifier of future, unseen, points than one which 'merely' separates the points correctly, but with a small margin.

Large margin bounds for TDLs

Here, we suppose (following Bennett et al.) that, for each example in the training sample, each plane clears all examples by a certain margin (and not just the examples it 'deals with').

Suppose h is a threshold decision list, with s terms, and suppose that the tests are threshold functions $t_1, t_2, \ldots, t_s$. Suppose t_i is represented by weight vector w_i and threshold θ_i. Then h classifies the labelled example (x, b) (correctly, and) with margin $\gamma > 0$ if $h(x) = b$ and all $1 \leq i \leq s$, $|\langle w_i, x \rangle - \theta_i| \geq \gamma$.

Given $\mathbf{s} = ((x_1, b_1), \ldots, (x_m, b_m))$, the error of h on $\mathbf{s}$ at margin γ, denoted $\mathrm{er}_{\mathbf{s}}^{\gamma}(h)$, is proportion of labelled examples in $\mathbf{s}$ that are *not* classified by h with margin γ.

So, $\mathrm{er}_{\mathbf{s}}^{\gamma}(h)$ is the fraction of the sample points that are either misclassified by h, or are classified correctly but are distance less than γ from one of the planes.

Can allow different margins for each test:

Given $\Gamma = (\gamma_1, \gamma_2, \ldots, \gamma_s)$, say h classifies (x, b) with margin Γ if $h(x) = b$ and, for all $1 \leq i \leq s$, $|\langle w_i, x \rangle - \theta_i| \geq \gamma_i$.

Then $\mathrm{er}_{\mathbf{s}}^{\Gamma}(h)$ is the proportion of labelled examples in s not classified with margin Γ.

Following a method used by Bennett et al, together with covering number bounds of Zhang, we can get generalization error bounds that are better than in the non-margin case (and improve upon results of Bennett et al. on 'perceptron decision trees').

Assume, for simplicity, $R \geq 1$ and $\gamma_i \leq 1$.

Theorem Let $Z = B_R \times \{0, 1\}$, where $B_R = \{x \in \mathbb{R}^n : \|x\| \leq R\}$. Fix $s \in \mathbb{N}$ and let H be the set of all threshold decision lists with s terms, defined on B_R. Let $\gamma_1, \gamma_2, \ldots, \gamma_s \in (0, 1]$ be given. Then, with probability at least $1 - \delta$, for $s \in Z^m$: if $h \in H$ and $\Gamma = (\gamma_1, \gamma_2, \ldots, \gamma_s)$, then

$$\mathrm{er}_P(h) < \mathrm{er}_s^{\Gamma}(h) + \sqrt{\frac{8}{m}\left(576\, R^2 D(\Gamma) \log_2(8m) + \ln\left(\frac{2}{\delta}\right)\right)},$$

where $D(\Gamma) = \Sigma_{i=1}^{s}(1/\gamma_i^2)$.

So, compared with the non-margin analysis (simplifying to $\gamma_i = \gamma$), we're replacing n with R^2/γ^2.

Versions of this can be given in which either the observed margin error is zero and/or (quite technical) the margins and k are not prescribed in advance.

Proof uses ‘symmetrization’, and a covering number bound of Zhang.

The key observation is that, as in many proofs in learning theory, probability of large error can be related to a sample-based probability:

If

$$Q = \{\mathbf{s} \in Z^m : \exists h \in H \text{ with } \mathrm{er}_P(h) \geq \mathrm{er}_{\mathbf{s}}^{\ulcorner}(h) + \epsilon\}$$

and

$$R = \{(s, s') \in Z^m \times Z^m : \exists h \in H \text{ with } \mathrm{er}_{s'}(h) \geq \mathrm{er}_{s}^{\ulcorner}(h) + \epsilon/2\},$$

then $P^m(Q) \leq 2\, P^{2m}(R)$.

Probability of R is then bounded by using permutations and taking an empirical cover.

Large margin bounds for Multithreshold functions

Assume all $\gamma_i = \gamma$. (Not a real restriction.) The result for TDLs works with a covering for each of the s terms of the list. Instead, for Multithreshold functions, can work more directly with a covering of the set of functions.

Theorem Suppose $Z = B_R \times \{0,1\}$, where $B_R = \{x \in \mathbb{R}^n : \|x\| \leq R\}$. Fix $s \in \mathbb{N}$ and let H be the set of all s-level threshold functions defined on domain B_R. Let P be any probability distribution on Z, and suppose $\gamma \in (0,1]$ and $\delta \in (0,1)$. Then, with P^m-probability at least $1-\delta$, $\mathrm{er}_P(h) < \mathrm{er}_s^{\gamma}(h) +$

$$\sqrt{\frac{8}{m}\left(\frac{1152R^2}{\gamma^2}\log_2(9m) + s\ln\left(\frac{10R}{\gamma}\right) + \ln\left(\frac{2}{\delta}\right)\right)}.$$

The generalization error bound from the TDL bound is worse than this more particular one. Suppressing constants,

$$\frac{R^2 k}{\gamma^2} \ln m$$

is replaced by

$$\frac{R^2}{\gamma^2} \ln m + k \ln \left(\frac{R}{\gamma} \right).$$

Representing BFs by threshold networks

Using threshold decision lists gives us a way of representing Boolean functions by threshold networks, distinct from the obvious one based on a DNF.

Interested in linear threshold networks with one hidden layer. This has n inputs and some number, k, of threshold units in a single hidden layer, together with one output threshold unit.

If output node computes threshold function given by weight vector $\beta \in \mathbb{R}^k$ and threshold ϕ, and the threshold function computed by hidden node i is $f_i \leftarrow [w^{(i)}, \theta^{(i)}]$, then the network as a whole computes $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(y) = 1 \iff \sum_{i=1}^{k} \beta_i f_i(y) \geq \phi.$$

So,

$$f(y_1 y_2 \dots y_n) = \text{sgn}\left(\sum_{i=1}^{k} \beta_i \, \text{sgn}\left(\sum_{j=1}^{n} w_j^{(i)} y_j - \theta^{(i)}\right) - \phi\right),$$

where $\text{sgn}(x) = 1$ if $x \geq 0$ and $\text{sgn}(x) = 0$ if $x < 0$.

The *state* of the network is the (concatenated) vector

$$\omega = (w^{(1)}, \theta^{(1)}, w^{(2)}, \theta^{(2)}, \ldots, w^{(k)}, \theta^{(k)}, \beta, \phi) \in \mathbb{R}^{nk+2k+1}.$$

A fixed network architecture of this type (that is, fixing n and k), computes a parameterised set of functions $\{f_\omega : \omega \in \mathbb{R}^{nk+2k+1}\}$. In state ω, the network computes the function $f_\omega : \{0,1\}^n \to \{0,1\}$.

Standard approach

ϕ a DNF formula for the BF f, $\phi = T_1 \vee T_2 \vee \cdots \vee T_k$, where $T_i = \left(\wedge_{j \in P_i} u_j\right) \wedge \left(\wedge_{j \in N_i} \bar{u}_j\right)$. Form network with k hidden units, one corresponding to each term of the DNF. Label these units $1, 2, \ldots, k$ and set the weight vector $w^{(i)}$ as:

$w_j^{(i)} = 1$ if $j \in P_i$,

$w_j^{(i)} = -1$ if $j \in N_i$, and $w_j^{(i)} = 0$ otherwise.

Take the threshold $\theta^{(i)}$ to be $|P_i|$.

Set weight on connection between each hidden unit and output to 1, and threshold output at 1/2.

Using threshold decision lists

Observation: a threshold decision list is a threshold function of threshold functions.

$$f = (f_1, c_1), (f_2, c_2), \ldots, (f_k, c_k)$$

where $f_i \leftarrow [w^{(i)}, \theta^{(i)}]$. Consider threshold network with n inputs, k threshold units in a single hidden layer, and one output. Let ω be the state:

$$\omega = (w^{(1)}, \theta^{(1)}, w^{(2)}, \theta^{(2)}, \ldots, w^{(k)}, \theta^{(k)}, \beta, 1),$$

where

$$\beta = (2^{k-1}(2c_1 - 1), 2^{k-2}(2c_2 - 1), \ldots, 2(2c_{k-1} - 1), (2c_k - 1)).$$

Then $f_\omega = f$.

Parity shows that the representation arising from TDLs can differ considerably from the standard one: n hidden units rather than 2^{n-1}.

If T is any term of a DNF formula, then T can be represented by a threshold function. So if $\phi = T_1 \vee T_2 \vee \cdots \vee T_k$ is a DNF representation of the function f, then f is also represented by the threshold decision list

$$(T_1, 1), (T_2, 1), \ldots, (T_k, 1).$$

So there is always a threshold decision list representation whose length is no more than that of any given DNF representation of the function.

Conclusions

- Threshold decision lists a powerful pattern classification technique; and special case of parallel planes has been of interest for some time.

- Have analysed generalization error using 'classical' PAC model.

- Can get sometimes-better results by considering margins.

- Interesting implications for representing BFs by threshold networks.