On Parallel Versus Serial Processing: A Computational Study of Visual Search

Eyal Cohen
Department of Psychology
Tel-Aviv University Tel Aviv 69978, Israel
eyalc@devil.tau.ac.il
Eytan Ruppin
Departments of Computer Science & Physiology
Tel-Aviv University Tel Aviv 69978, Israel
ruppin@math.tau.ac.il

Abstract

A novel neural network model of pre-attention processing in visual-search tasks is presented. Using displays of line orientations taken from Wolfe's experiments [1992], we study the hypothesis that the distinction between parallel versus serial processes arises from the availability of global information in the internal representations of the visual scene. The model operates in two phases. First, the visual displays are compressed via principal-component-analysis. Second, the compressed data is processed by a target detector module in order to identify the existence of a target in the display. Our main finding is that targets in displays which were found experimentally to be processed in parallel can be detected by the system, while targets in experimentally-serial displays cannot. This fundamental difference is explained via variance analysis of the compressed representations, providing a numerical criterion distinguishing parallel from serial displays. Our model yields a mapping of response-time slopes that is similar to Duncan and Humphreys's "search surface" [1989], providing an explicit formulation of their intuitive notion of feature similarity. It presents a neural realization of the processing that may underlie the classical metaphorical explanations of visual search.

1 Introduction

This paper presents a neural-model of pre-attentive visual processing. The model explains why certain displays can be processed very fast, "in parallel", while others require slower, "serial" processing, in subsequent attentional systems. Our approach stems from the observation that the visual environment is overflowing with diverse information, but the biological information-processing systems analyzing it have a limited capacity [1]. This apparent mismatch suggests that *data compression* should be performed at an early stage of perception, and that via an accompanying process of *dimension reduction*, only a few essential features of the visual display should be retained. We propose that only parallel displays incorporate global features that enable fast target detection, and hence they can be processed pre-attentively, with all items (target and distractors) examined at once. On the other hand, in serial displays' representations, global information is obscure and target detection requires a serial, attentional scan of local features across the display. Using principal-component-analysis (PCA), our main goal is to demonstrate that neural systems employing compressed, dimensionally reduced representations of the visual information can successfully process only parallel displays and not serial ones. The source of this difference will be explained via variance analysis of the displays' projections on the principal axes.

The modeling of visual attention in cognitive psychology involves the use of metaphors, e.g., Posner's beam of attention [2]. A visual attention system of a surviving organism must supply fast answers to burning issues such as detecting a target in the visual field and characterizing its primary features. An attentional system employing a constant-speed beam of attention [3] probably cannot perform such tasks fast enough and a pre-attentive system is required. Treisman's feature integration theory (FIT) describes such a system [4]. According to FIT, features of separate dimensions (shape, color, orientation) are first coded pre-attentively in a locations map and in separate feature maps, each map representing the values of a particular dimension. Then, in the second stage, attention "glues" the features together conjoining them into objects at their specified locations. This hypothesis was supported using the *visual-search* paradigm [4], in which subjects are asked to detect a target within an array of distractors, which differ on given physical dimensions such as color, shape or orientation. As long as the target is significantly different from the distractors in one dimension, the reaction time (RT) is short and shows almost no dependence on the number of distractors (low RT slope). This result suggests that in this case the target is detected pre-attentively, in parallel. However, if the target and distractors are similar, or the target specifications are more complex, reaction time grows considerably as a function of the number of distractors [5, 6], suggesting that the displays' items are scanned serially using an attentional process.

FIT and other related cognitive models of visual search are formulated on the conceptual level and do not offer a detailed description of the processes involved in transforming the visual scene from an ordered set of data points into given values in specified feature maps. This paper presents a novel computational explanation of the source of the distinction between parallel and serial processing, progressing from general metaphorical terms to a neural network realization. Interestingly, we also come out with a computational interpretation of some of these metaphorical terms, such as feature similarity.

2 The Model

We focus our study on visual-search experiments of line orientations performed by Wolfe et. al. [7], using three set-sizes composed of 4, 8 and 12 items. The number of items equals the number of distractors + target in target displays, and in non-target displays the target was replaced by another distractor, keeping a constant set-size. Five experimental conditions were simulated: (A) - a 20 degrees tilted target among vertical distractors (homogeneous background). (B) - a vertical target among 20 degrees tilted distractors (homogeneous background). (C) - a vertical target among heterogeneous background (a mixture of lines with ± 20, ± 40 , ± 60 , ± 80 degrees orientations). (E) - a vertical target among two flanking distractor orientations (at ± 20 degrees), and (G) - a vertical target among two flanking distractor orientations (± 40 degrees). The response times (RT) as a function of the set-size measured by Wolfe et. al. [7] show that type A, B and G displays are scanned in a parallel manner (1.2, 1.8, 4.8 msec/item for the RT slopes), while type C and E displays are scanned serially (19.7, 17.5 msec/item). The input displays of our system were prepared following Wolfe's prescription: Nine images of the basic line orientations were produced as nine matrices of gray-level values. Displays for the various conditions of Wolfe's experiments were produced by randomly assigning these matrices into a 4x4 array, yielding 128x100 display-matrices that were transformed into 12800 display-vectors. A total number of 2400 displays were produced in 30 groups (80 displays in each group): 5 conditions (A, B, C, E, G) $\times$ target/non-target $\times$ 3 set-sizes (4, 8, 12).

Our model is composed of two neural network modules connected in sequence as illustrated in Figure 1: a PCA module which compresses the visual data into a set of principal axes, and a Target Detector (TD) module. The latter module uses the compressed data obtained by the former module to detect a target within an array of distractors. The system is presented with line-orientation displays as described above.

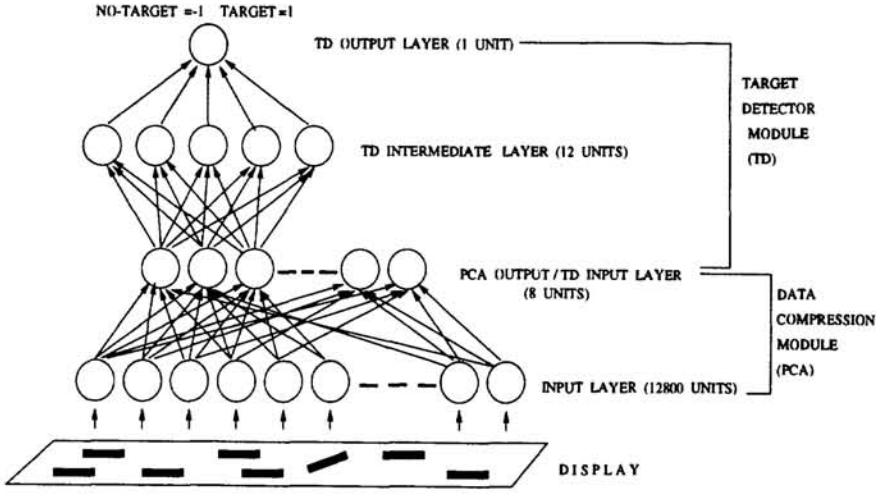

Figure 1: General architecture of the model

For the PCA module we use the neural network proposed by Sanger, with the connections' values updated in accordance with his Generalized Hebbian Algorithm (GHA) [8]. The outputs of the trained system are the projections of the display-vectors along the first few principal axes, ordered with respect to their eigenvalue magnitudes. Compressing the data is achieved by choosing outputs from the first

few neurons (maximal variance and minimal information loss). Target detection in our system is performed by a feed-forward (FF) 3-layered network, trained via a standard back-propagation algorithm in a supervised-learning manner. The input layer of the FF network is composed of the first eight output neurons of the PCA module. The transfer function used in the intermediate and output layers is the hyperbolic tangent function.

3 Results

3.1 Target Detection

The performance of the system was examined in two simulation experiments. In the first, the PCA module was trained only with "parallel" task displays, and in the second, only with "serial" task displays. There is an inherent difference in the ability of the model to detect targets in parallel versus serial displays. In parallel task conditions (A, B, G) the target detector module learns the task after a comparatively small number (800 to 2000) of epochs, reaching performance level of almost 100%. However, the target detector module is not capable of learning to detect a target in serial displays (C, E conditions). Interestingly, these results hold (1) whether the preceding PCA module was trained to perform data compression using parallel task displays or serial ones, (2) whether the target detector was a linear simple perceptron, or the more powerful, non-linear network depicted in Figure 1, and (3) whether the full set of 144 principal axes (with non-zero eigenvalues) was used.

3.2 Information Span

To analyze the differences between parallel and serial tasks we examined the eigenvalues obtained from the PCA of the training-set displays. The eigenvalues of condition B (parallel) displays in 4 and 12 set-sizes and of condition C (serial-task) displays are presented in Figure 2. Each training set contains a mixture of target and non-target displays.

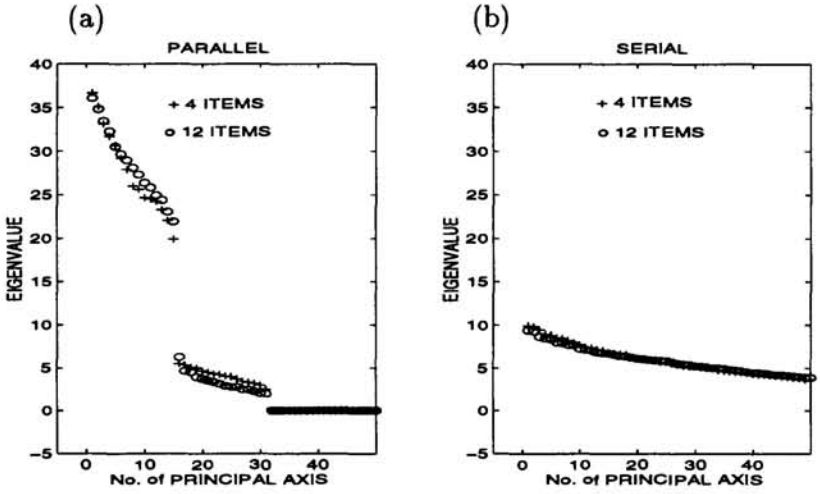

Figure 2: Eigenvalues spectrum of displays with different set-sizes, for parallel and serial tasks. Due to the sparseness of the displays (a few black lines on white background), it takes only 31 principal axes to describe the parallel training-set in full (see fig 2a. Note that the remaining axes have zero eigenvalues, indicating that they contain no additional information.), and 144 axes for the serial set (only the first 50 axes are shown in fig 2b).

As evident, the eigenvalues distributions of the two display types are fundamentally different: in the parallel task, most of the eigenvalues "mass" is concentrated in the first few (15) principal axes, testifying that indeed, the dimension of the parallel displays space is quite confined. But for the serial task, the eigenvalues are distributed almost uniformly over 144 axes. This inherent difference is independent of set-size: 4 and 12-item displays have practically the same eigenvalue spectra.

3.3 Variance Analysis

The target detector inputs are the projections of the display-vectors along the first few principal axes. Thus, some insight to the source of the difference between parallel and serial tasks can be gained performing a variance analysis on these projections. The five different task conditions were analyzed separately, taking a group of 85 target displays and a group of 85 non-target displays for each set-size. Two types of variances were calculated for the projections on the 5th principal axis: The "within groups" variance, which is a measure of the statistical noise within each group of 85 displays, and the "between groups" variance, which measures the separation between target and non-target groups of displays for each set-size. These variances were averaged for each task (condition), over all set-sizes. The resulting ratios Q of within-groups to between-groups standard deviations are: $Q_A = 0.0259$, $Q_B = 0.0587$, and $Q_G = 0.0114$ for parallel displays (A, B, G), and $Q_E = 0.2125$ $Q_C = 0.771$ for serial ones (E, C).

As evident, for parallel task displays the Q values are smaller by an order of magnitude compared with the serial displays, indicating a better separation between target and non-target displays in parallel tasks. Moreover, using Q as a criterion for parallel/serial distinction one can predict that displays with $Q << 1$ will be processed in parallel, and serially otherwise, in accordance with the experimental response time (RT) slopes measured by Wolfe et. al. [7]. This differences are further demonstrated in Figure 3, depicting projections of display-vectors on the sub-space spanned by the 5, 6 and 7th principal axes. Clearly, for the parallel task (condition B), the PCA representations of the target-displays (plus signs) are separated from non-target representations (circles), while for serial displays (condition C) there is no such separation. It should be emphasized that there is no other principal axis along which such a separation is manifested for serial displays.

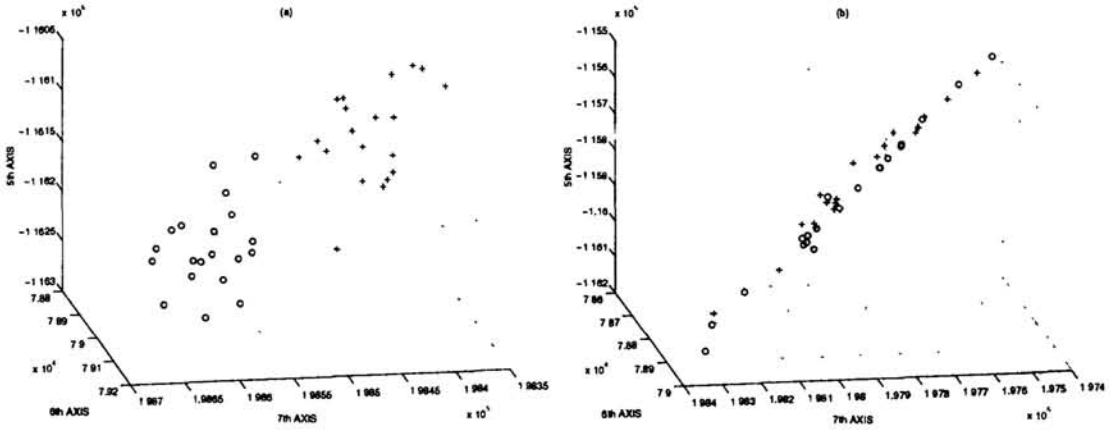

Figure 3: Projections of display-vectors on the sub-space spanned by the 5, 6 and 7th principal axes. Plus signs and circles denote target and non-target display-vectors respectively, (a) for a parallel task (condition B), and (b) for a serial task (condition C). Set-size is 8 items.

While Treisman and her co-workers view the distinction between parallel and serial tasks as a fundamental one, Duncan and Humphreys [5] claim that there is no sharp distinction between them, and that search efficiency varies continuously across tasks and conditions. The determining factors according to Duncan and Humphreys are the similarities between the target and the non-targets (T-N similarities) and the similarities between the non-targets themselves (N-N similarity). Displays with homogeneous background (high N-N similarity) and a target which is significantly different from the distractors (low T-N similarity) will exhibit parallel, low RT slopes, and vice versa. This claim was illustrated by them using a qualitative "search surface" description as shown in figure 4a. Based on results from our variance analysis, we can now examine this claim quantitatively: We have constructed a "search surface", using actual numerical data of RT slopes from Wolfe's experiments, replacing the N-N similarity axis by its mathematical manifestation, the within-groups standard deviation, and N-T similarity by between-groups standard deviation [1]. The resulting surface (Figure 4b) is qualitatively similar to Duncan and Humphreys's. This interesting result testifies that the PCA representation succeeds in producing a viable realization of such intuitive terms as inputs similarity, and is compatible with the way we perceive the world in visual search tasks.

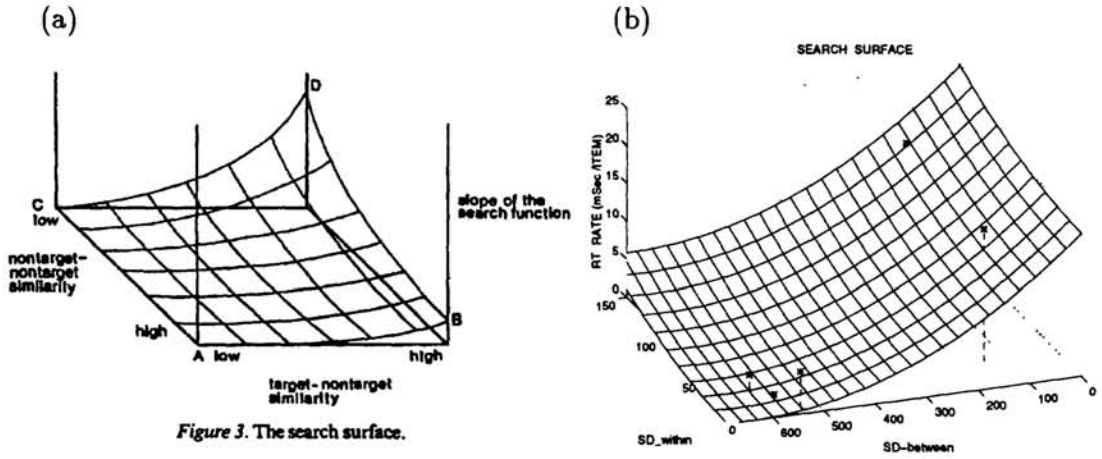

Figure 3. The search surface.

Figure 4: RT rates versus: (a) Input similarities (the search surface, reprinted from Duncan and Humphreys, 1989). (b) Standard deviations (within and between) of the PCA variance analysis. The asterisks denote Wolfe's experimental data.

4 Summary

In this work we present a two-component neural network model of pre-attentional visual processing. The model has been applied to the visual search paradigm performed by Wolfe et. al. Our main finding is that when global-feature compression is applied to visual displays, there is an inherent difference between the representations of serial and parallel-task displays: The neural network studied in this paper has succeeded in detecting a target among distractors only for displays that were experimentally found to be processed in parallel. Based on the outcome of the

[1] In general, each principal axis contains information from different features, which may mask the information concerning the existence of a target. Hence, the first principal axis may not be the best choice for a discrimination task. In our simulations, the 5th axis for example, was primarily dedicated to target information, and was hence used for the variance analysis (obviously, the neural network uses information from all the first eight principal axes).

variance analysis performed on the PCA representations of the visual displays, we present a quantitative criterion enabling one to distinguish between serial and parallel displays. Furthermore, the resulting 'search-surface' generated by the PCA components is in close correspondence with the metaphorical description of Duncan and Humphreys.

The network demonstrates an interesting generalization ability: Naturally, it can learn to detect a target in parallel displays from examples of such displays. However, it can also learn to perform this task from examples of serial displays only! On the other hand, we find that it is impossible to learn serial tasks, irrespective of the combination of parallel and serial displays that are presented to the network during the training phase. This generalization ability is manifested not only during the learning phase, but also during the performance phase; displays belonging to the same task have a similar eigenvalue spectrum, irrespective of the actual set-size of the displays, and this result holds true for parallel as well as for serial displays.

The role of PCA in perception was previously investigated by Cottrell [9], designing a neural network which performed tasks as face identification and gender discrimination. One might argue that PCA, being a global component analysis is not compatible with the existence of local feature detectors (e.g. orientation detectors) in the cortex. Our work is in line with recent proposals [10] that there exist two pathways for sensory input processing: A fast sub-cortical pathway that contains limited information, and a slow cortical pathway which is capable of providing richer representations of the stimuli. Given this assumption this paper has presented the first neural realization of the processing that may underline the classical metaphorical explanations involved in visual search.

References

[1] J. K. Tsotsos. Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13:423–469, 1990.

[2] M. I. Posner, C. R. Snyder, and B. J. Davidson. Attention and the detection of signals. *Journal of Experimental Psychology: General*, 109:160–174, 1980.

[3] Y. Tsal. Movement of attention across the visual field. *Journal of Experimental Psychology: Human Perception and Performance*, 9:523–530, 1983.

[4] A. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[5] J. Duncan and G. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96:433–458, 1989.

[6] A. Treisman and S. Gormican. Feature analysis in early vision: Evidence from search assymetries. *Psychological Review*, 95:15–48, 1988.

[7] J. M. Wolfe, S. R. Friedman-Hill, M. I. Stewart, and K. M. O'Connell. The role of categorization in visual search for orientation. *Journal of Experimental Psychology: Human Perception and Performance*, 18:34–49, 1992.

[8] T. D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Network*, 2:459–473, 1989.

[9] G. W. Cottrell. Extracting features from faces using compression networks: Face, identity, emotion and gender recognition using holons. *Proceedings of the 1990 Connectionist Models Summer School*, pages 328–337, 1990.

[10] J. L. Armony, D. Servan-Schreiber, J. D. Cohen, and J. E. LeDoux. Computational modeling of emotion: exploration through the anatomy and physiology of fear conditioning. *Trends in Cognitive Sciences*, 1(1):28–34, 1997.

Data-Dependent Structural Risk Minimisation for Perceptron Decision Trees

John Shawe-Taylor
Dept of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
Email: jst@dcs.rhbnc.ac.uk

Nello Cristianini
Dept of Engineering Mathematics
University of Bristol
Bristol BS8 1TR, UK
Email: nello.cristianini@bristol.ac.uk

Abstract

Perceptron Decision Trees (also known as Linear Machine DTs, etc.) are analysed in order that data-dependent Structural Risk Minimization can be applied. Data-dependent analysis is performed which indicates that choosing the maximal margin hyperplanes at the decision nodes will improve the generalization. The analysis uses a novel technique to bound the generalization error in terms of the margins at individual nodes. Experiments performed on real data sets confirm the validity of the approach.

1 Introduction

Neural network researchers have traditionally tackled classification problems by assembling perceptron or sigmoid nodes into feedforward neural networks. In this paper we consider a less common approach where the perceptrons are used as decision nodes in a decision tree structure. The approach has the advantage that more efficient heuristic algorithms exist for these structures, while the advantages of inherent parallelism are if anything greater as all the perceptrons can be evaluated in parallel, with the path through the tree determined in a very fast post-processing phase.

Classical Decision Trees (DTs), like the ones produced by popular packages as CART [5] or C4.5 [9], partition the input space by means of axis-parallel hyperplanes (one at each internal node), hence inducing categories which are represented by (axis-parallel) hyperrectangles in such a space.

A natural extension of that hypothesis space is obtained by associating to each internal node hyperplanes in general position, hence partitioning the input space by means of polygonal (polyhedral) categories.

This approach has been pursued by many researchers, often with different motivations, and hence the resulting hypothesis space has been given a number of different names: multivariate DTs [6], oblique DTs [8], or DTs using linear combinations of the attributes [5], Linear Machine DTs, Neural Decision Trees [12], Perceptron Trees [13], etc.

We will call them Perceptron Decision Trees (PDTs), as they can be regarded as binary trees having a simple perceptron associated to each decision node.

Different algorithms for Top-Down induction of PDTs from data have been proposed, based on different principles, [10], [5], [8],

Experimental study of learning by means of PDTs indicates that their performances are sometimes better than those of traditional decision trees in terms of generalization error, and usually much better in terms of tree-size [8], [6], but on some data set PDTs can be outperformed by normal DTs.

We investigate an alternative strategy for improving the generalization of these structures, namely placing maximal margin hyperplanes at the decision nodes. By use of a novel analysis we are able to demonstrate that improved generalization bounds can be obtained for this approach. Experiments confirm that such a method delivers more accurate trees in all tested databases.

2 Generalized Decision Trees

Definition 2.1 *Generalized Decision Trees (GDT).*

Given a space X and a set of boolean functions

$\mathcal{F} = \{f : X \to \{0,1\}\}$, the class GDT($\mathcal{F}$) of Generalized Decision Trees over $\mathcal{F}$ are functions which can be implemented using a binary tree where each internal node is labeled with an element of $\mathcal{F}$, and each leaf is labeled with either 1 or 0.

To evaluate a particular tree T on input $x \in X$, All the boolean functions associated to the nodes are assigned the same argument $x \in X$, which is the argument of $T(x)$. The values assumed by them determine a unique path from the root to a leaf: at each internal node the left (respectively right) edge to a child is taken if the output of the function associated to that internal node is 0 (respectively 1). The value of the function at the assignment of a $x \in X$ is the value associated to the leaf reached. We say that input x reaches a node of the tree, if that node is on the evaluation path for x.

In the following, the *nodes* are the internal nodes of the binary tree, and the *leaves* are its external ones.

Examples.

- Given $X = \{0,1\}^n$, a *Boolean Decision Tree (BDT)* is a GDT over
$$\mathcal{F}_{\mathrm{BDT}} = \{f_i : f_i(\mathbf{x}) = \mathbf{x}_i, \forall \mathbf{x} \in X\}$$
- Given $X = \mathbb{R}^n$, a *C4.5-like Decision Tree (CDT)* is a GDT over
$$\mathcal{F}_{\mathrm{CDT}} = \{f_{i,\theta} : f_{i,\theta}(\mathbf{x}) = 1 \Leftrightarrow x_i > \theta\}$$
This kind of decision trees defined on a continuous space are the output of common algorithms like C4.5 and CART, and we will call them - for short - CDTs.
- Given $X = \mathbb{R}^n$, a *Perceptron Decision Tree (PDT)* is a GDT over
$$\mathcal{F}_{\mathrm{PDT}} = \{w^T\mathbf{x} : w \in \mathbb{R}^{n+1}\},$$
where we have assumed that the inputs have been augmented with a coordinate of constant value, hence implementing a thresholded perceptron.

3 Data-dependent SRM

We begin with the definition of the fat-shattering dimension, which was first introduced in [7], and has been used for several problems in learning since [1, 4, 2, 3].

Definition 3.1 *Let $\mathcal{F}$ be a set of real valued functions. We say that a set of points X is γ-shattered by $\mathcal{F}$ relative to $r = (r_x)_{x \in X}$ if there are real numbers r_x indexed by $x \in X$ such that for all binary vectors b indexed by X, there is a function $f_b \in \mathcal{F}$ satisfying*

$$f_b(x) \begin{cases} \geq r_x + \gamma & \text{if } b_x = 1 \\ \leq r_x - \gamma & \text{otherwise.} \end{cases}$$

The fat shattering dimension $\text{fat}_{\mathcal{F}}$ *of the set $\mathcal{F}$ is a function from the positive real numbers to the integers which maps a value γ to the size of the largest γ-shattered set, if this is finite, or infinity otherwise.*

As an example which will be relevant to the subsequent analysis consider the class:

$$\mathcal{F}_{\text{lin}} = \{x \to \langle w, x \rangle + \theta : \|w\| = 1\}.$$

We quote the following result from [11].

Corollary 3.2 *[11] Let $\mathcal{F}_{\text{lin}}$ be restricted to points in a ball of n dimensions of radius R about the origin and with thresholds $|\theta| \leq R$. Then*

$$\text{fat}_{\mathcal{F}_{\text{lin}}}(\gamma) \leq \min\{9R^2/\gamma^2, n+1\} + 1.$$

The following theorem bounds the generalization of a classifier in terms of the fat shattering dimension rather than the usual Vapnik-Chervonenkis or Pseudo dimension.

Let T_θ denote the threshold function at θ: $T_\theta : \mathbb{R} \to \{0, 1\}$, $T_\theta(\alpha) = 1$ iff $\alpha > \theta$. For a class of functions $\mathcal{F}$, $T_\theta(\mathcal{F}) = \{T_\theta(f) : f \in \mathcal{F}\}$.

Theorem 3.3 *[11] Consider a real valued function class $\mathcal{F}$ having fat shattering function bounded above by the function* afat $: \mathbb{R} \to \mathbb{N}$ *which is continuous from the right. Fix $\theta \in \mathbb{R}$. If a learner correctly classifies m independently generated examples $\mathbf{z}$ with $h = T_\theta(f) \in T_\theta(\mathcal{F})$ such that $\text{er}_{\mathbf{z}}(h) = 0$ and $\gamma = \min |f(x_i) - \theta|$, then with confidence $1 - \delta$ the expected error of h is bounded from above by*

$$\epsilon(m, k, \delta) = \frac{2}{m}\left(k \log\left(\frac{8em}{k}\right)\log(32m) + \log\left(\frac{8m}{\delta}\right)\right),$$

where $k = \text{afat}(\gamma/8)$.

The importance of this theorem is that it can be used to explain how a classifier can give better generalization than would be predicted by a classical analysis of its VC dimension. Essentially expanding the margin performs an automatic capacity control for function classes with small fat shattering dimensions. The theorem shows that when a large margin is achieved it is as if we were working in a lower VC class.

We should stress that in general the bounds obtained should be better for cases where a large margin is observed, but that a priori there is no guarantee that such a margin will occur. Therefore a priori only the classical VC bound can be used. In view of corresponding lower bounds on the generalization error in terms of the VC dimension, the a posteriori bounds depend on a favourable probability distribution making the actual learning task easier. Hence, the result will only be useful if the distribution is favourable or at least not adversarial. In this sense the result is a distribution dependent result, despite not being distribution dependent in the

traditional sense that assumptions about the distribution have had to be made in its derivation. The benign behaviour of the distribution is automatically estimated in the learning process.

In order to perform a similar analysis for perceptron decision trees we will consider the set of margins obtained at each of the nodes, bounding the generalization as a function of these values.

4 Generalisation analysis of the Tree Class

It turns out that bounding the fat shattering dimension of PDT's viewed as real function classifiers is difficult. We will therefore do a direct generalization analysis mimicking the proof of Theorem 3.3 but taking into account the margins at each of the decision nodes in the tree.

Definition 4.1 *Let (X, d) be a (pseudo-) metric space, let A be a subset of X and $\epsilon > 0$. A set $B \subseteq X$ is an* ϵ-cover *for A if, for every $a \in A$, there exists $b \in B$ such that $d(a, b) < \epsilon$. The* ϵ-covering number *of A, $\mathcal{N}_d(\epsilon, A)$, is the minimal cardinality of an ϵ-cover for A (if there is no such finite cover then it is defined to be ∞).*

We write $\mathcal{N}(\epsilon, \mathcal{F}, \mathbf{x})$ for the ϵ-covering number of $\mathcal{F}$ with respect to the ℓ_∞ pseudo-metric measuring the maximum discrepancy on the sample $\mathbf{x}$. These numbers are bounded in the following Lemma.

Lemma 4.2 (Alon *et al.* **[1])** *Let $\mathcal{F}$ be a class of functions $X \to [0, 1]$ and P a distribution over X. Choose $0 < \epsilon < 1$ and let $d = \mathrm{fat}_{\mathcal{F}}(\epsilon/4)$. Then*

$$E\left(\mathcal{N}(\epsilon, \mathcal{F}, \mathbf{x})\right) \leq 2\left(\frac{4m}{\epsilon^2}\right)^{d\log(2em/(d\epsilon))},$$

where the expectation E is taken w.r.t. a sample $\mathbf{x} \in X^m$ drawn according to P^m.

Corollary 4.3 *[11] Let $\mathcal{F}$ be a class of functions $X \to [a, b]$ and P a distribution over X. Choose $0 < \epsilon < 1$ and let $d = \mathrm{fat}_{\mathcal{F}}(\epsilon/4)$. Then*

$$E\left(\mathcal{N}(\epsilon, \mathcal{F}, \mathbf{x})\right) \leq 2\left(\frac{4m(b-a)^2}{\epsilon^2}\right)^{d\log(2em(b-a)/(d\epsilon))},$$

where the expectation E is over samples $\mathbf{x} \in X^m$ drawn according to P^m.

We are now in a position to tackle the main lemma which bounds the probability over a double sample that the first half has zero error and the second error greater than an appropriate ϵ. Here, error is interpreted as being differently classified at the output of tree. In order to simplify the notation in the following lemma we assume that the decision tree has K nodes. We also denote $\mathrm{fat}_{\mathcal{F}_{\mathrm{lin}}}(\gamma)$ by $\mathrm{fat}(\gamma)$ to simplify the notation.

Lemma 4.4 *Let T be a perceptron decision tree with K decision nodes with margins $\gamma^1, \gamma^2, \ldots, \gamma^K$ at the decision nodes. If it has correctly classified m labelled examples generated independently according to the unknown (but fixed) distribution P, then we can bound the following probability to be less than δ,*

$$P^{2m}\left\{\mathbf{xy}: \exists \text{ a tree } T : T \text{ correctly classifies } \mathbf{x}, \right.$$
$$\left. \text{fraction of } \mathbf{y} \text{ misclassified } > \epsilon(m, K, \delta)\right\} < \delta,$$

where $\epsilon(m, K, \delta) = \frac{1}{m}(D\log(4m) + \log\frac{2^K}{\delta})$.

where $D = \sum_{i=1}^{K} k_i \log(4em/k_i)$ and $k_i = \mathrm{fat}(\gamma_i/8)$.

Proof: Using the standard permutation argument, we may fix a sequence **xy** and bound the probability under the uniform distribution on swapping permutations that the sequence satisfies the condition stated. We consider generating minimal $\gamma_k/2$-covers $B^k_{\mathbf{xy}}$ for each value of k, where $\gamma_k = \min\{\gamma' : \text{fat}(\gamma'/8) \leq k\}$. Suppose that for node i of the tree the margin γ^i of the hyperplane w_i satisfies $\text{fat}(\gamma^i/8) = k_i$. We can therefore find $f_i \in B^{k_i}_{\mathbf{xy}}$ whose output values are within $\gamma^i/2$ of w_i. We now consider the tree T' obtained by replacing the node perceptrons w_i of T with the corresponding f_i. This tree performs the same classification function on the first half of the sample, and the margin remains larger than $\gamma^i - \gamma_{k_i}/2 > \gamma_{k_i}/2$. If a point in the second half of the sample is incorrectly classified by T it will either still be incorrectly classified by the adapted tree T' or will at one of the decision nodes i in T' be closer to the decision boundary than $\gamma_{k_i}/2$. The point is thus distinguishable from left hand side points which are both correctly classified and have margin greater than $\gamma_{k_i}/2$ at node i. Hence, that point must be kept on the right hand side in order for the condition to be satisfied. Hence, the fraction of permutations that can be allowed for one choice of the functions from the covers is $2^{-\epsilon m}$. We must take the union bound over all choices of the functions from the covers. Using the techniques of [11] the numbers of these choices is bounded by Corollory 4.3 as follows

$$\Pi_{i=1}^{K} 2(8m)^{k_i \log(4em/k_i)} = 2^K (8m)^D,$$

where $D = \sum_{i=1}^{K} k_i \log(4em/k_i)$. The value of ϵ in the lemma statement therefore ensures that this the union bound is less than δ.

□

Using the standard lemma due to Vapnik [14, page 168] to bound the error probabilities in terms of the discrepancy on a double sample, combined with Lemma 4.4 gives the following result.

Theorem 4.5 *Suppose we are able to classify an m sample of labelled examples using a perceptron decision tree with K nodes and obtaining margins γ_i at node i, then we can bound the generalisation error with probability greater than $1-\delta$ to be less than*

$$\frac{1}{m}(D \log(4m) + \log \frac{(8m)^K \binom{2K}{K}}{(K+1)\delta})$$

where $D = \sum_{i=1}^{K} k_i \log(4em/k_i)$ and $k_i = \text{fat}(\gamma_i/8)$.

Proof: We must bound the probabilities over different architectures of trees and different margins. We simply have to choose the values of ϵ to ensure that the individual δ's are sufficiently small that the total over all possible choices is less than δ. The details are omitted in this abstract.

□

5 Experiments

The theoretical results obtained in the previous section imply that an algorithm which produces large margin splits should have a better generalization, since increasing the margins in the internal nodes, has the effect of decreasing the bound on the test error.

In order to test this strategy, we have performed the following experiment, divided in two parts: first run a standard perceptron decision tree algorithm and then for each decision node generate a maximal margin hyperplane implementing the same dichotomy in place of the decision boundary generated by the algorithm.

Input: Random m sample $\mathbf{x}$ with corresponding classification b.

Algorithm: Find a perceptron decision tree T which correctly classifies the sample using a standard algorithm;
Let k = number of decision nodes of T;
From tree T create T' by executing the following loop:

For each decision node i replace the weight vector w_i by the vector w_i' which realises the maximal margin hyperplane agreeing with w_i on the set of inputs reaching node i;

Let the margin of w_i' on the inputs reaching node i be γ_i;

Output: Classifier T', with bound on the generalisation error in terms of the number of decision nodes K and $D = \sum_{i=1}^{K} k_i \log(4em/k_i)$ where $k_i = \mathrm{fat}(\gamma_i/8)$.

Note that the classification of T and T' agree on the sample and hence, that T' is consistent with the sample.

As a PDT learning algorithm we have used OC1 [8], created by Murthy, Kasif and Salzberg and freely available over the internet. It is a randomized algorithm, which performs simulated annealing for learning the perceptrons. The details about the randomization, the pruning, and the splitting criteria can be found in [8].

The data we have used for the test are 4 of the 5 sets used in the original OC1 paper, which are publicly available in the UCI data repository [16].

The results we have obtained on these data are compatible with the ones reported in the original OC1 paper, the differences being due to different divisions between training and testing sets and their sizes; the absence in our experiments of cross-validation and other techniques to estimate the predictive accuracy of the PDT; and the inherently randomized nature of the algorithm.

The second stage of the experiment involved finding - for each node - the hyperplane which performes *the same* split as performed by the OC1 tree but with the maximal margin. This can be done by considering the subsample reaching each node as perfectly divided in two parts, and feeding the data accordingly relabelled to an algorithm which finds the optimal split in the linearly separable case. The maximal margin hyperplanes are then placed in the decision nodes and the new tree is tested on the same testing set.

The data sets we have used are: *Wiscounsin Breast Cancer*, *Pima Indians Diabetes*, *Boston Housing* transformed into a classification problem by thresholding the price at \$ 21.000 and the classical *Iris* studied by Fisher (More informations about the databases and their authors are in [8]). All the details about sample sizes, number of attributes and results (training and testing accuracy, tree size) are summarized in table 1.

We were not particularly interested in achieving a high testing accuracy, but rather in observing if improved performances can be obtained by increasing the margin. For this reason we did not try to optimize the performance of the original classifier by using cross-validation, or a convenient training/testing set ratio. The relevant quantity, in this experiment, is the different in the testing error between a PDT with arbitrary margins and the same tree with optimized margins. This quantity has turned out to be always positive, and to range from 1.7 to 2.8 percent of gain, on test errors which were already very low.

	train	OC1 test	FAT test	#trs	#ts	attrib.	classes	nodes
CANC	96.53	93.52	95.37	249	108	9	2	1
IRIS	96.67	96.67	98.33	90	60	4	3	2
DIAB	89.00	70.48	72.45	209	559	8	2	4
HOUS	95.90	81.43	84.29	306	140	13	2	7

References

[1] Ncga Alon, Shai Ben-David, Nicolò Cesa-Bianchi and David Haussler, "Scale-sensitive Dimensions, Uniform Convergence, and Learnability," in *Proceedings of the Conference on Foundations of Computer Science (FOCS)*, (1993). Also to appear in *Journal of the ACM.*

[2] Martin Anthony and Peter Bartlett, "Function learning from interpolation", Technical Report, (1994). (An extended abstract appeared in *Computational Learning Theory, Proceedings 2nd European Conference, EuroCOLT'95*, pages 211–221, ed. Paul Vitanyi, (Lecture Notes in Artificial Intelligence, 904) Springer-Verlag, Berlin, 1995).

[3] Peter L. Bartlett and Philip M. Long, "Prediction, Learning, Uniform Convergence, and Scale-Sensitive Dimensions," Preprint, Department of Systems Engineering, Australian National University, November 1995.

[4] Peter L. Bartlett, Philip M. Long, and Robert C. Williamson, "Fat-shattering and the learnability of Real-valued Functions," *Journal of Computer and System Sciences,* **52**(3), 434-452, (1996).

[5] Breiman L., Friedman J.H., Olshen R.A., Stone C.J., "Classification and Regression Trees", Wadsworth International Group, Belmont, CA, 1984.

[6] Brodley C.E., Utgoff P.E., Multivariate Decision Trees, Machine Learning 19, pp. 45-77, 1995.

[7] Michael J. Kearns and Robert E. Schapire, "Efficient Distribution-free Learning of Probabilistic Concepts," pages 382–391 in *Proceedings of the 31st Symposium on the Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[8] Murthy S.K., Kasif S., Salzberg S., A System for Induction of Oblique Decision Trees, Journal of Artificial Intelligence Research, 2 (1994), pp. 1-32.

[9] Quinlan J.R., "C4.5: Programs for Machine Learning", Morgan Kaufmann, 1993.

[10] Sankar A., Mammone R.J., Growing and Pruning Neural Tree Networks, IEEE Transactions on Computers, 42:291-299, 1993.

[11] John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, Martin Anthony, Structural Risk Minimization over Data-Dependent Hierarchies, NeuroCOLT Technical Report NC-TR-96-053, 1996.
(`ftp://ftp.dcs.rhbnc.ac.uk/pub/neurocolt/tech_reports`).

[12] J.A. Sirat, and J.-P. Nadal, "Neural trees: a new tool for classification", Network, 1, pp. 423-438, 1990

[13] Utgoff P.E., Perceptron Trees: a Case Study in Hybrid Concept Representations, Connection Science 1 (1989), pp. 377-391.

[14] Vladimir N. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, New York, 1982.

[15] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995

[16] University of California, Irvine - Machine Learning Repository, `http://www.ics.uci.edu/ mlearn/MLRepository.html`