An Analysis of the Anti-Learning Phenomenon for the Class Symmetric Polyhedron

Adam Kowalczyk[1] and Olivier Chapelle[2]

[1] National ICT Australia and RSISE, The Australian National University
Canberra, Australia
adam.kowalczyk@nicta.com.au
[2] Max Planck Institute for Biological Cybernetics,
Tübingen, Germany
olivier.chapelle@tuebingen.mpg.de

Abstract. This paper deals with an unusual phenomenon where most machine learning algorithms yield good performance on the training set but systematically *worse than random performance* on the test set. This has been observed so far for some natural data sets and demonstrated for some synthetic data sets when the classification rule is learned from a small set of training samples drawn from some high dimensional space. The initial analysis presented in this paper shows that anti-learning is a property of data sets and is quite distinct from over-fitting of a training data. Moreover, the analysis leads to a specification of some machine learning procedures which can overcome anti-learning and generate machines able to classify training and test data consistently.

1 Introduction

The goal of a supervised learning system for binary classification is to classify instances of an independent test set as well as possible on the basis of a model learned from a labeled training set. Typically, the model has similar classification behavior on both the training and test sets, i.e., it classifies training and test instances with precision higher than the expected accuracy of the random classifier. Thus it has what we refer to as "*the learning mode*". However, there are real life situations where better than random performance on the training set yields systematically worse than random performance on the off-training test set. One example is the Aryl Hydrocarbon Receptor classification task in KDD Cup 2002 [3, 9, 11]. These systems exhibit what we call "*the anti-learning mode*". As it has been discussed in [8], anti-learning can be observed in publicly available microarray data used for prediction of cancer outcomes, which can show both learning and anti-learning mode, depending on the features selected.

In this paper however, we focus on synthetic data which facilitates rigorous analysis. The aim is to demonstrate rigorously that anti-learning can occur, and can be primarily a feature of the data as it happens for many families of algorithms, universally across all setting of tunable parameters. In particular, we analyse a task of classification of binary labeled vertices of a class symmetric

polyhedron embedded in a sphere (Section 2). The classification task seems to be very easy: the data is linearly separable and any two labeled vertices determine all labels. However, this simplicity is very deceptive: we prove in Section 3 that none of the wide range of well establish algorithms such as perceptron, Support Vector Machines, generalised regression, κ-nearest neighbours can learn to classify consistently the data. In fact, given a proper subset of the domain to train, they can easily learn to classify it, but they always totally misclassify the remaining data. This effect is very different from poor generalization abilities where a classifier would perform close to random: here the predictions on the test set are not random, they are exactly the *opposite* of what they should be. In Section 3.2 we show that there exists kernel transformations which can actually change perfect anti-learning data into perfectly learnable data. Finally, Section 4 discuses the results.

2 Geometry of Class Symmetric Kernels

Let S be a set of labeled examples $\{(\mathbf{x}_i, y_i)\}_{i\in\mathbb{S}} \subset X \times \{-1, +1\}$ indexed uniquely by the index set $\mathbb{S}$, with $X \subset \mathbb{R}^N$. We are interested in classification rules of the form $\text{sign} \circ f : X \to \{-1, +1\}$, where $f = \mathcal{A}(T)$. $\mathcal{A}$ may also depend on some hyperparameters such as kernel $k : X \times X \to \mathbb{R}$, the regularization constant, etc.

2.1 Performance Measures

Assume we are given $f : X \to \mathbb{R}$ and a non-void test subset

$$T = \{(\mathbf{x}_i, y_i)\}_{i\in\mathbb{T}} \subset S$$

indexed uniquely by $\mathbb{T} \subset \mathbb{S}$ and containing samples from both labels.

Accuracy We define the *accuracy* of the decision rule $\mathbf{x} \mapsto \text{sign}(f(\mathbf{x}))$ as

$$\text{ACC}(f, T) = \frac{1}{2} \sum_{y=\pm 1} \mathbb{P}_T\big(y_j f(\mathbf{x}_j) > 0 \mid y_j = y\big)$$

Here $\mathbb{P}_T$ denotes the frequency calculated for the subset $T \subset S$. Note this is the *balanced* performance measure, independent of prior distribution of data classes.

Area Under ROC For f as above, we use the *Area under the Receiver Operating Characteristic* curve, $AROC(f, T)$[3], the plot of true vs. false positive rate, as another performance measure. Following [1] we use the formula

$$\begin{aligned}\text{AROC}(f, T) = \mathbb{P}_T\big(f(\mathbf{x}_i) < f(\mathbf{x}_j) \mid y_i = -1,\ y_j = 1\big) \\ + \frac{1}{2}\,\mathbb{P}_T\big(f(\mathbf{x}_i) = f(\mathbf{x}_j) \mid y_i \neq y_j\big)\end{aligned}$$

[3] Also known as *the area under the curve*, *AUC*; it is essentially the well known order statistics U.

Note that the second term in the above formula takes care of ties, when instances from different labels are mapped to the same value.

The expected value of both $\textsc{aroc}(f,T)$ and $\textsc{acc}(f,T)$ for the trivial, uniformly random predictor f, is 0.5. This is also the value for these metrics for the trivial constant classifier mapping all T to a constant value, ± 1. Note the following fact:

$$\textsc{aroc}(f,T)=1 \quad \Leftrightarrow \quad \exists C,\ \forall i \in \mathbb{T},\ \ y_i(f(\mathbf{x}_i)-C)>0; \tag{1}$$

$$\textsc{aroc}(f,T)=0 \quad \Leftrightarrow \quad \exists C,\ \forall i \in \mathbb{T},\ \ y_i(f(\mathbf{x}_i)-C)<0. \tag{2}$$

Remark 1. There are at least two reasons why we use AROC in this paper.

1. AROC is a widely used measure of classifier performance in practical applications, especially biological and biomedical classification. As we have indicated in the introduction, some biomedical classification problems are the ultimate target this paper is a step towards, so explicit usage of AROC makes a direct link to such applications.
2. AROC is independent of an additive bias term while accuracy or error rate are critically dependent on a selection of such a term. For instance, $\textsc{acc}(f+b,T)=0.5$ for any $b \geq \max(f(T))$, even if $\textsc{acc}(f,T)=0$. Typically, in such a case other intermediate values for $\textsc{acc}(f+b',T)$ could be also obtained for other values of the bias b'. However, $\textsc{aroc}(f+b,T)=\text{const}$, since AROC depends on the order in the set $(f+b)(T) \subset \mathbb{R}$ and this is independent of the additive constant b. (Note that modulo a constant factor, AROC is a well known order statistic U [1].) □

2.2 Class Symmetric Matrices

Now we introduce the basic object for theoretical analysis in this paper. In order to simplify deliberations we consider synthetic datasets for which the entries in the Gram matrix depend only on the classes of the corresponding points.

Definition 1. *A matrix $[k_{ij}]_{i,j\in\mathbb{S}}$ is called* class symmetric *if there exists constants $r>0$ and $c_y \in \mathbb{R}$ for $y \in \{0,\pm 1\}$ such that for $i,j \in \mathbb{S}$,*

$$k_{ij} := k(\mathbf{x}_i,\mathbf{x}_j) = \begin{cases} r^2, & i=j \\ r^2 c_{y_i}, & y_i=y_j, i\neq j \\ r^2 c_0, & y_i \neq y_j \end{cases} \tag{3}$$

We will also say that the kernel k is class symmetric *on S.*

Now we establish a necessary and sufficient condition on the coefficients of this matrix for it to be a positive definite kernel matrix.

Lemma 1. *The following conditions are equivalent:*
(i) For $D_y := \frac{1-c_y}{n_y} + c_y$, where $n_y := |i \in \mathbb{S}\ ;\ y_i = y|$ for $y=\pm 1$ we have

$$D_+ D_- > c_0^2,\ D_y > 0 \quad \text{and} \quad 1-c_y>0 \ \text{ for } y=\pm 1; \tag{4}$$

(*ii*) *The matrix* $[k_{ij}]_{i,j\in\mathbb{S}}$ *defined by* (3) *is positive definite;*

(*iii*) *There exist linearly independent vectors* $\mathbf{z}_i \in \mathbb{R}^{|S|}$, *such that* $k_{ij} = \mathbf{z}_i \cdot \mathbf{z}_j$ *for any* $i, j \in \mathbb{S}$.

See Appendix for the proof.

The points $\mathbf{z}_i$ as above belong to the sphere of radius r and the center at $0 \in \mathbb{R}^{|S|}$. They are vertices of *a class symmetric polyhedron, (CS-polyhedron)*, of $n = |S|$ vertices and $n(n+1)/2$ edges. The vertices of the same label y form an n_y-simplex, with all edges of constant length $d_y := r\sqrt{2-2c_y}$, $y = \pm 1$. The distances between any pair of vertices of opposite labels are equal to $d_0 := r\sqrt{2-2c_0}$. Note that the linear independence in Lemma 1 insures that the different labels on CS-polyhedron are *linearly separable*.

It is interesting to have a geometrical picture of what happens in the case $c_y < c_0,\ y \pm 1$. In that case, each point is nearer to all the points of the opposite class than to any point of the same class. In this kind of situation, the nearest κ-neighbors classifier would obviously lead to anti-learning. Thus we have:

Proposition 1. *Let* $\kappa > 1$ *be an odd integer,* k *be an CS-kernel on* S, *and* T *contains* $> \kappa/2$ *points from each label. If* $c_y < c_0$ *for* $y \pm 1$, *then the* κ*-nearest neighbours algorithm* f_κ *based on the distance in the feature space,* $\rho(\mathbf{x}, \mathbf{x}') := \sqrt{2r^2 - 2k(\mathbf{x}, \mathbf{x}')}$, *will allocate opposite labels to every point in* S, *i.e.* $\text{ACC}(f_\kappa, S) = \text{AROC}(f_\kappa, S) = 0$.

An example of four point perfect subset $S \subset \mathbb{R}^3$ satisfying assumptions of this Proposition is given in Figure 1.

The geometry of CS-polyhedron can be hidden in the data. An example which will be used in our simulations follows.

Example 1 (Hadamard matrix). Hadamard matrices are special (square) orthogonal matrices of 1's and -1's. They have applications in combinatorics, signal processing, numerical analysis. An $n \times n$ Hadamard matrix, H_n, with $n > 2$ exists only if n is divisible by 4. The Matlab function `hadamard(n)` handles only cases where n, $n/12$ or $n/20$ is a power of 2. Hadamard matrices give rise to CS-polyhedrons $S^o(H_n) = \left\{(\mathbf{x}_i, y_i)\right\}_{i=1,\ldots,n} \subset \mathbb{R}^{n-1}$. The recipe is as follows. Choose a non-constant row and use its entries as labels y_i. For data points, $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^{n-1}$, use the columns of the remaining $(n-1) \times n$ matrix. An example of 4×4-Hadamard matrix, the corresponding data for the 3-rd row used as labels and the kernel matrix follows :

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix};\ y = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix};\ [\mathbf{x}_1, ..., \mathbf{x}_4] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix};$$

$$[k_{ij}] = \begin{bmatrix} 3 & -1 & 1 & 1 \\ -1 & 3 & 1 & 1 \\ 1 & 1 & 3 & -1 \\ 1 & 1 & -1 & 3 \end{bmatrix}.$$

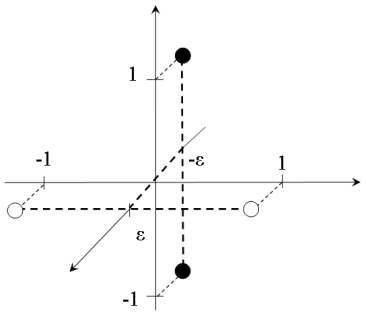

Fig. 1. Elevated XOR - an example of the perfect anti-learning data in 3-dimensions. The z-values are $\pm\epsilon$. The linear kernel satisfies the CS-condition (3) with $r^2 = 1 + \epsilon^2$, $c_0 = -\epsilon^2 r^{-2}$ and $c_{-1} = c_{+1} = (-1 + \epsilon^2)r^{-2}$. Hence the perfect anti-learning condition (6) holds if $\epsilon < 0.5$. It can be checked directly, that any linear classifier such as perceptron or maximal margin classifier, trained on a proper subset misclassify all the off-training points of the domain. This can be especially easily visualized for $0 < \epsilon \ll 1$.

Since the columns of Hadamard matrix are orthogonal, from the above construction we obtain $\mathbf{x}_i \cdot \mathbf{x}_j + y_i y_j = n\delta_{ij}$. Hence the dot-product kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ satisfies (3) with $c_0 = -c_y = 1/(n-1)$ and $r^2 = n - 1$.

Note that the vectors of this data set are linearly dependent, hence this set does not satisfy Lemma 1 (condition (iii) does not hold). It is instructive to check directly that the fist inequality in (ii) is violates as well. Indeed, in such case we have $D_y = \frac{1}{n-1}$, hence the equality $D_+ D_- = c_0^2$ holds.

In order to comply strictly with Lemma 1, one of the vectors form $S^o(H_n)$ has to be removed. After such removal we will have a set of $n-1$ vectors in the $n-1$ dimensional space which are linearly independent. This can be easily checked directly, but also we check equivalent condition (ii) of Lemma 1. Indeed, in such a case we obtain $D_+ = \frac{n+2}{(n-1)(n-2)}$ and $D_- = \frac{1}{n-1}$ assuming that the first, constant vector (with the positive label) has been removed. Hence,

$$D_+ D_- = \frac{n+2}{(n-1)^2(n+2)} > \frac{1}{(n-1)^2} = c_0^2$$

and all inequalities in (4) hold.

We shall denote this truncated Hadamard set by $S(H_n)$.

3 Perfect Learning/Anti-Learning Theorem

Standing Assumption: In order to simplify our considerations, from now on we assume that $\emptyset \neq T \subset S$ is a subset such that both T and $S\backslash T$ contain examples from *both labels*.

Now we consider the class of kernel machines [4, 12, 13]. We say that the function $f : X \to \mathbb{R}$ has *a convex cone expansion on* T or write $f \in \text{CONE}(k, T)$, if there exists coefficients $\alpha_i \geq 0$, $i \in \mathbb{T}$, such that $\alpha \neq 0$ and

$$f(\mathbf{x}) = \sum_{i \in \mathbb{T}} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) \quad \text{for every } \mathbf{x} \in X. \tag{5}$$

As for the κ-nearest neighbours algorithm, see Proposition 2 of Section 2.2, the learning mode for kernel machines with CS-kernel depends on the relative values of c_y and c_0. More precisely, the following theorem holds.

Theorem 1. *If k is a positive definite CS-kernel on S then the following three conditions (*the perfect anti-learning*) are equivalent:*

$$c_y < c_0 \qquad \textit{for } y = \pm 1, \tag{6}$$

$$\forall T \subset S,\ \forall f \in \text{CONE}(k, T), \quad \text{AROC}(f, S\backslash T) = 0, \tag{7}$$

$$\forall T \subset S,\ \forall f \in \text{CONE}(k, T),\ \exists b \in \mathbb{R}, \quad \text{ACC}(f + b, S\backslash T) = 0, \tag{8}$$

*Likewise, the following three conditions (*the perfect learning*) are equivalent*

$$c_y > c_0 \qquad \textit{for } y = \pm 1, \tag{9}$$

$$\forall T \subset S,\ \forall f \in \text{CONE}(k, T),\ \text{AROC}(f, S\backslash T) = 1, \tag{10}$$

$$\forall T \subset S,\ \forall f \in \text{CONE}(k, T),\ \exists b \in \mathbb{R}, \quad \text{ACC}(f + b, S\backslash T) = 1, \tag{11}$$

Proof. For f as in (5), (3) holding for the CS-kernel k and $b := \sum_{i \in \mathbb{T}} \alpha_i y_i c_0$, we have

$$y_j\big(f(\mathbf{x}_i) - b\big) = y_j \sum_{i \in \mathbb{T}} \alpha_i y_i (k_{ij} - c_0) = \sum_{i \in \mathbb{T}, y_i = y_j} \alpha_i (c_{y_j} - c_0) - \sum_{i \in \mathbb{T}, y_i \neq y_j} \alpha_i (c_0 - c_0)$$

$$= \begin{cases} < 0, \text{ if (6) holds;} \\ > 0, \text{ if (9) holds;} \end{cases}$$

for $j \in \mathbb{S} - \mathbb{T}$. This proves immediately the equivalences (6) $\Leftrightarrow$ (8) and (9) $\Leftrightarrow$ (11), respectively. Application of (1) and (2) completes the proof. □

Example 2. We discuss the Theorem 1 on example of elevated XOR, see Figure 1. In this case our standing assumption requires that both T and $S\backslash T$ contain examples from both labels, so each must contain two points. Assume $\mathbb{T} = \{1, 2\}$, $S\backslash\mathbb{T} = \{3, 4\}$ and $y_2 = y_4 = +1$. For the linear kernel k_{lin}, the classifier $f \in \text{CONE}(k_{lin}, T)$ has the form $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, where $\mathbf{x}, \mathbf{w} = \alpha_1 \mathbf{x}_1 - \alpha_2 \mathbf{x}_2 \in \mathbb{R}^3$, where $\alpha_1, \alpha_2 \geq 0$. This classifier orders training points consistently with the labels, i.e. satisfies the condition $\text{AROC}(f, T) > 0.5$ or equivalently $f(\mathbf{x}_2) > f(\mathbf{x}_1)$

iff the angle between vectors $\mathbf{w}$ and $\mathbf{x}_2-\mathbf{x}_1$ is $> \frac{\pi}{2}$. Similarly, $\text{AROC}(f, S\backslash T) > 0.5$ iff the angle between vectors $\mathbf{w}$ and $\mathbf{x}_4 - \mathbf{x}_3$ is $> \frac{\pi}{2}$.

It straightforward to check that for $\mathbf{w}$ as above,

$$\cos(\mathbf{x}_2 - \mathbf{x}_1, \mathbf{w}) \leq 2^{-0.5}\sqrt{1 + \frac{\epsilon^2}{1+\epsilon^2}},$$

hence $\cos(\mathbf{x}_2 - \mathbf{x}_1, \mathbf{w}) < \frac{\pi}{4} + \epsilon^2$. Now we can check that

$$\cos(\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_4 - \mathbf{x}_3) = -1 + \frac{4\epsilon^2}{1+2\epsilon^2}$$

hence $angle(\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_4 - \mathbf{x}_3) > \pi - 4\epsilon^2$. This implies that $angle(\mathbf{x}_4 - \mathbf{x}_3, \mathbf{w}) > \frac{3}{4}\pi - 5\epsilon^2 > \frac{\pi}{4}$, if $\epsilon < \sqrt{\frac{3}{20}\pi}$. Thus the test set samples are miss-ordered and $\text{AROC}(f, S\backslash T) = 0$. □

A number of popular supervised learning algorithms outputs solutions $f(\mathbf{x}) + b$ where $f \in \text{CONE}(k, T)$ and $b \in \mathbb{R}$. This includes the support vector machines (both with linear and non-linear kernels) [2, 4, 12], the classical or the kernel or the voting perceptron [5]. The class $\text{CONE}(k, T)$ is convex, hence boosting of weak learners from $\text{CONE}(k, T)$ produces also a classifier in this class. Others algorithms, such as regression or the generalized (ridge) regression [4, 12] applied to the CS-kernel k on T, necessarily output such a machine. Indeed the following proposition holds

Proposition 2. *Let k be a positive definite CS-kernel on S. The kernel ridge regression algorithm minimizes in feature space,*

$$R = \lambda\|\mathbf{w}\|^2 + \sum_{i\in T} \xi_i^2, \quad \textit{with } \xi_i := 1 - y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) + b),$$

and the optimal solution $\mathbf{x} \mapsto \mathbf{w}\cdot\Phi(\mathbf{x}) \in \text{CONE}(k, T)$

Proof. It is sufficient to consider the linear case, i.e. $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b$. Due to the linear independence and class symmetry in vectors $\mathbf{x}_i$, the vector $\mathbf{w}$ has the unique expansion

$$\lambda\mathbf{w} = \sum_i y_i\xi_i\mathbf{x}_i = n_+\xi_+ \sum_{i,y_i=+1} \mathbf{x}_i - n_-\xi_- \sum_{i,y_i=-1} \mathbf{x}_i.$$

Both slacks ξ_+ and ξ_- are ≥ 0 at the minimum. Indeed, if one of them is < 0, we can "shrink" $\|\mathbf{w}\|$, i.e. the replacement $\mathbf{w} \leftarrow a\mathbf{w}$, where $0 < a < 1$, and adjust b, in a way which will decrease the magnitude of this slack leaving the other one unchanged. So R would decrease. This contradicts that $(\mathbf{w}, b)$ minimizes R. □

3.1 Transforming the kernel matrix

Consider the case where the linear kernel $k_{lin}(\mathbf{x}_i, \mathbf{x}_j) := \mathbf{x}_i \cdot \mathbf{x}_j$ applied to some data (such as the Hadamard matrix) yields a CS-kernel for which anti-learning occurs (i.e. $c_y < c_0,\ y \pm 1$ from the Theorem 1). A natural question arises: can we instead apply a non-linear kernel which would suppress anti-learning?

First note that several non-linear kernels can be expressed as a composition with the linear kernel $k = \varphi \circ k_{lin}$. For instance, the polynomial kernel of degree d, $k_d = (k_{lin} + b)^d$ or the Gaussian kernel

$$k_\sigma(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) = \exp\left(-\frac{\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2k_{lin}(\mathbf{x}_i, \mathbf{x}_j)}{\sigma^2}\right).$$

If k is a class symmetric kernel satisfying (3) then $k_\sigma = \exp\left(-2(r^2 - k)/\sigma^2\right)$, thus it is a composition of $k_\sigma = \varphi_1 \circ k$, where $\varphi_1 : \xi \in \mathbb{R} \mapsto \exp\left(-2(r^2 - \xi)/\sigma^2\right)$ is a monotonically increasing function. Similarly, for the odd degree $d = 1, 3, ...$ we have $k_d = \varphi_2 \circ k$, where $\varphi_2 :\ \xi \in \mathbb{R} \mapsto (\xi + b)^d$, $\xi \in \mathbb{R}$, is a monotonically increasing function.

But when this composition function is a monotonically increasing one, the relative order of c_y and c_0 in the new non-linear kernel matrix will be unchanged and anti-learning will persist.

Corollary 1. *Let k, k' be two positive definite kernels on S such that $k' = \varphi \circ k$ where $\varphi : \mathbb{R} \to \mathbb{R}$ is a function monotonically increasing on the segment $(a, b) \subset \mathbb{R}$ containing all c_{-1}, c_{+1} and c_0. Then if one of these kernels is class symmetric, then the other on is too; if one is perfectly anti-learning (perfectly learning, respectively), then the other one is too.*

In the next section, we introduce a non-monotonic modification of the kernel matrix which can overcome anti-learning.

3.2 From Anti-Learning to Learning

Consider the special case of a $CS-$kernel with $c_0 = 0$, i.e. when examples from opposite labels are positioned on two mutually orthogonal hyperplanes. According to theorem 1, anti-learning will occur if $c_y < 0$, for $y \in \{\pm 1\}$. A way to reverse this behavior would be for instance to take the absolute values of the kernel matrix such that the new c_y becomes positive. Then, according to Theorem 1 again, perfect learning could take place.

This is the main idea behind the following theorem which makes it possible to go from anti-learning to learning, for the CS-kernel case at least. Its main task here is to establish that the new kernel matrix is positive definite.

Theorem 2. *Let k be a positive definite CS-kernel on S and $\varphi : \mathbb{R} \to \mathbb{R}$ be a function such that $\varphi(0) = 0$ and*

$$0 < \varphi(-\theta) \leq \varphi(\theta) \leq \varphi(\theta') \qquad \text{for } 0 < \theta \leq \theta'. \tag{12}$$

Then $k_\varphi := \varphi(k - c_0)$, is a positive definite CS-kernel on S satisfying the perfect learning condition (9) of Theorem 1.

Note that this theorem shows an advantage of using kernels for formulation results of this paper. The claims of this result are straightforward to formulate and prove using CS-kernels but next to impossible do it directly.

Proof. It is easy to see that the kernel k_φ satisfies conditions (3) of CS-symmetry with coefficients $r_\varphi^2 := \varphi(r^2(1-c_0)) > 0$, $c_{\varphi,0} = 0$ and $c_{\varphi,y} := \varphi\big(r^2(c_y - c_0)\big)/\varphi\big(r^2(1-c_0)\big) > c_{\varphi,0} = 0$ for $y = \pm 1$. Now a straightforward algebra gives the relation

$$D_{\varphi,y} = \frac{1-c_{\varphi,y}}{m_y} + c_{\varphi,y} \geq c_{\varphi,y}\left(1 - \frac{1}{n_y}\right) > 0 = |c_{\varphi,0}|,$$

hence the first condition of (4) holds. The second condition of (4) is equivalent to $\varphi\big(r^2(c_y - c_0)\big) < \varphi\big(r^2(1-c_0)\big)$. When $c_y - c_0 \geq 0$, this is satisfied thanks to (12) and the second condition of (4). When $c_y - c_0 \leq 0$, we have $2c_0 - c_y < c_y + 2\frac{1-c_y}{n_y} = c_y(1-\frac{2}{n_y}) + \frac{2}{n_y} < 1$, the first (resp. second) inequality coming from the first (resp. second) condition of (4). This gives $-(c_y - c_0) < 1 - c_0$ and the desired result through (12). □

The examples of functions φ satisfying the above assumptions are $\theta \mapsto |\theta|^d$, $d = 1, 2, ...$, leading to a family of Laplace kernels $|k - c_0|^d$ [13]; this family includes polynomial kernels $(k - c_0)^d$ of even degree d. This is illustrated by simulation results in Figure 2.

4 Discussion

CS-polyhedrons in real life. The CS-polyhedron is a very special geometrical object. Can it emerge in more natural settings? Surprisingly the answer is positive. For instance, Hall, at. al. have shown that it emerges in a high dimensions for low size sample data [6]. They studied a non-standard asymptotic, where dimension tends to infinity while the sample size is fixed. Their analysis shows a tendency for the data to lie deterministically at the vertices of a CS-polyhedron. Essentially all the randomness in the data appears as a random rotation of this polyhedron. This CS-polyhedron is always of the perfect learning type, in terms of Theorem 1. All abnormalities of support vector machine the authors have observed, reduce to sub-optimality of the bias term selected by the maximal margin separator.

CS-polyhedron structure can be also observed in some biologically motivated models of growth under constraints of competition for limited resources [7, 8]. In this case the models can generate both perfectly learning and perfectly anti-learning data, depending on modeling assumptions.

Relation to real life data. As mentioned already in the introduction, the example of the Aryl Hydrocarbon Receptor used in KDD'02 Cup competition is unsual: this dataset shows strong anti-learning while modeled by ordinary two-class SVM, but is "learnable" if non-standard one-class SVM is used, and this behavior changes abruptly if the continuous transition from one model to

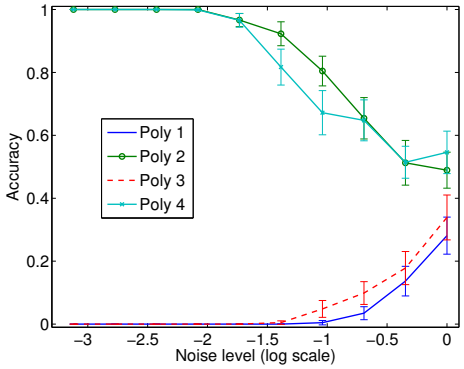

Fig. 2. Switch from anti-learning to learning upon non-monotonic transformation of the kernel, Theorem 2. Plots show independent test set accuracy, average over 30 trials. For the experiments we have used Hadamard data set $S(H_{128})$, see Example 1 of Section 2.2, with the gaussian noise $\mathcal{N}(0,\sigma)$ added independently to each data matrix entry. The data set has been split randomly into training and test sets (respectively two thirds and one third). We have used the hard margin SVM exclusively, so the training accuracy was always 1. We plot averages for 30 repeats of the experiments and also the standard deviation bars. We have used four different kernels, the linear kernel $k_1 = k_{lin}$ and its three non-monotonic transformations, $k_d := (k_{lin} - \hat{c_0})^d$, $d = 2, 3, 4$, where $\hat{c_0} := \text{mean}_{y_i \neq y_j} \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}$ was estimated from the training data.

the other is used [9, 11]. In [7] this has been also shown to be the case for a dedicated CS-polyhedron model, which with an addition noise, reproduces closely also other results observed for the KDD'02 data.

Anti-learning is not over-fitting. By over-fitting a supervised learning algorithm we understand a generation of a classifier performing on a level of random guessing (with a bias reflecting the class priors, perhaps). Such a classifier does not allow to discern any useful information for classification of independent data, providing expected $AROC \approx 0.5$. This is not the case of anti-learning. In its perfect form (Theorem 1) we obtain a predictor f which perfectly misclassify the off-training data, $AROC(f, S \backslash T) = 0$, hence its negation, $-f$, allows to recover the ordering fully consistent with labels, $AROC(-f, T - S) = 1$.

Noise Suppresses Anti-Learning. Furthermore, the anti-learning occurs for a data set, or more generally a kernel, with the special "symmetries". Addition of random noise suppresses these symmetries and kills the anti-learning effect. This is vividly demonstrated in Figure 2, where in tune with the increased variance of noise in the data, the average AROC on the test set for the linear SVM increases from ≈ 0 to the average level of random guessing, ≈ 0.5. At the same time, for the transformed quadratic kernel SVM, for which learning occurs in line with the predictions of Theorem 2, the average AROC decreases from ≈ 1 to the average level of random guessing, ≈ 0.5.

Relation to some other Research. Anti-learning should not be mixed with No-Free-Lunch theorems promoted by Wolpert [15]. The "No-Free-Lunch" type results make statements about averages across all target concepts (this is the crux of all its incarnations), while anti-learning deals with a single, very special, target concepts at a time and makes statements on behavior of wide classes of algorithms. However, anti-learnable datasets are constructive examples of data where many standard supervised learning algorithms has high error rate on the off-training data, a phenomenon envisaged by No-Free-Lunch Theorems.

There are some similarities of this paper with another recent one [14]. Both studies make use of Hadamard matrices, though in a different way. Also the both studies discuss problems with using classifiers within the span of the training set. However, the relation of both studies is not quite clear at this point. After a quite extensive joint study of the anti-learning problem by one of us (AK) and both authors of [14] in late 2004 we saw quite a few differences. The four main differences I (AK) summarize as follows. (*i*) The paper [14] makes statements exclusively about regression (squared loss), while this one exclusively about classification (AROC or ACC losses). The study of the square loss for regression for CS-kernel is full of its own surprises such as the divergence of test loss to infinity, etc., which were not observed in [14]; however, this is beyond scope of this paper and will be covered by some of our future papers. (*ii*) Like in the case of No-Free-Lunch Theorem, claims in [14] are about averages across multiple learning target concepts and also across both, the combined the training and the test sets. However, in this paper we are in position to evaluate the performance on each of these data sets separately and for a single target concept at a time. (*iii*) In [14], the target vector is still part of the data matrix. In our setting we have removed the row corresponding the target vector from the data matrix and additionally one column, though the latter has a marginal importance. So formally, we study different problems. (*iv*) In our study Hadamard matrix is only one example of CS-polyhedrons displaying anti-learning. This is clear from the main Theorem 1. Thus our results are applicable to a wider class of datasets.

Deceptive simplicity of classification of CS-polyhedrons. In the perfect anti-learning setting two labeled samples are sufficient to classify perfectly all data. To be concrete, let us consider the perfect anti-learning CS-kernel k on $(\mathbf{x}_i, y_i)_{i\in\mathbb{S}} \subset \mathbb{R}^N$, see (6). Given two labeled examples, say i_o and j_o, the classification rule

$$\mathbf{x} \in \mathbb{R}^N \mapsto = \begin{cases} y_{j_o}, & \text{if } k(\mathbf{x}_{i_o}, \mathbf{x}) = k(\mathbf{x}_{i_o}, \mathbf{x}_{j_o}); \\ -y_{j_o}, & \text{otherwise,} \end{cases}$$

perfectly allocates labels the whole space S. However, many well established "work-horses" of machine learning, such as SVM, ridge regression, perceptron, voting perceptron, k-nearest neighbours is provably unable to discover solutions (existing in their hypothesis spaces) doing this task, even when given all but two points for the training. Thus this is not the problem of "inadequate" hypothesis space or too poor representation. In fact the algorithms always find rules which systematically miss-order any two data points of different labels not included in the training. And boosting these "weak learners" leads to the similar result.

All this can be interpreted as follows. These learning algorithms are unable to estimate the distribution of labels in the data space, since the samples is too small. However, they are capable of discerning hidden "patterns" in the data very efficiently. Thus they learn although in a different way than expected. How to harness this capability is another issue.

Non-linear classifiers. Theorem 2 shows however, that at least in some situations, and non-linear transformation of the kernel can convert an anti-learning task into a learning task. In future we will present also some ensemble based machine learning algorithms capable achieving this goal as well.

Relation to Machine Learning Theory. Anti-learning is relevant to special cases of the classification, when inference is to be done form a very small training sample from a very high dimensional space. This occurs especially in bio-informatics [9, 11, 8, 7]. It is worth to note that this is the regime where ordinary machine theory and statistics is void. In particular, this paper does not contradict the VC theory [13] which states that the training error and the test error should be close to each other when the number of training samples is significantly larger than the VC dimension (i.e. capacity) of the learning system. In all the anti-learning examples we encountered here the number of learning examples is always not larger than the number of dimensions (i.e. VC-dimension).

Learning Distributed Concepts An interesting observation on anti-learning has been made by J. Langford [10], where anti-learning is linked to learning concepts which are distributed, rather than concentrated.

Final Summary. This study is a primarily introduction to a phenomenon of anti-learning. We have concentrated on a very simple synthetic data set for which anti-learning can be demonstrated formally for a large classes of learning algorithms. This dataset class is so simple that it can be analyzed analytically, but it is also reach enough to demonstrate unexpected and novel behavior of many "standard" learning algorithms. But we must stress, that the motivation for this research comes from real life cancer genomic data sets (unpublished at this stage) which consistently display anti-learning behavior. Obviously, this is not exactly the perfect anti-learning, but rather a consistent performance below random guessing in independent tests. So this paper is an initial step in an attempt to understand properties of some real datasets and ultimately to work out the practical ways to deal with such non-standard leaning problems. Its aim is also to build awareness and initial acceptance for this class of learning problems and to encourage other researchers to come forward with datasets which do display such "counter-intuitive" properties, rather than dismissing them as nonsense. (This last point reflect also our personal experience.)

We do not draw any definite conclusions in this paper, as to whether and how anti-learning data sets should be dealt with. It is too early for that. Our Theorem 2 says how to deal with some classes of anti-learning data, i.e. the CS-kernels or, equivalently, of CS-polyhedrons. However, such transformations are ineffective for noisy real life anti-learning data we are interested in. Thus alternative, more robust techniques to deal with this issue are still to be researched and potentially to be developed. We would like to add that the standard approach to learning

from a small size sample, namely aggressive feature selection, does not solve the problem, at some real life cases at least. In particular, this is demonstrated by experimental results reported in [8, 11].

Future Research. There is a number of directions this research can be extended to in future. We shall list some of our current preferences now.

1. Identify and research novel examples of anti-learning datasets, both synthetic and natural.
2. Develop techniques for consistent classification of anti-learning data.
3. Research techniques capable of seamless learning from both learnable and anti-learnable datasets.
4. Study the problem of regression (square loss) for the anti-learning datasets.
5. Research iid sampling models, in particular learning curves, for the anti-learning datasets.

Conclusions

Anti-learning does occur in some machine learning tasks when inference is done from very low sample sizes in high dimensional feature spaces. This warrants radical re-thinking of basic concepts of learnability and generalization which are currently totally biased towards the "learning mode" of discerning the knowledge from data. It also warrants further research into theoretical analysis and development of practical methods for dealing with anti-learning problems, since such do occur in important real life applications.

Acknowlegements

Many thanks to Cheng Soon Ong, Alex Smola and Grant Baldwin for help in preparation of this paper and to Manfred Warmuth and S.V.N. Vishwanathan for clarifying discussions.

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Council through Baking Australia's Ability and the ICT Center of Excellence program.

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. D. Bamber, *The area above the ordinal dominance graph and the area below the receiver operating characteristic graph*, J. Math. Psych. **12** (1975), 387 – 415.
2. C. Cortes and V. Vapnik, *Support vector networks*, Machine Learning **20** (1995), 273 – 297.

3. M. Craven, *The Genomics of a Signaling Pathway: A KDD Cup Challenge Task*, SIGKDD Explorations **4(2)** (2002).
4. N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, Cambridge, 2000.
5. Y. Freund and R.E. Schapire, *Large margin classification using the perceptron algorithm*, Machine Learning **37** (1999), 277–296.
6. P. Hall, J. S. Marron, and A. Neeman, *Geometric representation of high dimension low sample size data*, preprint, to appear in the Journal of the Royal Statistical Society, Series B, 2005.
7. A. Kowalczyk, O. Chapelle, and G. Baldwin, *Analysis of the anti-learning phenomenon*, http://users.rsise.anu.edu.au/∼akowalczyk/antilearning/, 2005.
8. A. Kowalczyk and C.S. Ong, *Anti-learning in binary classification*, http://users.rsise.anu.edu.au/∼akowalczyk/antilearning/, 2005.
9. A. Kowalczyk and B. Raskutti, *One Class SVM for Yeast Regulation Prediction*, SIGKDD Explorations **4(2)** (2002).
10. J. Langford, 2005, http://hunch.net/index.php?p=35.
11. B. Raskutti and A. Kowalczyk, *Extreme re-balancing for SVMs: a case study* , SIGKDD Explorations **6 (1)** (2004), 60–69.
12. B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, 2001.
13. V. Vapnik, *Statistical learning theory*, Wiley, New York, 1998.
14. M.K. Warmuth and S.V.N. Vishwanathan, *Leaving the Span*, COLT 2005, to appear.
15. D.H. Wolpert, *The supervised learning no-free-lunch theorems*, World Conference on Soft Computing 2001.

Appendix: Proof of Lemma 1, Section 2

Equivalence $(ii) \iff (iii)$ is a standard linear algebra result. The vectors $\mathbf{z}_i$ can be found with a Cholesky decomposition of k. Note that they are linearly independent if and only if the Gram matrix k is is non-singular.

We prove the crucial equivalence $(i) \iff (ii)$ now. In order to simplify the notation, without lost of generality we may assume that $r^2 = 1$. Let us assume that indices are ordered in such a fashion that $y_i = +1$ for $1 \leq i \leq n_+$ and $y_i = -1$ for $n_+ < i \leq n_+ + n_-$. The kernel matrix can be written as follows

$$k = [k_{ij}] = \begin{bmatrix} (1-c_+)\mathbb{I}_{n_+} + c_+ & c_0 \\ c_0 & (1-c_-)\mathbb{I}_{n_-} + c_- \end{bmatrix},$$

where $\mathbb{I}_n$ is the $(n \times n)$ identity matrix. First we observe that k being symmetric has $n_+ + n_-$ linearly independent eigenvectors with real eigenvalues. Now observe that for any vector $\mathbf{v} = (v_i) = ((\mathbf{v}_{+,i}), 0) \in \mathbb{R}^{n_+} \times \mathbb{R}^{n_-}$ such that $\sum_{i=1}^{n_+} v_{+,i} = 0$ we have $k\mathbf{v} = (1 - c_+)\mathbf{v}$. Thus this is an eigenvector with eigenvalue $\lambda = 1 - c_+$. Obviously the subspace of such vectors has dimensionality $(n_+ - 1)$. Similarly we find $(n_- - 1)$ dimensional subspace of vectors of the form $(0, \mathbf{v}_-) \in \mathbb{R}^{n_+} \times \mathbb{R}^{n_-}$ with eigenvalues $\lambda = 1 - c_-$.

The remaining two linearly independent eigenvectors are of the form $\mathbf{v} = (v_+, ..., v_+, v_-, ..., v_-) \in \mathbb{R}^{n_+} \times \mathbb{R}^{n_-}$, where $v_+, v_- \in \mathbb{R}$. For such a vector the eigenvector equation $kbv = \lambda \mathbf{v}$ reduces to two linear equations:

$$\begin{bmatrix} n_+ D_+ - \lambda & n_- c_0 \\ n_+ c_0 & n_- D_- - \lambda \end{bmatrix} \begin{bmatrix} v_+ \\ v_- \end{bmatrix} = \lambda \begin{bmatrix} v_+ \\ v_- \end{bmatrix}.$$

This 2×2 matrix has positive eigenvalues if and only if its determinant and trace are positive or equivalently if $D_+ D_- > c_0^2$ and $D_y > 0,\ y = \pm 1$. $\square$