An Average Classification Algorithm

Brendan van Rooyen*†, Aditya Krishna Menon†∗

*The Australian National University †National ICT Australia

{brendan.vanrooyen, aditya.menon}@nicta.com.au

Abstract

Many classification algorithms produce a classifier that is a weighted average of kernel evaluations. When working with a high or infinite dimensional kernel, it is imperative for speed of evaluation and storage issues that as few training samples as possible are used in the kernel expansion. Popular existing approaches focus on altering standard learning algorithms, such as the Support Vector Machine, to induce sparsity [21, 24], as well as post-hoc procedures for sparse approximations [12, 23]. Here we adopt the latter approach. We begin with a very simple classifier, given by the kernel mean

\[ f(x) = \frac{1}{n} \sum_{i=1}^{n} y_i K(x_i, x) \]

as in chapter one section one of [26]. We then find a sparse approximation to this kernel mean via herding [30, 9, 3]. The result is an accurate, easily parallelized algorithm for learning classifiers.

1 Basic Notation

Let \( Y^X \) be the set of functions with domain \( X \) and range \( Y \). Let \( P(X) \) be the set of probability distributions on a set \( X \). Denote by \( \mathcal{H} \) an abstract Hilbert space, with inner products \( \langle v_1, v_2 \rangle \). For any \( v \in \mathcal{H} \), denote by \( \hat{v} \) the unit vector in the direction of \( v \). We work with loss functions \( L : \{-1, 1\} \times \mathbb{R} \rightarrow \mathbb{R} \). Finally, for a boolean predicate \( p : X \rightarrow \{\text{True}, \text{False}\} \), let \( \lfloor p(x) \rfloor = 1 \) if \( p(x) \) is true and 0 otherwise.

2 Kernel Classifiers

Let \( X \) be the instance space and \( Y = \{-1, 1\} \) the label space. A classifier is a function \( f \in [-1, 1]^X \), with \( \text{sign}(f(x)) \) the predicted label for a given instance \( x \). A classification algorithm is a function \( A : \bigcup_{n=1}^{\infty} (X \times Y)^n \rightarrow \mathbb{R}^X \), that given a training set \( S \) outputs a classifier. Define the misclassification loss \( L_{01}(y, v) = \lfloor yv < 0 \rfloor \). For any loss \( L \), define the risk and sample risk of \( f \) as

\[ \mathcal{R}_L(P, f) := \mathbb{E}_{(x, y) \sim P} L(y, f(x)) \] and \[ \mathcal{R}_L(S, f) := \frac{1}{|S|} \sum_{(x, y) \in S} L(y, f(x)) \]

respectively. Assuming that the \((x, y)\) are drawn iid from a distribution \( P \), good classification algorithms should output classifiers with low misclassification risk. Many classification algorithms, such as the support vector machine, logistic regression, boosting and so on, output a classifier of the form

\[ f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) \]

with \( \alpha_i \geq 0 \), \( \sum \alpha_i = 1 \) and \( K(x, x') = \langle \phi(x), \phi(x') \rangle \): a kernel function, an inner product of feature vectors in a (possibly infinite dimensional) feature space. It is imperative for fast evaluation of the outputted classifier that there are as few non-zero \( \alpha_i \) as possible [12, 23, 21, 24]. One approach for learning sparse classifiers is
to alter the learning algorithm to promote sparsity. Here we use a post-hoc sparse approximation, applicable to any kernel classifier. Given an \( f \) of the form above, we use kernel herding \([30, 29, 3]\) to produce a sparse approximation

\[
\tilde{f}(x) = \sum_{i=1}^{n} \alpha_i y_i' K(x_i, x)
\]

with \( n \ll N \). We are free to start with any \( f \) (the output of a SVM for example); for simplicity, we use the mean \( f(x) = \frac{1}{N} \sum_{i=1}^{N} y_i K(x, x) \). For any sample \( S \in \cup_{n=1}^{\infty} (X \times Y)^n \) define the mean vector \( \omega_S = \frac{1}{|S|} \sum_{(x,y) \in S} y_v \phi(x) \). The mean classifier can then be written as

\[
f(x) = \langle \omega_S, \phi(x) \rangle.
\] (1)

We first argue for the mean classifier, before discussing how to sparsely approximate it.

## 3 Why the Mean?

The classifier in equation (1) is a simple and intuitive classification rule. It classifies by the average similarity of the instance to the previously observed positive and negative instances, with the most similar class being the output of the classifier. It has been studied previously, for example in chapter one of \([26]\) and further in \([14, 5]\). The main drawbacks of (1) are prohibitive storage and evaluation costs. In fact, this is the motivation given for the Support Vector Machine (SVM) in \([26]\). The SVM can be seen as one means of sparsifying (1). Herding provides another.

The classification rule (1) is not only intuitively appealing, it also arises as the optimal classifier for the following linear loss considered previously in \([25, 28]\). Let \( L_{\text{linear}}(y, v) = 1 - y_v \). If \( v \in \{-1, 1\} \), then \( L_{01}(y, v) = \frac{1}{2} L_{\text{linear}}(y, v) \). Allowing \( v \in [-1, 1] \) provides a convexification of misclassification loss. For \( v \in [-1, 1] \), \( L_{01}(y, v) \leq L_{\text{linear}}(y, v) \). Furthermore, we have the following surrogate regret bound.

**Theorem 1** (Surrogate Regret Bound for Linear Loss (\([29]\), proposition 6)). For all distributions \( P \)

\[
f^* = \arg \min_{f \in [-1, 1]^X} \mathcal{R}_{\text{linear}}(P, f) \in \arg \min_{f \in [-1, 1]^X} \mathcal{R}_{01}(P, f).
\]

Furthermore for all \( f \in [-1, 1]^X \)

\[
\mathcal{R}_{01}(P, f) - \mathcal{R}_{01}(P, f^*) \leq \mathcal{R}_{\text{linear}}(P, f) - \mathcal{R}_{\text{linear}}(P, f^*).
\]

By theorem 1 linear loss is a suitable surrogate loss for learning classifiers much like the hinge, logistic and exponential loss functions \([25]\). As usual, rather than minimizing over all bounded functions, to avoid overfitting the sample we work with a restricted function class. For a feature map \( \phi : X \to \mathcal{H} \), define the linear function class

\[
\mathcal{F}_\phi := \{f_\omega(x) = \langle \omega, \phi(x) \rangle : \omega \in \mathcal{H}, \|\omega\| \leq 1 \}.
\]

We will assume throughout that the feature map is bounded, \( \|\phi(x)\| \leq 1 \) for all \( x \). As shorthand we write \( \mathcal{R}(P, \omega) := \mathcal{R}(P, f_\omega) \). By the Cauchy-Schwarz inequality \( \mathcal{F}_\phi \subseteq [-1, 1]^X \).

**Theorem 2** (The Mean Classifier Minimizes Linear Loss).

\[
\hat{\omega}_S = \arg \min_{\omega : \|\omega\| \leq 1} \frac{1}{N} \sum_{(x,y) \in S} 1 - y(\omega, \phi(x)) = \arg \min_{\omega : \|\omega\| \leq 1} 1 - \langle \omega, \omega_S \rangle
\]

with minimum linear loss given by \( 1 - \|\omega_S\| \). Furthermore classifying using \( \langle \hat{\omega}_S, \phi(x) \rangle \) is equivalent to classifying according to equation (1).
This has been noted in [28], we include it for completeness. The proof is a straight forward application of the Cauchy-Schwarz inequality. As \( \hat{\omega}_S = \lambda \omega_S \), \( \lambda > 0 \), they both produce the same classifier. The quantity

\[
\| \omega \|^2 = \frac{1}{|S|^2} \sum_{(x,y) \in S} \sum_{(x',y') \in S} y y' K(x, x')
\]

can be thought of as the “self-similarity” of the sample. For a distribution \( P \), define \( \omega_P = \mathbb{E}_{(x,y) \sim P} y \phi(x) \). It is easily verified that

\[
\hat{\omega}_P = \arg \min_{\omega : \|\omega\| \leq 1} \mathbb{E}_{(x,y) \sim P} 1 - y \langle \omega, \phi(x) \rangle = \arg \min_{\omega : \|\omega\| \leq 1} 1 - \langle \omega, \omega_P \rangle.
\]

Furthermore, we have the following generalization bound on the linear loss performance of \( \hat{\omega}_S \).

**Theorem 3** (Mean Classifier Generalization Bound). For all distributions \( P \) and all feature maps \( \phi : X \rightarrow \mathcal{H} \), with probability at least \( 1 - \delta \) on a draw \( S \sim P^N \)

\[
\mathcal{R}_{\text{linear}}(P, \hat{\omega}_S) \leq \mathcal{R}_{\text{linear}}(S, \hat{\omega}_S) + 2 \sqrt{\frac{1 + \log(\frac{1}{\delta})}{N}}.
\]

The proof is via an appeal to the PAC-Bayes theorem and is included in the appendix. Also in the appendix is a bound on the estimation error in using the sample mean, which is a specific case of more general bounds presented in [5]. We now discuss connections to existing techniques.

### 3.1 Learning with Symmetric Label Noise

In learning with symmetric label noise [2], the learner has access to samples from a corrupted distribution \( \hat{P} \), where noise has been added to the labels of \( P \). To sample from \( \hat{P} \), sample \((x, y) \sim P \) and then flip the label \( y \) with probability \( \sigma \). Many standard loss functions used for learning classifiers, such as hinge loss, are not robust to symmetric label noise, i.e.

\[
\arg \min_{f \in \mathcal{F}_\phi} \mathcal{R}_L(\hat{P}, f) \neq \arg \min_{f \in \mathcal{F}_\phi} \mathcal{R}_L(P, f).
\]

In [29] it is shown that linear loss is robust to symmetric label noise.

### 3.2 Relation to Maximum Mean Discrepancy

Let \( P_+ \in \mathbb{P}(X) \) be the conditional distribution over instances given a positive or negative label respectively. For a feature map \( \phi \) define the Maximum Mean Discrepancy (MMD) [18] to be

\[
D_\phi(P_+, P_-) := \max_{\omega : \|\omega\| \leq 1} \frac{1}{2} |\mathbb{E}_{x \sim P_+} \langle \omega, \phi(x) \rangle - \mathbb{E}_{x \sim P_-} \langle \omega, \phi(x) \rangle| = \| \omega_{P_+} - \omega_{P_-} \|.
\]

\( D_\phi(P_+, P_-) \) can be seen as a restricted variational divergence

\[
V(P_+, P_-) = \max_{f \in [-1,1]^X} \frac{1}{2} |\mathbb{E}_{x \sim P_+} f(x) - \mathbb{E}_{x \sim P_-} f(x)|
\]

a commonly used metric on probability distributions [25], where \( f \in \mathcal{F}_\phi \subseteq [-1,1]^X \). Define the distribution \( P \in \mathbb{P}(X \times Y) \) that first samples \( y \) uniformly from \( \{-1,1\} \) and then samples \( x \sim P_y \). Then

\[
D_\phi(P_+, P_-) = \max_{\omega : \|\omega\| \leq 1} |\mathbb{E}_{(x,y) \sim P} \langle \omega, y \phi(x) \rangle| = \| \omega_P \|.
\]

Therefore, if we assume that positive and negative classes are equally likely, the mean classifier classifies using the \( \omega \) that “witnesses” the MMD.
3.3 Relation to the SVM

For a regularization parameter \( \lambda \), the SVM solves the following convex objective

\[
\min_{\omega} \frac{1}{|S|} \sum_{(x,y) \in S} [1 - y\langle \omega, \phi(x) \rangle]_+ + \lambda \|\omega\|^2
\]

where \([x]_+ = \max(x, 0)\). This is the Lagrange multiplier problem associated with

\[
\min_{\omega:||\omega||^2 \leq c} \frac{1}{|S|} \sum_{(x,y) \in S} [1 - y\langle \omega, \phi(x) \rangle]_+.
\]

If we take \( c = 1 \), by Cauchy-Schwarz \([1 - y\langle \omega, \phi(x) \rangle]_+ = 1 - y\langle \omega, \phi(x) \rangle\) and the above objective is equivalent to that in theorem 2. The mean classifier is the optimal solution to a highly regularized SVM, and is therefore preferentially optimizing the margin over the sample hinge loss. Prior evidence exists suggesting that high regularization and feature normalization (regularization in disguise) is a good idea [15].

3.4 Relation to Kernel Density Estimation

On the surface the mean classifier is a discriminative approach. Restricting to positive kernels, such as the Gaussian kernel \( K(x,x') = e^{-\frac{|x-x'|^2}{2\sigma^2}} \), it can be seen as the following generative approach: estimate \( P \) with \( \tilde{P} \), with class conditional distributions estimated by kernel density estimation. Letting \( S_{\pm} = \{(x, \pm 1)\} \subseteq S \), set

\[
\tilde{P}(X = x|Y = \pm 1) \propto \frac{1}{|S_{\pm}|} \sum_{x' \in S_{\pm}} K(x, x')
\]

and \( \tilde{P}(Y = 1) = \frac{|S_+|}{|S|} \). To classify new instances, use the Bayes optimal classifier for \( \tilde{P} \). This yields the same classification rule as [1]. This is the “potential function rule” discussed in [13].

3.5 Extension to Multiple Kernels

To ensure the practical success of any kernel based method, it is imperative that the correct feature map be chosen. Thus far we have only considered the problem of learning with a single feature map, and not the problem of learning the feature map at the same time. Given \( k \) feature maps \( \phi_i : X \rightarrow H_i, i \in [1; k] \), multiple kernel learning [4, 20, 10] considers learning over a function class that is the convex hull of the classes \( \mathcal{F}_{\phi_i} \),

\[
\mathcal{F} := \left\{ f(x) = \sum_{i=1}^{k} \alpha_i \langle \omega^i, \phi_i(x) \rangle : \|\omega^i\| \leq 1, \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1 \right\}.
\]

Let \( \Delta_k \) be the \( k \) simplex. By an easy calculation,

\[
\min_{f \in \mathcal{F}} \frac{1}{|S|} \sum_{(x,y) \in S} 1 - yf(x) = \min_{\alpha \in \Delta_k, \omega^i} \sum_{i=1}^{k} \alpha_i (1 - \langle \omega_i, \omega^i_S \rangle)
\]

\[
= \min_{i \in [1;k]} (1 - \|\omega^i_S\|)
\]

where \( \omega^i_S \) is the sample mean in the \( i \)th feature space, and the second follow from the linearity of the loss. In words, we pick the feature space which minimizes \( 1 - \|\omega^i_S\| \). This is in contrast to usual multiple kernel learning techniques that do not in general pick out a single feature map. Furthermore, we have the following generalization bound.

**Theorem 4** (Multiple Mean Classifier Generalization Bound). For all distributions \( P \) and for all finite collections of feature maps \( \phi_i : X \rightarrow H_i, i \in [1; k] \), with probability at least \( 1 - \delta \) on a draw \( S \sim P^N \)

\[
\mathcal{R}_{\text{linear}}(P, \omega^i_S) \leq \mathcal{R}_{\text{linear}}(S, \omega^i_S) + 2\sqrt{\frac{(1 + \log(k) + \log(\frac{1}{\delta}))}{N}}.
\]
where $\omega_\mathcal{S}^*$ is the kernel mean that minimizes $1 - \|\omega_\mathcal{S}\|$.

The proof proceeds in the same way as theorem 3.

## 4 Herding for Sparse Approximation

The only problem with using $\omega_\mathcal{S}$ is its dependence on the entire sample. We show how to correct this.

For any set $Z$, mapping $\psi : Z \to \mathcal{H}$ and distribution $P \in \mathbb{P}(Z)$, define the mean $\omega_P = \mathbb{E}_{z \sim P} \psi(z)$. We recover our previous definition by taking $Z = X \times Y$ and $\psi(x, y) = y\phi(x)$. Given a convex set $C \subseteq \mathcal{H}$, herding [30, 9, 3] provides means to sparsely approximate $\omega_P$ with $\tilde{\omega} \in C$. In [3] it was shown that herding is an application of the Frank-Wolfe optimization algorithm to the convex problem

$$\min_{\omega \in C} \|\omega_P - \tilde{\omega}\|^2.$$ 

For a set of finite training sample $S \subseteq Z$, we take $C$ to be the convex hull of their feature vectors, $C = \text{Hull}\{\psi(z) : z \in S\}$. Define the kernel $\mathcal{K}(z, z') = \langle \psi(z), \psi(z') \rangle$. Herding proceeds as in algorithm 1.

Intuitively, herding begins by selecting the point in $S$ that is most similar to $P$ on average as measured by $\mathcal{K}$. When selecting a new representative, herding chooses the point in $S$ that is most similar on average to draws from $P$ while being different from previously chosen points. If herding runs for $m$ iterations, then an approximation of $\omega_P$ with only $m$ elements is obtained. One can also take $\alpha = \frac{1}{|H|+1}$, leading to uniform weights (termed uniform herding). The step size for line search is available in closed form [3].

Herding can also be viewed as minimizing $D_\psi(P, Q)$, where the approximating distribution $Q$ is concentrated on $S$ [9]. Originally, herding was motivated as means to produce “super samples” from a distribution $P$. Standard monte-carlo techniques lead to convergence at rate $\frac{1}{\sqrt{m}}$ of the square error $\|\omega_P - \tilde{\omega}\| \to 0$. Using herding, faster rates can be achieved.

**Data:** Distribution $P \in \mathbb{P}(Z)$, set of possible representative points $S \subseteq Z$, kernel function $\mathcal{K}(z, z')$ and error tolerance $\epsilon$.

**Result:** Weighted set of representatives $H = \{(\alpha_i, z_i)\}_{i=1}^n$ such that $\left\|\omega_P - \sum_{(\alpha, z) \in H} \alpha \psi(z_i)\right\| \leq \epsilon$.

**Initialization:** $z = \arg \max_{z \in S} \mathbb{E}_{z' \sim P} \mathcal{K}(z, z')$, $H = \{(\alpha, z)\}$;

**while** $\left\|\omega_P - \sum_{(\alpha, z) \in H} \alpha \psi(z_i)\right\| > \epsilon$ **do**

- Let $z = \arg \max_{z \in S} \mathbb{E}_{z' \sim P} \mathcal{K}(z, z') - \sum_{(\alpha, z') \in H} \alpha \mathcal{K}(z, z')$;
- Pick $\alpha \in [0, 1]$ by line search;
- Multiply all weights in $H$ by $1 - \alpha$;
- Add $(\alpha, z)$ to $H$

**end**

**Algorithm 1:** Herding, see text.

For our application, $P$ is the empirical distribution over the set $S$, $\frac{1}{|S|} \sum_{z \in S} \delta_z$, or equivalently $\omega_S = \frac{1}{|S|} \sum_{z \in S} \psi(z)$. $m$ iterations of the algorithm run in time order $Nm^2$, with a startup cost of $N^2$ to populate the kernel matrix. As we will see, herding converges rapidly: $O(\log(\frac{1}{\epsilon})N)$ time gives an approximation of accuracy $\epsilon$.

### 4.1 Rates of Convergence for Herding

Let $\hat{\omega}_m$ be the approximation to $\omega_P$ obtained from running herding for $m$ iterations. As discussed previously, herding can be used as a means of sampling from a distribution, with rates of convergence $\|\omega_P - \hat{\omega}_m\| \to 0$.
faster than that for random sampling. While in the worst case, one can not do better than a $\frac{1}{\sqrt{m}}$ rate, if $\omega_P \in C$ faster rates can be obtained \cite{3}. Let $D$ be the diameter of $C$ and $d$ the distance from $\omega_P$ to the boundary of $C$. For herding with line search as in algorithm 1,

$$\|\omega_P - \tilde{\omega}_m\| \leq \|\omega_P - \tilde{\omega}_i\| e^{-\alpha m}$$

where $\alpha = \frac{d}{2\pi \|\omega_P\|-D}$ \cite{6}. For our application $\omega_S \in C$, furthermore $d > 0$. Hence the herded approximation converges quickly to $\omega_S$.

### 4.2 Comparisons with Previous Work

Herding has appeared under a different name in the field of statistics \cite{22}. They consider an algorithm closely related to the Frank-Wolfe algorithm (projection pursuit), and prove $\frac{1}{\sqrt{m}}$ convergence for the general case when $\omega_P \notin C$. The appendix of \cite{19} features a theoretical discussion of sparse approximations. They show the existence of a $m$-sparse approximation with $\|\omega - \tilde{\omega}\| \leq \frac{\sqrt{2\epsilon m(S)}}{\sqrt{m}}$, with $\epsilon P (S)$ the entropy numbers of the set $S$. We further explore the connections to their approach in the additional material.

### 4.3 Parallel Extension

It is very easy to use parallelize the herding algorithm. Rewriting the mean as a “mean of means”, one has

$$\frac{1}{N} \sum_{i=1}^{N} \psi(z_i) = \sum_{i=1}^{M} \frac{N_i}{N} \left( \frac{1}{N_i} \sum_{j=1}^{N_i} \psi(z_{ij}) \right)$$

where we have split the $N$ data points into $M$ disjoint groups with $z_{ij}$ the $j$th element of the $i$th group. We can use herding to approximate each sub mean $\frac{1}{N_i} \sum_{j=1}^{N_i} \psi(z_{ij})$ separately. Furthermore, if we approximate each sub mean to tolerance $\epsilon$, combining the approximations yields an approximation to the total mean with tolerance $\epsilon$.

**Lemma 1** (Parallel Means). Let $\omega = \sum \lambda_i \omega_i$ with $\lambda_i \geq 0$ and $\sum \lambda_i = 1$. Suppose that for each $i$ there exists $\tilde{\omega}_i$ with $\|\omega_i - \tilde{\omega}_i\| \leq \epsilon$. Then $\|\omega - \sum \lambda_i \tilde{\omega}_i\| \leq \epsilon$.

The proof is a simple application of the triangle inequality and the homogeneity of norms. Lemma 1 allows one to use a map-reduce algorithm to herd large sets of data. One splits the data into $M$ groups, herds each group in parallel and then combines the groups, possibly herding the result. This algorithm runs in order $(\frac{N}{M})^2$ on each machine used for herding.

### 4.4 Discriminative Herding

Our goal is to approximate equation (1). To this end, we run herding on the sample $S$ with $K((x, y), (x', y')) = y y' K(x, x')$, the discriminative kernel associated with $K$. We take

$$\tilde{\omega}_S = \sum_{(\alpha, (x, y)) \in H} \alpha y \phi(x)$$

where $H$ is the representative set of instance, label pairs obtained from herding $S$ to tolerance $\epsilon$. Our approximate classifier is $\tilde{f}(x) = \langle \tilde{\omega}_S, \phi(x) \rangle$. We have by a simple application of the Cauchy-Schwarz inequality

$$|f(x) - \tilde{f}(x)| = ||\omega_S - \tilde{\omega}_S, \phi(x)|| \leq \epsilon.$$

Hence the tolerance used in the herding algorithm directly controls the approximation accuracy. Figure 2 gives visualization of discriminative herding on a simple two dimensional data set. In blue are the samples from the positive class, in red the samples from the negative class. The large blue and red points are the representative points from each class picked by running discriminative herding with a Gaussian kernel of bandwidth 0.2 to tolerance 0.01. The representatives attempt to define the “vertical boundaries” of each class, as these are important for discriminative purposes for this particular task.
4.5 Relating $\epsilon$ to the Margin

Define the margin loss at margin $\gamma$ to be $L_{\gamma}(y,v) = \left[ yv < \gamma \right]$. For $\gamma = 0$, $L_{\gamma} = L_{01}$. The margin loss is used in place of misclassification loss to produce tighter generalization bounds for minimizing misclassification loss \cite{17, 19, 27}. Maximizing the margin while forcing $R_{\gamma}(S,\omega) = 0$ is the original motivation for the hard margin SVM \cite{11}. Here we relate the margin loss of a classifier $f$ to the amount of slop allowed in approximating $f$.

**Theorem 5 (Margins and Approximation).** Let $f, \tilde{f}$ be classifiers such that $|f(x) - \tilde{f}(x)| \leq \epsilon$ for all $x$. Then for all distributions $P$,

$$R_{01}(P, \tilde{f}) \leq R_{\epsilon}(P, f).$$

For simplicity we will assume $R_{01}(S,\omega_S) = 0$. Setting $\epsilon = \max\{\gamma : R_{\gamma}(S,\omega_S) = 0\}$ ensures $R_{01}(S,\tilde{\omega}_S) = 0$. Furthermore by equation (2), $||\omega_S - \tilde{\omega}_m|| \rightarrow 0$ as $e^{-\alpha m}$. Therefore to obtain an approximation with $||\omega_S - \tilde{\omega}_m|| \leq \epsilon$, order $\log(\frac{1}{\epsilon})$ iterations of herding must be performed. This result is very similar in spirit to those in \cite{17}. There, they use the perceptron algorithm to produce a solution with sparsity of order $\frac{1}{\epsilon^2}$. The margin therefore provides means of assessing the quality of a feature representation: Feature representations which produce large margin classifiers afford sparser approximations. As in \cite{17}, we can establish that compression is good from a generalization perspective.

5 Generalization via Sparsity

Thus far we have motivated the herded mean classifier as a sparse approximation to the mean classifier. We can also motivate herding for any generic classifier through compression bounds \cite{16}.

A classification algorithm $A$ is called a compression scheme if $A = R \circ C$, where $C : \cup_{n=1}^{\infty}(X \times Y)^n \rightarrow \cup_{n=1}^{\infty}(X \times Y)^n$ such that $|C(S)| \leq |S|$ and $R : \cup_{n=1}^{\infty}(X \times Y)^n \rightarrow \mathbb{R}^X$. Intuitively, $C$ compresses the sample $S$ into a smaller set, and $R$ takes this set and produces a classifier.

**Theorem 6 (General Compression Bound (theorem 3 in \cite{16})).** Let $A$ be a compression scheme. Then for all distributions $P$, with probability at least $1 - \delta$ on a draw $S \sim P^N$

$$R_{01}(P, A(S)) \leq \frac{N}{N - |C(S)|} R_{01}(S, A(S)) + \sqrt{\frac{(2 + |C(S)|) \log(N) + \log\left(\frac{1}{\delta}\right)}{N}}.$$  

We can view uniform herding as a compression scheme in two ways. We can either set a tolerance of $\epsilon$ or a maximum herd size $n$. We then define $H = C(S)$ to be the herded sample. We then define $R(H)$ to be the mean classifier associated with the herd

$$f_H(x) = \frac{1}{|H|} \sum_{(x',y') \in H} y'K(x,x').$$
Theorem 6 suggests such an early stopping procedure, that is directly optimizing for sparsity, is justified from the view of generalization. However, the bound from theorem 3 is tighter.

6 Experimental Results

While the main contributions of this paper are theoretical, we include a proof of concept experiment to highlight the performance of herding as a means of compressing data sets. Keeping up with the current fashion, here we consider classifying 3’s versus 8’s from the MNIST data set, comprising 11982 training examples and 1984 test examples in roughly equal proportion. We normalize all pixel values to lie in the interval [0, 1] and use a Gaussian kernel with bandwidth 1. We plot the test set performance of the learned classifier as a function of the size of the herded data set. To produce the plot, we ran parallel herding with 50 splits (roughly 240 data points for each split) for tolerances ranging from 0.025 to 0.65 in steps of 0.025. Each blue dot signifies a herd. As a baseline we take the test set performance of the classifier that uses the sample mean of the entire training set.

The baseline method achieves test set performance of 98.74%. We see that with little as 150 points (less than 1.3% of the training set), an accuracy of over 92% is obtained. The performance of the herded samples rapidly approaches that of the full mean, with 1900 samples (less than 16% of the training set) obtaining accuracy of 97%. The final herd with roughly 10000 elements actually out performs the sample mean of the entire training set. Finally, the empirical margin on the test set is very small, of order $10^{-5}$, indicating that this kernel may not provide the best feature representation for herding. In [29] further experiments are performed to assess the validity of linear loss minimization.

Figure 2: Herding MNIST, see text (best viewed in colour).
7 Concluding Remarks

We have taken a simple classifier, given by the sample mean, and have placed it on a firm theoretical grounding. We have shown its relation to maximum mean discrepancy, highly regularized support vector machines and finally to kernel density estimation. We have proven a surrogate regret bound highlighting its usefulness in learning classifiers, as well as generalization bounds for single and multiple feature maps. Finally we have shown how herding can be used to speed up its evaluation and how the margin is related to the performance degradation incurred by our approximation.
8 Additional Material

8.1 Proof of Theorem 1

**Theorem.** For all distributions $P$

$$f^* = \arg\min_{f \in [-1, 1]^X} \mathcal{R}_{\text{linear}}(P, f) \in \arg\min_{f \in [-1, 1]^X} \mathcal{R}_{01}(P, f).$$

Furthermore for all $f \in [-1, 1]^X$

$$\mathcal{R}_{01}(P, f) - \mathcal{R}_{01}(P, f^*) \leq \mathcal{R}_{\text{linear}}(P, f) - \mathcal{R}_{\text{linear}}(P, f^*).$$

**Proof.** It is well known that $f^* \in \arg\min_{f \in [-1, 1]^X} \mathcal{R}_{01}(P, f)$. From $P$ define $P_X$ to be the marginal distribution over instances and $\eta(x) = P(Y = 1|X = x)$. Then

$$\mathcal{R}_{\text{linear}}(P, f) = \mathbb{E}_{(x, y) \sim P} 1 - yf(x)$$

$$= \mathbb{E}_{x \sim P_X} 1 + (1 - 2\eta(x))f(x).$$

Minimizing over $f \in [-1, 1]^X$ gives $f(x) = -1$ if $1 - 2\eta(x) \geq 0$ ie when $\eta(x) \leq \frac{1}{2}$ and $f(x) = 1$ otherwise. This proves the first claim. We have

$$\mathcal{R}_{\text{linear}}(P, f^*) = \mathbb{E}_{x \sim P_X} 1 - |(1 - 2\eta(x))|.$$

Therefore

$$\mathcal{R}_{\text{linear}}(P, f) - \mathcal{R}_{\text{linear}}(P, f^*) = \mathbb{E}_{x \sim P_X} (1 - 2\eta(x))f(x) + |(1 - 2\eta(x))|$$

$$= \mathbb{E}_{x \sim P_X} |(1 - 2\eta(x))| - \text{sign}(2\eta(x) - 1)(1 - 2\eta(x))|f(x)|$$

$$= \mathbb{E}_{x \sim P_X} |(1 - 2\eta(x))|(1 - \text{sign}(2\eta(x) - 1))f(x)).$$

It is well known [25] that

$$\mathcal{R}_{01}(P, f) - \mathcal{R}_{01}(P, f^*) = \mathbb{E}_{x \sim P_X} |(1 - 2\eta(x))| \text{sign}(2\eta(x) - 1)f(x) \leq 0.$$

We complete the proof by noting $[v \leq 0] \leq 1 - v$ for $v \in [-1, 1]$.

8.2 PAC-Bayesian Bounds for Linear Loss

Here we prove general bounds for use in theorems 3 and 4. Let $\mathcal{F} \subseteq \mathbb{R}^X$. We consider randomized algorithms $A : \cup_{n=1}^\infty (X \times Y)^n \to \mathbb{P}(\mathcal{F})$. For any algorithm $A$, define $\bar{A} : \cup_{n=1}^\infty (X \times Y)^n \to \mathbb{R}^X$, $\bar{A}(S)(x) = \mathbb{E}_{f \sim \mathcal{A}(S)} f(x)$.  

**Theorem 7** (PAC-Bayes Linear Loss theorem). For all distributions $P$, feature maps $\phi : X \to \mathcal{H}$, priors $\pi$, randomized algorithms $A$ and $\beta > 0$

$$\mathbb{E}_{S \sim P^N} - \frac{1}{\beta} \log(\mathbb{E}_{(x, y) \sim P} \mathbb{E}_{f \sim \mathcal{A}(S)} e^{-\beta(1 - yf(x))}) \leq \mathbb{E}_{S \sim P^N} \left[ \mathcal{R}_{\text{linear}}(S, \bar{A}(S)) + \frac{D_{KL}(\mathcal{A}(S), \pi)}{\beta n} \right].$$

Furthermore, with probability at least $1 - \delta$ on a draw from $S \sim P^N$ with $A$, $\pi$ and $\beta$ fixed before the draw

$$-\frac{1}{\beta} \log(\mathbb{E}_{(x, y) \sim P} \mathbb{E}_{\omega \sim \mathcal{A}(S)} e^{-\beta(1 - yf(x))}) \leq \mathcal{R}_{\text{linear}}(S, \bar{A}(S)) + \frac{D_{KL}(\mathcal{A}(S), \pi) + \log(\frac{1}{\delta})}{\beta n}.$$  

**Proof.** This is theorem 2.1 of [31] for the loss $L((x, y), f) = 1 - yf(x)$, coupled with the convexity of $-\log$. 

\[ \Box \]
We term $\mathcal{A}(S)$ the posterior. The left of the inequality is sometimes referred to as the annealed loss. It is an example of a generalized mean. For all $\beta > 0$,
\[
-\frac{1}{\beta} \log(\mathbb{E}_{(x,y) \sim P} \mathbb{E}_{f \sim \mathcal{A}(S)} e^{-\beta(1-yf(x))}) \leq R_{\text{linear}}(P, f)
\]
furthermore, as $\beta \to 0_+$,
\[
-\frac{1}{\beta} \log(\mathbb{E}_{(x,y) \sim P} \mathbb{E}_{f \sim \mathcal{A}(S)} e^{-\beta(1-yf(x))}) \to R_{\text{linear}}(P, f).
\]

For linear function classes we identify $f_A \in \mathbb{F}$ with its weight vector $\omega$. We take $\mathcal{A}(S) \in \mathbb{P}(\mathcal{H})$ and with a slight abuse of notation define $\bar{\mathcal{A}}(S) = \mathbb{E}_{\omega \sim \mathcal{A}(S)} \omega$. We have
\[
\bar{\mathcal{A}}(S)(x) = \mathbb{E}_{\omega \sim \mathcal{A}(S)}(\omega, \phi(x)) = (\bar{\mathcal{A}}(S), \phi(x)) \in \mathcal{F}_\phi.
\]

For linear function classes, the sample risk of the posterior is determined by its mean weight vector. To exploit this, we focus on posteriors and priors of simple form, allowing exact calculation of the annealed loss and the KL divergence term. We assume $\pi = N(\omega, 1)$ and $\mathcal{A}(S) = N(\bar{\mathcal{A}}(S), 1)$. In words, priors and posteriors are normal distributions with identity covariance. This restriction and the following lemma lead to the bound 3.

**Lemma 2.** Let $f : Z \to [0, 2]$. For all $\beta \in \mathbb{R}_+$ and all $P \in \mathbb{P}(Z)$
\[
\mathbb{E}_P f - \frac{\beta}{2} \leq -\frac{1}{\beta} \log(\mathbb{E}_P e^{-\beta f})
\]

**Proof.** See appendix A.1 of [8].

**Theorem 8.** For all distributions $P$, feature maps $\phi$, prior vectors $\omega_\pi \in \mathcal{H}$, sample dependent weight vectors $\bar{A} : (X \times Y)^n \to \mathcal{H}$ and $\beta > 0$ such that $\|\phi(x)\| \leq 1 \forall x$ and $\|\bar{A}(S)\| \leq 1 \forall S$
\[
\mathbb{E}_{S \sim P^n} R_{\text{linear}}(P, \bar{A}(S)) \leq \mathbb{E}_{S \sim P^n} \left[ R_{\text{linear}}(S, \bar{A}(S)) + \frac{\|\bar{A}(S) - \omega_\pi\|^2}{\beta n} \right] + \beta.
\]

Furthermore, with probability at least $1 - \delta$ on a draw from $S \sim P^n$ with $\bar{A}$, $\omega_\pi$ and $\beta$ fixed before the draw
\[
R_{\text{linear}}(P, \bar{A}(S)) \leq R_{\text{linear}}(S, \bar{A}(S)) + \frac{\|\bar{A}(S) - \omega_\pi\|^2 + \log(\frac{1}{\delta})}{\beta n} + \beta.
\]

**Proof.** We begin with theorem 7 and the function class $\mathcal{F}_\phi$. For priors and posteriors given by normal distributions
\[
D_{KL}(\mathcal{A}(S), \pi) = \|\bar{A}(S) - \omega_\pi\|^2.
\]

For the left hand side of the bound,
\[
-\frac{1}{\beta} \log(\mathbb{E}_{(x,y) \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathcal{A}(S)} e^{-\beta(1-y\phi(x))})
\]
\[
= -\frac{1}{\beta} \log(\mathbb{E}_{(x,y) \sim \mathcal{P}} \mathbb{E}_{\omega \sim N(\bar{A}(S), 1)} e^{-\beta(1-y\phi(x))})
\]
\[
= -\frac{1}{\beta} \log(\mathbb{E}_{(x,y) \sim \mathcal{P}} e^{-\beta(1-(\bar{A}(S), \phi(x))) + \frac{\beta^2}{2} \|\phi(x)\|^2})
\]

where the final line follows from standard results on the moment generating function of normal distributions. We can lower bound this quantity as follows
\[- \frac{1}{\beta} \log (E_{(x,y) \sim P} e^{-\beta (1 - \langle \hat{A}(S), y\phi(x) \rangle)} + \frac{\beta^2}{2} \| \phi(x) \|^2) \geq - \frac{1}{\beta} \log (E_{(x,y) \sim P} e^{-\beta (1 - \langle \hat{A}(S), y\phi(x) \rangle)}) - \frac{\beta}{2} \]
\[
\geq E_{(x,y) \sim P} (1 - \langle \hat{A}(S), y\phi(x) \rangle - \beta = 1 - \langle \hat{A}(S), \omega_P \rangle - \beta
\]

where the first line follows as \(- \log\) is a decreasing function and \(\| \phi(x) \| \leq 1\), and the second follows from lemma 2, which can be applied as by Cauchy-Schwarz

\[
|1 - \langle \hat{A}(S), y\phi(x) \rangle| \in [0, 2].
\]

By theorem 9, we have

\[
E_{S \sim P^n} 1 - \langle \hat{A}(S), \omega_P \rangle - \beta \leq E_{S \sim P^n} \left[ 1 - \langle \hat{A}(S), \omega_S \rangle + \frac{\| \hat{A}(S) - \omega_S \|^2}{\beta n} \right]
\]

with a corresponding high probability version.

Setting \(\omega_\pi = 0\), upper bounding \(\| \hat{A}(S) \| \leq 1\) yields

\[
E_{S \sim P^n} 1 - \langle \hat{A}(S), \omega_P \rangle \leq E_{S \sim P^n} \left[ 1 - \langle \hat{A}(S), \omega_S \rangle + \frac{1}{\beta n} \right] + \beta.
\]

Minimizing the right hand side of this bound over \(\hat{A}\) yields \(\hat{\omega}_S\), the mean classifier. Using the high probability bound and minimizing over \(\beta\) yields theorem 3.

### 8.3 PAC-Bayesian Bounds for Learning over Multiple Feature Maps

It is common for the learner to have access to several feature maps \(\phi_i : X \to \mathcal{H}_i\), for \(i \in\) a (possibly infinite) index set \(\mathcal{I}\). Define

\[
\mathcal{F}_\mathcal{I} = \cup_{i \in \mathcal{I}} \mathcal{F}_{\phi_i}.
\]

The disjoint union of the function classes \(\mathcal{F}_{\phi_i}\). Rather than priors and posteriors on a single \(\mathcal{F}_{\phi_i}\), we consider distributions on \(\mathcal{F}_\mathcal{I}\) that are infinite mixtures of normals,

\[
\mathcal{A}(S) = i \sim \alpha(S), \ x_i \sim \mathcal{N}(\bar{A}_i(S), 1) \quad \pi = i \sim \alpha_\pi, \ w_i \sim \mathcal{N}(\bar{w}_i, 1)
\]

where \(\pi_\omega, \mathcal{A}_i(S) \in \mathcal{H}_i\) and \(\alpha_\pi, \alpha(S) \in \mathcal{P}(\mathcal{I})\). These distributions first pick a tag \(i\) and then generate a weight vector \(\omega_i \in \mathcal{H}_i\).

**Theorem 9.** For all distributions \(P\), collections of feature maps \(\phi_i\), prior weights \(\alpha_\pi \in \mathcal{P}(\mathcal{I})\), prior vectors \(\omega_\pi \in \mathcal{H}_i\), sample dependent weights \(\alpha(S) \in \mathcal{P}(\mathcal{I})\), sample dependent weight vectors \(\bar{A}_i(S) \in \mathcal{H}_i\) and \(\beta > 0\) such that \(\| \phi(x) \| \leq 1 \forall x \) and \(\| \bar{A}_i(S) \| \leq 1 \forall S\)

\[
E_{S \sim P^n} E_{i \sim \alpha(S)} R_{\text{linear}} (P, \bar{A}_i(S)) 
\leq E_{S \sim P^n} \left[ E_{i \sim \alpha(S)} R_{\text{linear}} (S, \bar{A}_i(S)) + \frac{D_{KL}(\alpha(S), \alpha_\pi) + E_{i \sim \alpha(S)} \| \bar{A}_i(S) - \omega_\pi \|^2}{\beta n} \right] + \beta.
\]

Furthermore, with probability at least \(1 - \delta\) on a draw from \(S \sim P^n\) with \(\bar{A}_i, \omega_\pi\) and \(\beta\) fixed before the draw

\[
E_{i \sim \alpha(S)} R_{\text{linear}} (P, \bar{A}_i(S)) 
\leq E_{i \sim \alpha(S)} R_{\text{linear}} (S, \bar{A}_i(S)) + \frac{D_{KL}(\alpha(S), \alpha_\pi) + E_{i \sim \alpha(S)} \| \bar{A}_i(S) - \omega_\pi \|^2}{\beta n} + \beta.
\]
To recover theorem 4, take lemma 2, which can be applied as by Cauchy-Schwarz.

We can lower bound this quantity as follows

\[ E_{\omega \sim \mathcal{N}(\mathcal{A}(S), \omega_\pi)} e^{-\beta(1 - \langle \omega, y\phi(x) \rangle)} \geq E_{\omega \sim \mathcal{N}(\mathcal{A}(S), \omega_\pi)} (1 - \langle A(S), y\phi(x) \rangle) - \beta \]

where the first line follows from standard results on the moment generating function of normal distributions. We can lower bound this quantity as follows

\[-\frac{1}{\beta} \log(E_{(x,y) \sim p} E_{\omega \sim \mathcal{N}(A(S), \omega_\pi)} e^{-\beta(1 - \langle A(S), y\phi(x) \rangle)}) \geq -\frac{1}{\beta} \log(E_{(x,y) \sim p} E_{\omega \sim \mathcal{N}(A(S), \omega_\pi)} e^{-\beta(1 - \langle A(S), y\phi(x) \rangle)}) - \beta \frac{2}{\beta} \]

By theorem 7 we have

\[ E_{S \sim p^n} E_{i \sim \alpha(S)} 1 - \langle \bar{A}^i(S), \omega_P \rangle \leq E_{S \sim p^n} \left[ E_{i \sim \alpha(S)} 1 - \langle A^i(S), \omega_S \rangle + D_{KL}(\alpha(S), \alpha_P) \right] + \beta \frac{2}{\beta n} \]

with a corresponding high probability version.

To recover theorem 4, take \( \mathcal{I} = [1; k] \), a finite set of \( k \) kernels, \( \alpha_\pi \) the uniform distribution on \( \mathcal{I} \), \( \omega_\pi^i = 0 \) and upper bound \( ||A^i(S)|| \leq 1 \). Furthermore, restrict \( \alpha(S) \) to point mass distributions. We have

\[ E_{S \sim p^n} E_{i \sim \alpha(S)} 1 - \langle A^i(S), \omega_P \rangle \leq E_{S \sim p^n} \left[ E_{i \sim \alpha(S)} 1 - \langle A^i(S), \omega_S \rangle + \frac{1 + \log(k)}{\beta n} \right] + \beta. \]

Minimizing the right hand side of this bound over the \( A^i \) and \( \alpha \) yields the feature map that minimizes \( 1 - ||\omega^i_S|| \). Using the high probability bound and minimizing over \( \beta \) yields theorem 4.

### 8.4 Simpler Generalization Bounds

The PAC-Bayes theorems presented previously are certainly enough to assess the generalization of linear loss minimization. While the bounds are adaptive and could prove useful in future methods akin to those in [1], one can obtained useful bounds for the linear loss performance of \( \omega_S \) using only Hoeffding’s inequality and a union bound. The results here are specializations of those presented in [5] that hold for general similarity functions.
Theorem 10 (Hoeffding’s Inequality). Let $f : Z \to [0, 2]$. For all distributions $P \in \mathcal{P}(Z)$, with probability at least $1 - \delta$ on a draw $S \sim P^N$

$$\frac{1}{N} \sum_{z \in S} f(z) \leq \mathbb{E}_{z \sim P} f(z) + \sqrt{\frac{2 \log(\frac{1}{\delta})}{n}}$$

Proof. See appendix A.1 of [8].

Theorem 11. For all distributions $P$ and for all feature maps $\phi : X \to \mathcal{H}$, with probability at least $1 - \delta$ on a draw $S \sim P^N$

$$\mathcal{R}_{\text{linear}}(P, \omega_S) \leq \mathcal{R}_{\text{linear}}(P, \omega_P) + \sqrt{\frac{2 \log(\frac{1}{\delta})}{n}}$$

Proof. $\mathcal{R}_{\text{linear}}(P, \omega_S) = \frac{1}{N} \sum_{(x,y) \in S} 1 - \langle \omega_S, y\phi(x) \rangle$ which is the average of $N$ iid random variables each with mean $1 - \langle \omega_S, \omega_P \rangle$. Furthermore, by Cauchy-Schwarz, $|1 - \langle \omega_S, y\phi(x) \rangle| \leq 2$. Using Hoeffding’s bound yields the desired result.

Theorem 12. For all distributions $P$ and for all finite collections of feature maps $\phi_i : X \to \mathcal{H}_i$, $i \in [1; k]$, with probability at least $1 - \delta$ on a draw $S \sim P^N$

$$\mathcal{R}_{\text{linear}}(P, \omega^i_S) \leq \mathcal{R}_{\text{linear}}(P, \omega^i_P) + \sqrt{\frac{2 \left( \log(\frac{1}{\delta}) + \log(k) \right)}{n}}$$

for all $i \in [1; k]$.

Proof. Use Hoeffding’s bound for each kernel separately followed by a union bound over the $k$ kernels.

8.5 Proof of Theorem 6

Theorem. Let $f, \tilde{f}$ be classifiers such that $|f(x) - \tilde{f}(x)| \leq \epsilon$ for all $x$. Then for all distributions $P$,

$$\mathcal{R}_{01}(P, \tilde{f}) \leq \mathcal{R}_{\epsilon}(P, f).$$

Before the proof we prove the following simple lemma.

Lemma. Let $v, \tilde{v} \in \mathbb{R}$ with $|v - \tilde{v}| \leq \epsilon$. Then $\tilde{v} < 0$ implies $v < \epsilon$.

Proof. We have $v - \epsilon \leq \tilde{v} \leq v + \epsilon$. If $\tilde{v} < 0$, then $v - \epsilon < 0$.

We now prove the theorem.

Proof. By the conditions of the theorem, $|f(x) - \tilde{f}(x)| \leq \epsilon$, meaning $|yf(x) - y\tilde{f}(x)| \leq \epsilon$. By the previous lemma, $yf(x) < 0$ implies $yf(x) < \epsilon$. This means $[yf(x) < \epsilon] \leq [yf(x) < \epsilon]$. Averaging over $P$ yields the theorem.

\[ \square \]
8.6 Comparison with Makovoz’s Theorem

We call \( \omega \in \text{Hull}(S) \) \( m \)-sparse if it is a combination of only \( m \) elements of \( S \). Makovoz’s theorem is an existential result concerning the degree to which one can approximate any \( \omega \in \text{Hull}(S) \), with an \( m \)-sparse approximation \( \tilde{\omega}_m \). Let \( \{ B(z_i, \epsilon) \}_{i=1}^{N} \) be a collection of \( N \) balls in \( \mathcal{H} \). We say such a collection of balls covers \( S \) if \( S \subseteq \bigcup_{i=1}^{N} B(z_i, \epsilon) \). We call \( \epsilon \) the radius of the cover. Define the \( n \)th entropy number of \( S \) as

\[
\epsilon_m(S) := \inf \{ \epsilon : \exists \text{ a cover of } S \text{ with radius } \epsilon \text{ and } N \leq m \}.
\]

The entropy number of \( S \) is a fine grained means to assess its complexity. Intuitively, the simpler \( S \) is the faster \( \epsilon_m(S) \) decays as \( n \to \infty \). The following is theorem 27 in [19].

**Theorem 13.** Let \( \mathcal{H} \) be a Hilbert space of dimension \( d \). Then for all finite \( S \subseteq \mathcal{H} \), for all \( \omega \in \text{Hull}(S) \), and for all even \( m \leq |S| \) there exists an \( m \)-sparse \( \tilde{\omega} \in \text{Hull}(S) \) such that

\[
\| \omega - \tilde{\omega} \| \leq \frac{\sqrt{2} \epsilon_m(S)}{\sqrt{m}}.
\]

Theorem 13 has the advantage over equation 2 in that it includes more information about the sample than just the diameter of \( S \) and the distance from the sample mean to the boundary of \( S \) in the form of the entropy numbers of \( S \). It is known for \( S \) the \( d \)-dimensional unit ball, \( m^{-\frac{d}{2}} \leq \epsilon_m(S) \leq 4m^{-\frac{d}{2}} \) (see equation 1.1.10 of [7]). Naively, this means theorem 13 gives rates of convergence

\[
\| \omega - \tilde{\omega} \| \leq \frac{4\sqrt{2}}{m^{\frac{d}{2}}},
\]

where \( d \) can be replaced by \( |S| \) for infinite dimensional problems. This suggests that herding outperforms the bound in theorem 13. Ideally one wants a version of equation 2 that has direct reference to the entropy numbers of \( S \). This will be the subject of future work.
References

[1] Amiran Ambroladze, Emilio Parrado-Hernández, and John Shawe-Taylor. Tighter PAC-bayes bounds. Advances in neural information processing systems, 19:9, 2007.
[2] Dana Angluin and Philip Laird. Learning from noisy examples. Machine Learning, 2(4):343–370, 1988.
[3] F. Bach, S. Lacoste-Julien, and G. Obozinski. On the Equivalence between Herding and Conditional Gradient Algorithms. In Proceedings of the International Conference on Machine Learning (ICML), 2012.
[4] Francis R. Bach. Consistency of the group lasso and multiple kernel learning. The Journal of Machine Learning Research, 9:1179–1225, 2008.
[5] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. Machine Learning, 72(1-2):89–112, 2008.
[6] Amir Beck and Marc Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. Mathematical Methods of Operations Research, 59(2):235–247, 2004.
[7] Bernd Carl and Irina Stephani. Entropy, compactness, and the approximation of operators, volume 98. Cambridge University Press, 1990.
[8] Nicolò Cesa-Bianchi and Gábor Lugosi. Prediction, learning, and games. Cambridge University Press Cambridge, 2006.
[9] Yutian Chen, Max Welling, and Alexander J. Smola. Super Samples from Kernel Herding. In Uncertainty in Artificial Intelligence (UAI), 2010.
[10] Corinna Cortes, Marius Kloft, and Mehryar Mohri. Learning kernels using local Rademacher complexity. In Advances in Neural Information Processing Systems, pages 2760–2768, 2013.
[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.
[12] Andrew Cotter, Shai Shalev-Shwartz, and Nati Srebro. Learning optimally sparse support vector machines. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pages 266–274, 2013.
[13] Thomas M. Cover and Jay A. Thomas. Elements of Information Theory. Wiley, 2012.
[14] Luc Devroye, László Györfi, and Gábor Lugosi. A probabilistic theory of pattern recognition. Springer, 1996.
[15] Thor Graepel and Ralf Herbrich. A PAC-Bayesian margin bound for linear classifiers: Why SVMs work. In Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, volume 13, page 224. MIT Press, 2001.
[16] Thor Graepel, Ralf Herbrich, and Robert C. Williamson. PAC-Bayesian compression bounds on the prediction error of learning algorithms for classification. Machine Learning, 59(1-2):55–76, 2005.
[17] Thor Graepel, Ralf Herbrich, and Robert C. Williamson. From margin to sparsity. Advances in neural information processing systems, pages 210–216, 2001.
[18] Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel method for the two-sample-problem. In Advances in neural information processing systems, pages 513–520, 2006.
[19] Ralf Herbrich and Robert C. Williamson. Algorithmicluckiness. The Journal of Machine Learning Research, 3:175–212, 2003.
[20] Zakria Hussain and John Shawe-Taylor. Improved loss bounds for multiple kernel learning. In International Conference on Artificial Intelligence and Statistics, pages 370–377, 2011.
[21] Thorsten Joachims and Chun-Nam John Yu. Sparse kernel SVMs via cutting-plane training. Machine Learning, 76(2-3):179–193, 2009.
[22] Lee K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. The annals of Statistics, pages 608–613, 1992.
[23] S Sathiya Keerthi, Olivier Chapelle, and Dennis DeCoste. Building support vector machines with reduced classifier complexity. The Journal of Machine Learning Research, 7:1493–1515, 2006.
[24] Dung Duc Nguyen, Kazunori Matsumoto, Yasuhiro Takishima, and Kazuo Hashimoto. Condensed vector machines: learning fast machine for large data. Neural Networks, IEEE Transactions on, 21(12):1903–1914, 2010.
[25] Mark D. Reid and Robert C. Williamson. Information, divergence and risk for binary experiments. The Journal of Machine Learning Research, 12:731–817, 2011.
[26] B Schölkopf and A J Smola. Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond. Adaptive computation and machine learning series. University Press Group Limited, 2002.
[27] John Shawe-Taylor and John Langford. PAC-Bayes & margins. Advances in neural information processing systems, 15:439, 2003.
[28] Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf. Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions. In In Neural Information Processing Systems (NIPS) 2009, pages 1750–1758, 2009.
[29] Brendan van Rooyen, Aditya Krishna Menon, and Robert C. Williamson. Learning with Symmetric Label Noise: The Importance of Being Unhinged. arXiv preprint arXiv:1505.07634, 2015.
[30] Max Welling. Herding dynamical weights to learn. In Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009.
[31] Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. IEEE Transactions on Information Theory, 2006.