Active Meta Learning using MMD

Isak Falk*

November 25, 2019

Contents

1 Setup

We will follow the notation used in [1]. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ denote the data space, with $\mathcal{X} \subseteq \mathbb{R}^d, \mathcal{Y} \subseteq \mathbb{R}$ the input and output space respectively. We denote by z, x, y elements of the corresponding spaces where $z = (x, y)$ denotes an input/output pair. The base loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ measures the loss between two outputs y, y'. We will also write this in the form of $\ell(h, z) = \ell(h(x), y)$ where $h : \mathcal{X} \to \mathcal{Y}$. We represent the norm and inner product as $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ and unless specified, they denote the standard euclidean norm and scalar product. We write a matrix as $\mathbf{M}$ and the transpose as $\mathbf{M}^\top$. For a Hilbert space $\mathcal{H}$ we let $\mathcal{B}_1 \subseteq \mathcal{H}$ represent the zero-centred unit ball, the Hilbert space will be clear from the context. For any non-empty set S let $M_1^+(S)$ be the set of all probability measures on S. For any $k \in \mathbb{N}$ let $[\![k]\!] = \{1, \dots, k\}$.

Let ρ be a distribution over $M_1^+(\mathcal{Z})$, thus a sample $\mu \sim \rho$ is a distribution on $\mathcal{Z}$. We call ρ a *meta-distribution* and μ a *base-distribution*. An algorithm $\mathcal{A}$ is a function which maps from train sets to a hypothesis class $\mathcal{H}$, such that $\mathcal{A} : \mathcal{Z}^* \to \mathcal{H}, (z_i)_{i=1}^n \mapsto h = \mathcal{A}((z_i)_{i=1}^n)$. We sample m base-distributions $(\mu_i)_{i=1}^m \sim \rho^m$ iid and each of these base-distributions give rise to a data set $\mathcal{T}_i = (z_j^i)_{j=1}^n \sim \mu_i^n$ sampled iid of size n called a *task*, which is split into a train and validation set, $\mathcal{T}_i = D_i^{tr} \cup D_i^{val}$ of size n_{tr}, n_{val} respectively. For a data set $(x_i, y_i)_{i=1}^n$ we define the matrix and vector $\mathbf{X}_{ij} = x_{ij}$, $\mathbf{Y}_i = y_i$.

Given a set of tasks $M = (\mathcal{T}_i)_{i=1}^m$ we want to find an algorithm $\mathcal{A}$ that performs well on the *meta-risk*, also called the *transfer-risk*, defined as

$$\mathcal{E}_\rho(\mathcal{A}) = \mathbb{E}_{\mu \sim \rho} \mathbb{E}_{(D^{tr} \cup D^{val}) \sim \mu^n} \mathbb{E}_{D^{val}} \left[\ell(\mathcal{A}(D^{tr}), z) \right]. \tag{1}$$

We call the innermost expression the *meta-loss*,

$$L(\mathcal{A}, \mathcal{T}) = L(\mathcal{A}, D^{tr}, D^{val}) := \frac{1}{|D^{val}|} \sum_{z \in D^{val}} \ell(\mathcal{A}(D^{tr}), z). \tag{2}$$

so that we can express the meta-loss as

$$\mathcal{E}_\rho(\mathcal{A}) = \mathbb{E}_{\mu \sim \rho} \mathbb{E}_{\mathcal{T} \sim \mu^n} \left[L(\mathcal{A}, \mathcal{T}) \right]. \tag{3}$$

2 MMD bound

Inspired by the MMD bound of [2], we decompose the empirical counterpart to the meta-risk. Let M_t be any subset of tasks, $M_t \subseteq M$ and $|M_t| = t$, and let the empirical meta-risk for a set of tasks be $\mathcal{E}_M(\mathcal{A}) = \frac{1}{m} \sum_{\mathcal{T} \in M} L(\mathcal{A}, \mathcal{T})$ and similarly for M_t. This corresponds to the empirical risk in supervised learning.

*ucabitf@ucl.ac.uk

2.1 Bound on empirical excess risk

Given a class of functions $\mathcal{G} \subseteq \{f : 2^{\mathcal{Z}} \to \mathbb{R},\ f \text{ measurable}\}$ for any $g \in \mathcal{G}$ we can write

$$\begin{aligned}|\mathcal{E}_M(\mathcal{A}) - \mathcal{E}_{M_t}(\mathcal{A})| &= \left|\frac{1}{m}\sum_{i=1}^{m} L(\mathcal{A}, \mathcal{T}_i) - \frac{1}{t}\sum_{j=1}^{t} L(\mathcal{A}, \mathcal{T}_j)\right| \\ &\leq \left|\frac{1}{m}\sum_{i=1}^{m} L(\mathcal{A}, \mathcal{T}_i) - \frac{1}{m}\sum_{i=1}^{m} g(\mathcal{T}_i)\right| + \left|\frac{1}{m}\sum_{i=1}^{m} g(\mathcal{T}_i) - \frac{1}{t}\sum_{i=1}^{t} g(\mathcal{T}_j)\right| + \left|\frac{1}{t}\sum_{j=1}^{t} L(\mathcal{A}, \mathcal{T}_j) - \frac{1}{t}\sum_{i=1}^{t} g(\mathcal{T}_j)\right|.\end{aligned}$$

The middle expression can be controlled by assuming that $g \in \mathcal{B}_R \subseteq \mathcal{G}$, where $\mathcal{B}_R$ is the ball of radius R. We then have that

$$\left|\frac{1}{m}\sum_{i=1}^{m} g(\mathcal{T}_i) - \frac{1}{t}\sum_{i=1}^{t} g(\mathcal{T}_j)\right| \leq R \sup_{g \in \mathcal{B}_1} |\mathbb{E}_M[g(\mathcal{T})] - \mathbb{E}_{M_t}[g(\mathcal{T})]| = R \cdot \mathrm{IPM}_{\mathcal{B}_1}(M, M_t)$$

where $\mathrm{IPM}_{\mathcal{B}_1}(M, M_t)$ is the integral probability metric [3, 4] of the unit ball in $\mathcal{G}$. By choosing the space $\mathcal{G}$ we can recover many common distances over distributions, amongst others the Dudley metric, Wasserstein / Kantorovich metric and the maximum mean discrepancy [4]. We will here focus on the maximum mean discrepancy, which means that we will choose $\mathcal{G}$ to be an RKHS.

The other two expressions can be controlled by noticing that

$$\left|\frac{1}{m}\sum_{i=1}^{m} L(\mathcal{A}, \mathcal{T}_i) - \frac{1}{m}\sum_{i=1}^{m} g(\mathcal{T}_i)\right| \leq \max_{\mathcal{T} \in M} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})|$$

and since $M_t \subseteq M$,

$$\left|\frac{1}{t}\sum_{j=1}^{t} L(\mathcal{A}, \mathcal{T}_j) - \frac{1}{t}\sum_{i=1}^{t} g(\mathcal{T}_j)\right| \leq \max_{\mathcal{T} \in M_t} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})| \leq \max_{\mathcal{T} \in M} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})|.$$

which means that, optimising over $g \in \mathcal{B}_R$, we can bound the excess risk as

$$\begin{aligned}|\mathcal{E}_M(\mathcal{A}) - \mathcal{E}_{M_t}(\mathcal{A})| &\leq \inf_{g \in \mathcal{B}_R} \left|\frac{1}{m}\sum_{i=1}^{m} L(\mathcal{A}, \mathcal{T}_i) - \frac{1}{m}\sum_{i=1}^{m} g(\mathcal{T}_i)\right| + \left|\frac{1}{m}\sum_{i=1}^{m} g(\mathcal{T}_i) - \frac{1}{t}\sum_{i=1}^{t} g(\mathcal{T}_j)\right| + \left|\frac{1}{t}\sum_{j=1}^{t} L(\mathcal{A}, \mathcal{T}_j) - \frac{1}{t}\sum_{i=1}^{t} g(\mathcal{T}_j)\right| \\ &\leq R \cdot \mathrm{MMD}_{\mathcal{B}_1}(M, M_t) + 2 \inf_{g \in \mathcal{B}_R} \max_{\mathcal{T} \in M} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})|\end{aligned}$$

Now, if we let the kernel of $\mathcal{G}$ be $K(\cdot, \cdot) : 2^{\mathcal{Z}} \times 2^{\mathcal{Z}} \to \mathbb{R}_{\geq 0}$ which is a kernel on sequences of elements from $\mathcal{Z}$[1] which can also be seen as a kernel on empirical distributions or point clouds, see [5, Ch. 2] and denote the kernel mean embedding of a distribution ρ with respect to K as $\mathrm{KME}_K(\rho)$ we can further write this as

$$|\mathcal{E}_M(\mathcal{A}) - \mathcal{E}_{M_t}(\mathcal{A})| \leq R \cdot \|\mathrm{KME}_K(M) - \mathrm{KME}_K(M_t)\|_{\mathcal{G}} + 2 \inf_{g \in \mathcal{B}_R} \max_{\mathcal{T} \in M} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})|. \tag{4}$$

2.2 Choosing $\mathcal{G}$

The term $\mathrm{MMD}_{\mathcal{B}_1}(M, M_t)$ can be optimised by choosing what tasks to add to M_t from M in a greedily sequential manner using kernel herding [6] which is a special case of the frank-wolfe algorithm [7, 8] which yields a convergence of order $O(\frac{1}{t})$ compared to $O(\frac{1}{\sqrt{t}})$ for uniformly sampling the tasks. MMD is a pseudo-metric and if $\mathcal{G}$ is a characteristic RKHS, then it's also a metric. Under what conditions can we choose a characteristic $\mathcal{G}$ so that MMD is a metric but also $\inf_{g \in \mathcal{B}_R} \max_{\mathcal{T} \in M} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})|$ is small or zero?

Up until this point we have not put any restrictions on ℓ, $\mathcal{Z}$ or $\mathcal{A}$, and the inequality (4) holds in general. If we assume that for a fixed $\mathcal{A}$ the function $L(\mathcal{A}, \cdot) : 2^{\mathcal{Z}} \to \mathbb{R}_{\geq 0}$ is in $\mathcal{G}$, and we define $\kappa = \sup_{\mathcal{T}} \sqrt{K(\mathcal{T}, \mathcal{T})}$, then we can upper bound

$$\begin{aligned}\inf_{g \in \mathcal{B}_R} \max_{\mathcal{T} \in M} |L(\mathcal{A}, \mathcal{T}) - g(\mathcal{T})| &= \inf_{g \in \mathcal{B}_R} \max_{\mathcal{T} \in M} |\langle L(\mathcal{A}, \cdot) - g, \mathcal{T}\rangle| \\ &\leq \inf_{g \in \mathcal{B}_R} \max_{\mathcal{T} \in M} \|K(\mathcal{T}, \cdot)\|_{\mathcal{G}} \|L(\mathcal{A}, \cdot) - g\|_{\mathcal{G}} \\ &= \kappa \cdot \inf_{g \in \mathcal{B}_R} \|L(\mathcal{A}, \cdot) - g\|_{\mathcal{G}}\end{aligned}$$

[1] We allow duplicates hence we use sequences and not sets.

and if we denote $g^* := L(\mathcal{A}, \cdot)$ and $R^* = \|g^*\|_{\mathcal{G}}$ then we can write out $\inf_{g \in \mathcal{B}_R} \|L(\mathcal{A}, \cdot) - g\|_{\mathcal{G}}$ explicitly

$$\begin{aligned}
&= \inf_{g \in \mathcal{B}_R} \|g^* - g\|_{\mathcal{G}} \\
&= \left\| g^* - g^* \frac{R}{R^*} \right\|_{\mathcal{G}} \mathbb{1}_{\{R^* > R\}} \\
&= \left\| g^* (1 - \frac{R}{R^*}) \right\|_{\mathcal{G}} \mathbb{1}_{\{R^* > R\}} \\
&= (R^* - R) \mathbb{1}_{\{R^* > R\}} \\
&= \max(0, R^* - R)
\end{aligned}$$

where we have used the fact that the closest element to g^* is g^* projected onto $\mathcal{B}_R$. If the above is true ($L(\mathcal{A}, \cdot) \in \mathcal{G}$) then (4) can be written as

$$|\mathcal{E}_M(\mathcal{A}) - \mathcal{E}_{M_t}(\mathcal{A})| \leq R \cdot \|\mathrm{KME}_K(M) - \mathrm{KME}_K(M_t)\|_{\mathcal{G}} + 2\kappa \max(0, R^* - R) \tag{5}$$

$$= R \left(\|\mathrm{KME}_K(M) - \mathrm{KME}_K(M_t)\|_{\mathcal{G}} + 2\kappa \max(0, \frac{R^*}{R} - 1) \right) \tag{6}$$

2.3 Functional class of $L(\mathcal{A}, \cdot)$

Given a class of algorithms, each algorithm $\mathcal{A}$ yields a different function $L_{\mathcal{A}}(\mathcal{T})$ which can be expanded as

$$\begin{aligned}
L_{\mathcal{A}}(\mathcal{T}) &= \frac{1}{|D^{val}|} \sum_{z \in D^{val}} \ell(\mathcal{A}(D^{tr}), z) \\
&= \frac{1}{|D^{val}|} \sum_{(x,y) \in D^{val}} \ell(\mathcal{A}(D^{tr})(x), y)
\end{aligned}$$

and we see that the smoothness of $L_{\mathcal{A}}$ depends on both D^{tr} and D^{val}, where D^{tr} enters through the output of the training, $\mathcal{A}(D^{tr})$, changes with respect to D^{tr} and D^{val} through $\ell(\mathcal{A}(D^{tr})(x), y)$. So if ℓ is well-behaved[2] in both of its arguments and $\mathcal{A}(D^{tr})(x)$ is smooth with respect to both D^{tr}, x then the meta-loss will be well-behaved as well.

Below we show different conditions when different classes of algorithms give smooth functions, which means that we can approximate $L_{\mathcal{A}}$ well by an element in some RKHS $\mathcal{G}$.

2.3.1 Kernel Ridge Regression

We first consider the class of functions from ERM leading to Kernel Ridge Regression (KRR), which means that we set $\ell(y, y') = (y - y')^2$ and let $\mathcal{H}$ be the corresponding RKHS (the hypothesis space) with kernel $l(\cdot, \cdot)$. In this case we have that the algorithm $\mathcal{A}_{\lambda, h_0}$ is given by

$$\mathcal{A}_{\lambda, h_0}(D^{tr}) = \arg\min_{h \in \mathcal{H}} \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} (h(x_i) - y_i)^2 + \lambda \|h - h_0\|_{\mathcal{H}}^2$$

and normal KRR is recovered when we set $h_0 = 0$. If we parameterise $h_0 = \mathrm{span}(\{\psi_p\}_{p=1}^P)$ where $\{\psi_p\}_{p=1}^P \subseteq \mathcal{H}^P$ then the semi-parametric representer theorem [9] says that the unique minimiser (due to the loss and regularisation terms both being convex and increasing) is of the form $h(x) = \sum_{i=1}^{n_{tr}} \alpha_i l(x_i, x) + \sum_{p=1}^P \beta_p \psi_p(x) = f(x) + b(x)$ where h minimizes the problem

$$J(h) = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} (h(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

If we now let $\mathbf{\Psi}_{ip} = \psi_p(x_i)$ and $\mathbf{L}_{ij} = l(x_i, x_j)$ we can rewrite this in a dual form depending only on the vectors of coefficients $\alpha \in \mathbb{R}^{n_{tr}}, \beta \in \mathbb{R}^P$. The dual problem can be shown to be equal to

$$J(\alpha, \beta) = \frac{1}{n_{tr}} \|\mathbf{K}\alpha + \mathbf{\Psi}\beta - \mathbf{Y}\|_{\mathbb{R}^{n_{tr}}}^2 + \lambda \alpha^\top \mathbf{K} \alpha \tag{7}$$

[2] Well-behaved in the sense that for different assumptions on $\ell, \mathcal{A}$ we will get different behaviour of $L_{\mathcal{A}}$ e.g. $L_{\mathcal{A}} \in C^k$ for some $k \in \mathbb{N} \cup \{+\infty\}$.

which if we define the following, $\theta = [\alpha, \beta]^\top \in \mathbb{R}^{n_{tr}+P}$, $\mathbf{L} = [\mathbf{K}, \mathbf{\Psi}] \in \mathbb{R}^{n_{tr}\times(n_{tr}+P)}$ and

$$\mathbf{R} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(n+P)\times(n+P)}$$

then we can rewrite (7) as follows

$$J(\theta) = \frac{1}{n}\|\mathbf{L}\theta - \mathbf{Y}\|^2_{\mathbb{R}^n} + \lambda\theta^\top \mathbf{R}\theta.$$

Jacobian and Hessian looks as follows

$$\nabla_\theta J = \frac{2}{n}(\mathbf{L}^\top\mathbf{L}\theta - \mathbf{L}^\top\mathbf{Y} + n\lambda\mathbf{R}\theta) = \frac{2}{n}(\mathbf{L}^\top(\mathbf{L}\theta - \mathbf{Y}) + n\lambda\mathbf{R}\theta) = \frac{2}{n}((\mathbf{L}^\top\mathbf{L} + n\lambda\mathbf{R})\theta - \mathbf{L}^\top\mathbf{Y})$$
$$\nabla^2_\theta J = \frac{2}{n}(\mathbf{L}^\top\mathbf{L} + n\lambda\mathbf{R})$$

and the solution is

$$\begin{aligned}\theta^* &= \left(\mathbf{L}^\top\mathbf{L} + n\lambda\mathbf{R}\right)^{-1}\mathbf{L}^\top\mathbf{Y} \\ &= \left(\begin{bmatrix} \mathbf{K}^2 & \mathbf{K}\mathbf{\Psi} \\ \mathbf{\Psi}^\top\mathbf{K} & \mathbf{\Psi}^\top\mathbf{\Psi} \end{bmatrix} + n\lambda\begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right)^{-1}\begin{bmatrix} \mathbf{K}\mathbf{Y} \\ \mathbf{\Psi}^\top\mathbf{Y} \end{bmatrix}\end{aligned}$$

where we require the Hessian to be p.d. Letting $P = 0$ we recover the usual KRR solution

$$\mathcal{A}_\lambda(D^{tr})(x) = \sum_{i=1}^{n_{tr}} \alpha_i^* K(x_i, x) = \mathbf{K}_x^\top\alpha^* = \mathbf{K}_x^\top(\mathbf{K} + n_{tr}\lambda I_{n_{tr}})^{-1}\mathbf{Y}.$$

and for a general set $\{\psi_p\}_{p=1}^P$ where we let $\psi(x) = [\psi_1(x), \dots, \psi_P(x)]^\top$ and let $\mathcal{A}_{\lambda,\{\psi_p\}_{p=1}^P}$ denote the algorithm with regularisation parameter λ and h_0 in the span of $\{\psi_p\}_{p=1}^P$, then

$$\mathcal{A}_{\lambda,\{\psi_p\}_{p=1}^P}(D^{tr})(x) = \mathbf{K}_x^\top\alpha^* + \psi(x)^\top\beta^* = \begin{bmatrix} \mathbf{K}_x \\ \psi(x) \end{bmatrix}^\top\theta^*.$$

In this case we can see that when the Hessian is assumed to the p.d. and the kernel is smooth that the algorithm as a function of the train set is smooth since the inverse and matrix multiplication are smooth functions and the estimator is also smooth in x since the kernel is smooth and the inner product is smooth. This shows that for KRR, the function $L_{\mathcal{A}_{\lambda,\{\psi\}_{p=1}^P}}(\mathcal{T})$ is smooth given that $\{\psi\}_{p=1}^P$ leads to a p.d. Hessian and for a suitable RKHS $\mathcal{G}$ dense in the set of smooth functions from $2^{\mathcal{Z}} \to \mathbb{R}$ (6) holds.

2.3.2 Gradient Descent

Assuming that we are doing ERM where the algorithm $\mathcal{A}$ is defined to be the solution to the ERM problem with some pre-specified hypothesis space $\mathcal{H}$,

$$\mathcal{A}(D^{tr}) = \arg\min_{h\in\mathcal{H}} \mathcal{E}_{D^{tr}}(h)$$

if we instead of solving this analytically do K-step gradient descent, KGD, where we parameterise h by θ given some initial starting point θ_0 and a learning rate scheme $(\gamma_k)_{k=1}^\infty$ then we can define the GD-update operator as

$$GD^\gamma(\theta) = \theta - \gamma\nabla_{\tilde{\theta}}\mathcal{E}_{D^{tr}}(\tilde{\theta})|_{\tilde{\theta}=\theta}. \tag{8}$$

From this we can define a new algorithm $\mathcal{A}_{KGD}(D^{tr})$ which depends implicitly on the initialisation point and learning rate $\theta_0, (\gamma_k)_{k=1}^\infty$ as

$$\mathcal{A}_{KGD}(D^{tr}) = GD^{\gamma_K} \circ \cdots \circ GD^{\gamma_1}(\theta_0) \tag{9}$$

and if we write $\theta_k = GD^{\gamma_k}(\theta_{k-1})$ and let $\mathbf{g}_k = \nabla_{\tilde{\theta}}\mathcal{E}_{D^{tr}}(\tilde{\theta})|_{\tilde{\theta}=\theta_k}$ then we can equivalently express this as

$$\mathcal{A}_{KGD}(D^{tr}) = \theta_0 - \sum_{k=0}^{K-1}\gamma_{k+1}\mathbf{g}_k. \tag{10}$$

Since sums of smooth functions are smooth we have that $\mathcal{A}_{KGD}(D^{tr})$ is a smooth function of D^{tr} as long as the gradients, $\mathbf{g}_k$ are. This can be made precise, but we can avoid it by assuming that the loss and functions in the hypothesis class is infinitely differentiable. This means that the function $L_{\mathcal{A}}(\mathcal{T})$ is smooth in this case.

2.4 Equivalence of Curriculum Learning and Active Learning

The formal definition of a *curriculum* according to [10] is as follows, let $P(z)$ be the target distribution and let $Q_\lambda(z)$ be a set of distributions indexed by the parameter $\lambda \in [0, 1]$, where $\lambda = 0$ is a simpler toy problem and $\lambda = 1$ is the original problem, $Q_1 = P$. Let $0 \leq W_\lambda(z) \leq 1$. Then $Q_\lambda(z) \propto W_\lambda(z)P(z)$ and $\int Q_\lambda(z)\mathrm{d}P(z) = 1$. For a monotonically increasing sequence of values $(\lambda_l)_{l=1}^L$ with $\lambda_1 = 0, \lambda_L = 1$, the sequence of distribution $(Q_{\lambda_l})_{l=1}^L$ is a curriculum if the entropy of the sequence of distribution is increasing, $H(Q_{\lambda_l}) < H(Q_{\lambda_{l+1}})$ and $(W_\lambda)_{l=1}^L$ is non-decreasing for all z, $W_{\lambda_l}(z) \leq W_{\lambda_{l+1}}(z)$. To harmonise with our notation, we let $t := l$ and $L := n$.

For active learning, we simply let $Q_t := Q_{\lambda_t}$ which is an empirical distribution of size t, thus $W_t(z) = \mathbb{1}_{\{z \in Q_t\}}$ and we have that the sequence $(W_t)_{l=1}^n$ is monotonically increasing, since $Q_t \subseteq Q_{t+1}$ and $H(Q_t) = -\sum_{i=1}^t t^{-1} \log\left(t^{-1}\right) = \log(t)$ which is increasing in t. Hence as long as we use uniform weight over the active learning set, active learning with uniform weights is a curriculum (which extends to any setting, supervised or meta learning).

This means that the analysis done for active meta learning and the MMD bound in section 2 applies to curriculum learning problems as well.

References

[1] Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Learning to learn around a common mean. In *Advances in Neural Information Processing Systems*, pages 10169–10179, 2018.

[2] Tom J. Viering, Jesse H. Krijthe, and Marco Loog. Nuclear discrepancy for active learning, 2017.

[3] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

[4] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

[5] Dougal J Sutherland. Scalable, flexible and active learning on distributions. Technical report, Carnegie Mellon University Pittsburgh United States, 2016.

[6] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012.

[7] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[8] Francis Bach, Simon Lacoste-Julien, and Guillaume Obozinski. On the equivalence between herding and conditional gradient algorithms. *arXiv preprint arXiv:1203.4523*, 2012.

[9] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. *Computational Learning Theory*, pages 416–426, 2001.

[10] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.