Efficient and principled score estimation with Nyström kernel exponential families

Danica J. Sutherland* **Heiko Strathmann*** **Michael Arbel** **Arthur Gretton**
Gatsby Computational Neuroscience Unit, University College London
{danica.j.sutherland,heiko.strathmann,michael.n.arbel,arthur.gretton}@gmail.com

Abstract

We propose a fast method with statistical guarantees for learning an exponential family density model where the natural parameter is in a reproducing kernel Hilbert space, and may be infinite-dimensional. The model is learned by fitting the derivative of the log density, the *score*, thus avoiding the need to compute a normalization constant. Our approach improves the computational efficiency of an earlier solution by using a low-rank, Nyström-like solution. The new solution retains the consistency and convergence rates of the full-rank solution (exactly in Fisher distance, and nearly in other distances), with guarantees on the degree of cost and storage reduction. We evaluate the method in experiments on density estimation and in the construction of an adaptive Hamiltonian Monte Carlo sampler. Compared to an existing score learning approach using a denoising autoencoder, our estimator is empirically more data-efficient when estimating the score, runs faster, and has fewer parameters (which can be tuned in a principled and interpretable way), in addition to providing statistical guarantees.

1 INTRODUCTION

We address the problem of efficiently estimating the natural parameter of a density in the exponential family, where this parameter may be infinite-dimensional (a member of a function space). While finite-dimensional exponential families are a keystone of parametric statistics (Brown 1986), their generalization to the fully non-parametric setting has proved challenging, despite the benefits and applications envisaged for such models (Canu and Smola 2006): it is difficult to construct a practical, consistent maximum likelihood solution for infinite-dimensional natural parameters (Barron and Sheu 1991; Gu and Qiu 1993; Fukumizu 2009). In the absence of a tractable estimation procedure, the infinite exponential family has not seen the widespread adoption and practical successes of other nonparametric generalizations of parametric models, for instance the Gaussian and Dirichlet processes.

Recently, Sriperumbudur et al. (2017) developed a procedure to fit infinite exponential family models to sample points drawn i.i.d. from a probability density, where the natural parameter is a member of a reproducing kernel Hilbert space. The approach employs a score matching procedure (Hyvärinen 2005), which minimizes the *Fisher distance*: the expected squared distance between the model *score* (derivative of the log density) and the score of the (unknown) true density, which can be evaluated using integration by parts. Unlike the maximum likelihood case, a Tikhonov-regularized solution can be formulated to obtain a well-posed and straightforward solution, which is a linear system defined in terms of first and second derivatives of the RKHS kernels at the sample points. Details of the model and its empirical fit are given in Section 2. Sriperumbudur et al. (2017) established consistency in Fisher, L^r, Hellinger, and KL distances, with rates depending on the smoothness of the density.

Strathmann et al. (2015) used the infinite-dimensional exponential family to approximate Hamiltonian Markov chain Monte Carlo when gradients are unavailable. In this setting, the score of the stationary distribution of the Markov chain is learned from the chain history, and used in formulating new, more efficient proposals for a Metropolis-Hastings algorithm. Computing the full solution from Sriperumbudur et al. (2017) has memory cost $\mathcal{O}(n^2d^2)$ and computational cost $\mathcal{O}(n^3d^3)$, where n is the number of training samples and d is the dimen-

Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

*These authors contributed equally.

sion of the problem; thus approximations were needed for practical implementation. Strathmann et al. proposed two heuristics: one using random Fourier features (Rahimi and Recht 2007; Sutherland and Schneider 2015; Sriperumbudur and Szábo 2015), and the second using a finite, random set of basis points. While these heuristics greatly improved the runtime, no convergence guarantees are known, nor how quickly to increase the complexity of these solutions with increasing n.

We present an efficient learning scheme for the infinite-dimensional exponential family, using a Nyström approximation to the solution established in Theorem 1. Our main theoretical contribution, in Theorem 2, is to prove guarantees on the convergence of this algorithm for an increasing number m of Nyström points and n training samples. Depending on the problem difficulty, convergence is attained in the regime $m \sim n^{1/3} \log n$ to $m \sim n^{1/2} \log n$, thus yielding guaranteed cost savings. The overall Fisher distance between our solution and the true density decreases as $m, n \to \infty$ with rates that match those of the full solution from Sriperumbudur et al. (2017, Theorem 6); convergence in other distances (e.g., KL and Hellinger) either matches or is slightly worse, depending on the problem smoothness. These tight generalization bounds draw on recent state-of-the-art techniques developed for least-squares regression by Rudi et al. (2015), which efficiently and directly control the generalization error as a function of the Nyström basis, rather than relying on indirect proofs via the reconstruction error of the Gram matrix, as in e.g. Cortes et al. (2010). Sections 3 and 4 give details.

In our experiments (Section 5), we compare our approach against the full solution of Sriperumbudur et al. (2017), the heuristics of Strathmann et al. (2015), and the autoencoder score estimator of Alain and Bengio (2014) (discussed in Section 2.3). We address two problem settings. First, we evaluate score function estimation for known, multimodal densities in high dimensions. Second, we consider adaptive Hamiltonian Monte Carlo in the style of Strathmann et al. (2015), where the score is used to propose Metropolis-Hastings moves; these will be accepted more often as the quality of the learned score improves. Our approach is more accurate, faster, and easier to tune than the autoencoder score estimate. Moreover, our method performs as well as the full kernel exponential family solution at a much lower computational cost, and on par with previous heuristic approximations.

2 UNNORMALIZED DENSITY AND SCORE ESTIMATION

Suppose we are given a set of points $X = \{X_b\}_{b \in [n]} \subset \mathbb{R}^d$ sampled i.i.d. from an unknown distribution with density p_0. Our setting is that of *unnormalized density estimation*: we wish to fit a model $p(\cdot) = p'(\cdot)/Z(p')$ such that $p \approx p_0$ in some sense, but without concerning ourselves with the partition function $Z(p')$, which normalizes p' such that $\int p(x)\,\mathrm{d}x = 1$. In many powerful classes of probabilistic models, computing the partition function is intractable, but several interesting applications do not require it, including mode finding and sampling via Markov Chain Monte Carlo (MCMC). This setting is closely related to that of *energy-based learning* (LeCun et al. 2006).

Exponential family models with infinite-dimensional natural parameters are a particular case for which the partition function is problematic. Here fitting by maximum likelihood is difficult, and becomes completely impractical in high dimensions (Barron and Sheu 1991; Gu and Qiu 1993; Fukumizu 2009).

Hyvärinen (2005) proposed an elegant approach to estimate an *unnormalized* density, by minimizing the Fisher divergence, the expected squared distance between *score functions*[1] $\nabla_x \log p(x)$. The divergence $J(p_0 \| p)$ is given by

$$\frac{1}{2} \int p_0(x) \left\| \nabla_x \log p(x) - \nabla_x \log p_0(x) \right\|_2^2 \mathrm{d}x, \qquad (1)$$

which under some mild regularity conditions is equal to a constant (depending only on p_0) plus

$$\int p_0(x) \sum_{i=1}^{d} \left[\partial_i^2 \log p(x) + \frac{1}{2} \left(\partial_i \log p(x) \right)^2 \right] \mathrm{d}x. \qquad (2)$$

We use $\partial_i f(x)$ to mean $\frac{\partial}{\partial x_i} f(x)$. Crucially, (2) is independent of the normalizer Z and, other than the constant, depends on p_0 only through an expectation, so it can be estimated by a simple Monte Carlo average.

The score function is in itself a quantity of interest, and is employed directly in several algorithms. Perhaps best known is Hamiltonian Monte Carlo (HMC; e.g. Neal 2011), where the score is used in constructing Hamiltonian dynamics that yield fast mixing chains. Thus, if the score can be learned from the chain history, it can be used in constructing an approximate HMC sampler with mixing properties close to those attainable using the population score (Strathmann et al. 2015). Another application area is in constructing control functionals for Monte Carlo integration (Oates et al. 2017): again, learned score functions could be used where closed-form expressions do not exist.

Computing unnormalized densities from a nonparametrically learned score function can be a more challenging

[1] Here we use *score* in the sense of Hyvärinen (2005); in traditional statistical parlance, this is the score with respect to a hypothetical location parameter of the model.

task. Numerical integration of the score estimate can lead to accumulating errors; moreover, as discussed by Alain and Bengio (2014, Section 3.6), a given score estimate might not correspond to a valid gradient function, or might not yield a normalizable density. The exponential family model does not suffer these drawbacks, as we will see next.

2.1 Kernel exponential families

We now describe the *kernel exponential family* $\mathcal{P}$ (Canu and Smola 2006; Fukumizu 2009), and how to perform unnormalized density estimation within it. $\mathcal{P}$ is an infinite-dimensional exponential family:

$$\mathcal{P} = \{p_f(x) := \exp\left(f(x) - A(f)\right) q_0(x) \mid f \in \mathcal{F}\},$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space (Berlinet and Thomas-Agnan 2004), $\mathcal{F} \subseteq \mathcal{H}$ is the set of functions for which $A(f) = \log \int \exp(f(x))\, q_0(x)\, \mathrm{d}x$, the log-partition function, is finite, and q_0 is a base measure with appropriately vanishing tails. That this is a member of the exponential family becomes apparent when we recall the reproducing property $f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}}$: the feature map $x \mapsto k(x, \cdot)$ is the sufficient statistic, and f is the natural parameter.

Example 1 of Sriperumbudur et al. (2017) shows how various standard finite-dimensional members of the exponential family, including Gamma, Poisson, Binomial and so on, fit into this framework with particular kernel functions. When $\mathcal{H}$ is infinite-dimensional, $\mathcal{P}$ can be very rich: for instance, when the kernel on $\mathbb{R}^d$ is a continuous function vanishing at infinity and integrally strictly positive definite, then $\mathcal{P}$ is dense in the family of continuous densities vanishing at infinity for which $\|p/q_0\|_\infty$ is bounded, with respect to the KL, TV, and Hellinger divergences (Sriperumbudur et al. 2017, Corollary 2).

As discussed earlier, maximum likelihood estimation is difficult due to the intractability of $A(f)$. Instead, Sriperumbudur et al. (2017) propose to use a score-matching approach to find an f such that p_f approximates p_0. Their empirical estimator of (2) is

$$\hat{J}(f) = \frac{1}{n}\sum_{b=1}^{n}\sum_{i=1}^{d} \partial_i^2 f(X_b) + \frac{1}{2}\left(\partial_i f(X_b)\right)^2; \tag{3}$$

this additionally drops an additive constant from (2) that depends on p_0 and q_0 but not f. The regularized loss $\hat{J}(f) + \frac{1}{2}\lambda\|f\|_{\mathcal{H}}^2$ is minimized over $f \in \mathcal{H}$ by

$$f_{\lambda,n} = -\frac{\hat{\xi}}{\lambda} + \sum_{a=1}^{n}\sum_{i=1}^{d} \beta_{(a,i)}\partial_i k(X_a, \cdot), \tag{4}$$

$$\hat{\xi} = \frac{1}{n}\sum_{a=1}^{n}\sum_{i=1}^{d} \partial_i^2 k(X_a, \cdot) + \partial_i k(X_a, \cdot)\partial_i \log q_0(X_a),$$

where $\beta_{(a,i)}$ denotes the $(a-1)d+i$th entry of a vector $\beta \in \mathbb{R}^{nd}$. We use $\partial_i k(x, y)$ to mean $\frac{\partial}{\partial x_i} k(x, y)$, and $\partial_{i+d} k(x, y)$ for $\frac{\partial}{\partial y_i} k(x, y)$. To evaluate the estimated unnormalized log-density $f_{\lambda,n}$ at a point x, we take a linear combination of $\partial_i k(X_a, x)$ and $\partial_i^2 k(X_a, x)$ for each sample X_a. The weights β in (4) are obtained by solving the nd-dimensional linear system

$$(G + n\lambda I)\beta = h/\lambda, \tag{5}$$

where $G \in \mathbb{R}^{nd\times nd}$ is the matrix collecting partial derivatives of the kernel at the training points, $G_{(a,i),(b,j)} = \partial_i\partial_{j+d} k(X_a, X_b)$, and $h \in \mathbb{R}^{nd}$ evaluates derivatives of $\hat{\xi}$, $h_{(b,i)} = \partial_i \hat{\xi}(X_b)$.

Solving (5) takes $\mathcal{O}(n^3 d^3)$ time and $\mathcal{O}(n^2 d^2)$ memory, which quickly becomes infeasible as n grows, especially for large d. We will propose a more scalable approximation in Section 3.

2.2 Fast approximate kernel regression

The system of (5) is related to the problem of kernel ridge regression, which suffers from similar $\mathcal{O}(n^3)$ computational cost. Thus we will briefly review methods for speeding up kernel regression.

Nyström methods We refer here to a class of broadly related Nyström-type methods (Williams and Seeger 2000; Smola and Schölkopf 2000; Rudi et al. 2015). The representer theorem (Schölkopf et al. 2001) guarantees that the minimizer of the empirical regression loss for a training set $X = \{X_b\}_{b\in[n]}$ over the RKHS $\mathcal{H}$ with kernel k will lie in the subspace $\mathcal{H}_X = \operatorname{span}\{k(X_b, \cdot)\}_{b\in[n]}$. Nyström methods find an approximate solution by optimizing over a smaller subspace $\mathcal{H}_Y$, usually given by $\mathcal{H}_Y = \operatorname{span}\{k(y, \cdot)\}_{y\in Y}$ for a set of m points $Y \subseteq X$ chosen uniformly at random. This decreases the computational burden both of training ($\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$ time, $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$ memory) and testing ($\mathcal{O}(n)$ to $\mathcal{O}(m)$ time and memory).

Guarantees on the performance of Nyström methods have been the topic of considerable study. Earlier approaches have worked by first bounding the error in a Nyström approximation of the kernel matrix on the sample (Drineas and Mahoney 2005), and then separately evaluating the impact of regression with an approximate kernel matrix (Cortes et al. 2010). This approach, however, results in suboptimal rates; better rates can be obtained by considering the whole problem at once (El Alaoui and Mahoney 2015), including its direct impact on generalization error (Rudi et al. 2015).

Random feature approximations Another popular method for scaling up kernel methods is to use random Fourier features (Rahimi and Recht 2007; Sutherland and Schneider 2015; Sriperumbudur and Szábo

2015) and their variants. Rather than finding the best solution in a subspace of $\mathcal{H}$, these methods choose a set of parametric features, often independent of the data, such that expected inner products between the features coincide with the kernel. These methods have some attractive computational properties but generally also require the number of features to increase with the data size in a way that can be difficult to analyze: see Rudi and Rosasco (2017) for such an analysis in regression.

Sketching Another scheme for improving the speed of kernel ridge regression, sketching (Yang et al. 2017; Woodruff 2014) compresses the kernel matrix and the labels by multiplying with a sketching matrix. These methods have some overlap with Nyström-type approaches, and our method will encompass certain classes of sketches (Rudi et al. 2015, Appendix C.1).

2.3 Prior methods for direct score estimation

Alain and Bengio (2014) proposed a deep learning-based approach to directly learn a score function from samples. Denoising autoencoders are networks trained to recover the original inputs from versions with noise added. A denoising autoencoder trained with L_2 loss and noise $\mathcal{N}(0, \sigma^2 I)$ can be used to construct a score estimator: $(r_\sigma(x) - x)/\sigma^2 \approx \nabla_x \log p_0(x)$, where r_σ is the autoencoder's reconstruction function. When the autoencoder has infinite capacity and reaches its global optimum, Alain and Bengio (2014) show that this estimator is consistent as $\sigma \to 0$. For realistic autoencoders with finite representation capacity, however, the consistency of this approach remains an open question. Moreover, this technique has many hyper-parameters to choose, both in the architecture of the network and in its trained, with no theory yet available to guide those choices.

3 NYSTRÖM METHODS FOR ESTIMATION IN KERNEL EXPONENTIAL FAMILIES

To alleviate the computational costs of the linear system in (5), we apply the Nyström idea to the estimator of the full kernel exponential family model in (4). More precisely, we select a set of m "basis" points $Y = \{Y_a\}_{a\in[m]}$, and restrict the optimization in (4) to

$$\mathcal{H}_Y := \operatorname{span}\left\{\partial_i k(Y_a, \cdot)\right\}_{\substack{a\in[m]}}^{i\in[d]}, \tag{6}$$

which is a subspace of $\mathcal{H}$ with elements that can be represented using md coefficients, similar to (4). Typically $Y \subset X$; in particular, Y is usually chosen as a uniformly random subset of X. We could, however, use any set of points Y different from X, or even a different set of spanning vectors than $\partial_i k(Y_a, \cdot)$.

Dimension subsampling A further reduction of the computational load can be achieved by only using certain components, $\mathcal{I} \subset [n] \times [d]$ with $|\mathcal{I}| \leq md$, of the basis points Y. Thus (4) is optimized over

$$\operatorname{span}\left\{\partial_i k(Y_a, \cdot) \mid (a, i) \in \mathcal{I}\right\}. \tag{7}$$

In this case, each double sum over all basis points' components $\sum_{a=1}^{m}\sum_{i=1}^{d}$, such as in (4), would be replaced by $\sum_{(a,i)\in\mathcal{I}}$. Our theoretical framework will support choosing whether or not to include each of the md component according to user-specified probability ρ, such that the expected number of components $|\mathcal{I}|$ is ρmd, or of choosing exactly $\ell \leq d$ components from each of m points. For the sake of notational simplicity, we will give all results in the main body for the case of using all components as in (6), i.e. $|\mathcal{I}| = md$, commenting on implications for subsampling across dimensions where appropriate. We will explore the practical impact of subsampling in the experiments.

Theorem 1. *The regularized minimizer of the empirical Fisher divergence (3) over $\mathcal{H}_Y$ (6) is*

$$f^m_{\lambda,n} = \sum_{a=1}^{m}\sum_{i=1}^{d}(\beta_Y)_{(a,i)}\partial_i k(Y_b, \cdot),$$
$$\beta_Y = -(\tfrac{1}{n}B_{XY}^\intercal B_{XY} + \lambda G_{YY})^\dagger h_Y. \tag{8}$$

Here † denotes the pseudo-inverse, and $B_{XY} \in \mathbb{R}^{nd\times md}, G_{YY} \in \mathbb{R}^{md\times md}$, $h_Y \in \mathbb{R}^{md}$ are given by

$$(B_{XY})_{(b,i),(a,j)} = \partial_i\partial_{j+d}k(X_b, Y_a)$$
$$(G_{YY})_{(a,i),(a',j)} = \partial_i\partial_{j+d}k(Y_a, Y_{a'})$$
$$(h_Y)_{(a,i)} = \frac{1}{n}\sum_{b=1}^{n}\sum_{j=1}^{d}\partial_i\partial^2_{j+d}k(Y_a, X_b) + \partial_i\partial_{j+d}k(Y_a, X_b)\partial_j \log q_0(X_b).$$

The proof, which is similar to the kernel ridge regression analogue (Rudi et al. 2015), is given in Appendix B. In fact, we show a slight generalization (Lemma 4), which also applies to more general subspaces $\mathcal{H}_Y$.

It is worth emphasizing that in order to evaluate an estimate $f^m_{\lambda,n}$, we need only evaluate derivatives of the kernel between the basis points Y and the test point x. We no longer need X at all: its full contribution is summarized in β_Y. The same is true when subsampling across dimensions, but we need to keep all points with any used components; we will come back to this in the experiments.

When $Y \subseteq X$, the above quantities are simply block-subsampled versions of the terms in the full solution

(5). When using dimension subsampling with $|\mathcal{I}| < md$, we subsample further within the blocks. Note, however, that when $Y = X$ we do not exactly recover the solution (5), because $\hat{\xi}$ contains components of the form $\partial_i^2 k(X_b, \cdot) \notin \mathcal{H}_Y$ even when $Y = X$.

Computing the $md \times md$ matrix in (8) takes $\mathcal{O}(nmd^2)$ memory and $\mathcal{O}(nm^2d^3)$ time, both linear in n. Computing the pseudo-inverse takes $\mathcal{O}(m^3d^3)$ computation, independent of n. Evaluating $f^m_{\lambda,n}$ takes $\mathcal{O}(md)$ time, as opposed to the $\mathcal{O}(nd)$ time for $f_{\lambda,n}$. All matrix computations can be reduced further by not using all d components as in (7), resulting in a $|\mathcal{I}| \times |\mathcal{I}|$ matrix with $|\mathcal{I}| < md$ in (8).

Finite and lite kernel exponential families Strathmann et al. (2015) proposed two alternative approximations to the full model of Section 2, used for efficient score learning in adaptive HMC. Both approaches currently lack convergence guarantees.

The *finite* form uses an m-dimensional $\mathcal{H}$, defined e.g. by random Fourier features (Rahimi and Recht 2007), where (4) can be computed directly in $\mathcal{H}$ in time linear in n. Such parametric features limit the expressiveness of the model: Strathmann et al. (2015) observed that the score estimate oscillates in regions where little or no data has been observed, leading to poor HMC behavior when the sampler enters those regions. We thus do not further pursue this approach in the present work.

The *lite* approximation instead finds the best estimator $f \in \operatorname{span}\{k(x, \cdot)\}_{x \in X}$. This has a similar spirit to Nyström approaches, but note the differing basis from (4), which is based on kernel *derivatives*, and that it uses the entirety of X, so the dependence on n is improved only by simple subsampling. Strathmann et al. (2015) derived an estimator only for Gaussian kernels. Our generalized version of Theorem 1 (Lemma 4 in the appendix) covers the basis used by the lite approximation, allowing us to generalize this method to basis sets $Y \neq X$ and to kernels other than the Gaussian; Appendix B.1 discusses this in more detail.

4 THEORY

We analyze the performance of our estimator in the well-specified case: assuming that the true density p_0 is in $\mathcal{P}$ (and thus corresponds to some $f_0 \in \mathcal{H}$), we obtain both the parameter convergence of $f^m_{\lambda,n}$ to f_0 and the convergence of the corresponding density $p_{f^m_{\lambda,n}}$ to the true density p_0.

Theorem 2. *Assume the conditions listed in Appendix A.3 (similar to those of Sriperumbudur et al. (2017) for the well-specified case), and use the $\mathcal{H}_Y$ of (6) with the basis set Y chosen uniformly at random from the size-m subsets of the training set X, and all md components included. Let $\beta \geq 0$ be the range-space smoothness parameter of the true density f_0, and define $b = \min\left(\beta, \frac{1}{2}\right)$, $\theta = \frac{1}{2(b+1)} \in [\frac{1}{3}, \frac{1}{2}]$. As long as $m = \Omega\left(n^\theta \log n\right)$, then with $\lambda = n^{-\theta}$ we obtain*

$$\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} = \mathcal{O}_{p_0}\left(n^{-\frac{b}{2(b+1)}}\right),$$
$$J(p_0 \| p_{f^m_{\lambda,n}}) = \mathcal{O}_{p_0}\left(n^{-\frac{2b+1}{2(b+1)}}\right).$$

The first statement implies that $p_{f^m_{\lambda,n}}$ also converges to p_0 in L_r $(1 \leq r \leq \infty)$ and Hellinger distances at a rate $\mathcal{O}_{p_0}\left(n^{-\frac{b}{2(b+1)}}\right)$, and that $\mathrm{KL}(p_0 \| p_{f^m_{\lambda,n}}), \mathrm{KL}(p_{f^m_{\lambda,n}} \| p_0)$ are each $\mathcal{O}_{p_0}\left(n^{-\frac{b}{b+1}}\right)$.

The rate of convergence in J exactly matches the rate for the full-data estimator $f_{\lambda,n}$ shown by Sriperumbudur et al. (2017) in J; the rates in other divergences essentially match, except that ours saturate slightly sooner as β increases. Thus, for any problem satisfying the assumptions, we can achieve the same statistical properties as the full-data setting with $m = \Omega\left(\sqrt{n} \log n\right)$, while in the smoothest problems we need only $m = \Omega\left(n^{1/3} \log n\right)$.

This substantial reduction in computational expense is in contrast to the comparable analysis for kernel ridge regression (Rudi et al. 2015), which for the hardest problems requires $m = \Omega(n \log n)$, giving no computational savings at all. In the best general case, it also needs $m = \Omega(n^{1/3} \log n)$. This rate was itself a significant advance: a prior analysis based on stability of the kernel approximation (Cortes et al. 2010) results in a severe additional penalty when using Nyström, matching the worst-case error rates for the full solution, yet still requiring $m = \Omega(n)$ (i.e., according to the earlier reasoning, we would not be guaranteed to benefit from improved rates in easier problems).

A finite-sample version of Theorem 2, with explicit constants, is shown in Appendix C (and used to prove Theorem 2). That version also includes rates for dimension subsampling.

Proof outline Each of the losses considered in Theorem 2 can be bounded in terms of $\|f - f_0\|_{\mathcal{H}}$. We decompose this loss relative to $f^m_\lambda = \operatorname{argmin}_{f \in \mathcal{H}_Y} J(f) + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}}$, the best regularized estimator in population with the particular basis Y. That is,

$$\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} \leq \|f^m_{\lambda,n} - f^m_\lambda\|_{\mathcal{H}} + \|f^m_\lambda - f_0\|_{\mathcal{H}}. \quad (9)$$

The first term on the right-hand side of (9) is the *estimation error*, which represents our error due to having a finite number of samples n: this term decreases as $n \to \infty$, but it will increase as $\lambda \to 0$. It could conceivably increase as $m \to \infty$ as well, but we show

using concentration inequalities in $\mathcal{H}$ that no matter the m, the estimation error is $\mathcal{O}_{p_0}\left(\frac{1}{\lambda\sqrt{n}}\right)$.

The last term of (9) is the *approximation error*, where "approximation" refers both to the regularization by λ and the restriction to the subspace $\mathcal{H}_Y$. This term is independent of n; it decreases as $\mathcal{H}_Y$ grows (i.e. as $m \to \infty$), and also with $\lambda \to 0$, as we allow ourselves to more directly minimize the population risk. The key to bounding this term is to exploit the nature of the space $\mathcal{H}_Y$. This can be done by analogy with the treatment of the "computational error" term of Rudi et al. (2015), where we show that any components of f_0 not lying within $\mathcal{H}_Y$ are relatively small in the parts of the space we observe; this is the only step of the proof that depends on the specific basis $\mathcal{H}_Y$. Having handled this contribution, we show that the approximation error term is $\mathcal{O}_{p_0}\left(\lambda^b\right)$ as long as $m = \Omega\left(\frac{1}{\lambda}\log\frac{1}{\lambda}\right)$.

The decay of the two terms is then optimized when $\lambda = n^\theta$, with θ as given in the proof.

The rate in Fisher divergence J is better because that metric is weighted towards parts of the space where we actually see data, as opposed to uniformly across $\mathcal{H}$ as in (9). Our proof technique, similarly to that of Sriperumbudur et al. (2017), allows us to account for this with an improved dependence on λ in the evaluation of both estimation and approximation errors.

Remarks Our proof uses techniques both from the analysis of the full-data estimator (Sriperumbudur et al. 2017) and from an analysis of generalization error for Nyström-subsampled kernel ridge regression (Rudi et al. 2015). There are some major differences from the regression case, however. The decomposition (9) differs from the regression decomposition (Rudi et al.'s Appendix E), as differences in the structure of the problem make the latter inapplicable. Correlations between dimensions in our setup also make certain concentration results much more difficult: compare our Appendix D.2 to Rudi et al.'s Proposition 8.

Approaches like those of El Alaoui and Mahoney (2015) and Yang et al. (2017), which bound the difference in training error of Nyström-type approximations to kernel ridge regression, are insufficient for our purposes: we need to ensure that the estimated function $f^m_{\lambda,n}$ converges to f_0 everywhere, so that the full distribution matches, not just its values at the training points. In doing so, our work is heavily indebted to Caponnetto and De Vito (2007), as are Rudi et al. (2015) and Sriperumbudur et al. (2017).

We previously noted that using $Y = X$ does not yield an identical estimator, $f^n_{\lambda,n} \neq f_{\lambda,n}$. In fact, we could achieve this by additionally including $\hat{\xi}$ within (6), but since evaluating $\hat{\xi}$ requires touching all the data points we would lose the test-time improvements achieved by the estimator of Theorem 1. Alternatively, we could still "forget" points, but double the size of the basis, by including $\partial_i^2 k(X_a, \cdot)$. In practice, $f^n_{\lambda,n}$ performs about as well as $f_{\lambda,n}$, so neither method seems necessary. See also Appendix C.1.1 for more theoretical intution on why this may not be needed.

5 EXPERIMENTS

We now validate our estimator empirically. We first consider synthetic densities in Section 5.1, where we know the true densities and can evaluate convergence of the score estimates analytically with (1), including a case with subsampled basis components in Section 5.1.1. In Section 5.2 we evaluate our estimator in the gradient-free Hamiltonian Monte Carlo setting of Strathmann et al. (2015), where (in the absence of a ground truth) we compare the efficiency of the resulting sampler.

For all exponential family variants, we take q_0 to be a uniform distribution with support encompassing the samples, and use a Gaussian kernel $k(x, y) = \exp\left(-\|x - y\|^2/\sigma\right)$, tuning the bandwidth σ and regularization parameter λ via a validation set. We compare the following models:

full: Sriperumbudur et al. (2017)'s model, (4) and (5).

lite: Strathmann et al. (2015)'s heuristic approximation, which subsamples the dataset X to size m, and uses the basis $\{k(X_a, \cdot)\}$, ignoring the remaining datapoints. We use the regularization from their latest code, $\lambda(\|f\|^2_{\mathcal{H}} + \|\beta\|^2_2)$.

nyström: The estimator of Theorem 1, choosing m distinct data points uniformly at random for Y. For numerical stability, we add $10^{-5}I$ to the matrix being inverted in (8), corresponding to a small L_2 regularizer on the weights β.

dae: The model of Alain and Bengio (2014), where we train a two-layer denoising autoencoder, with tanh code activations and linear decoding. We train with decreasing noise levels (100σ, 10σ, σ), using up to 1000 iterations of BFGS each. We tune the number of hidden units and σ; while Alain and Bengio (2014) recommend simply choosing some small σ, this plays a similar role to a bandwidth, and its careful choice is essential. We differentiate the score estimate to obtain the second derivative needed in (2).

See `github.com/karlnapf/nystrom-kexpfam` for code for the models and to reproduce the experiments.

5.1 Score convergence on synthetic densities

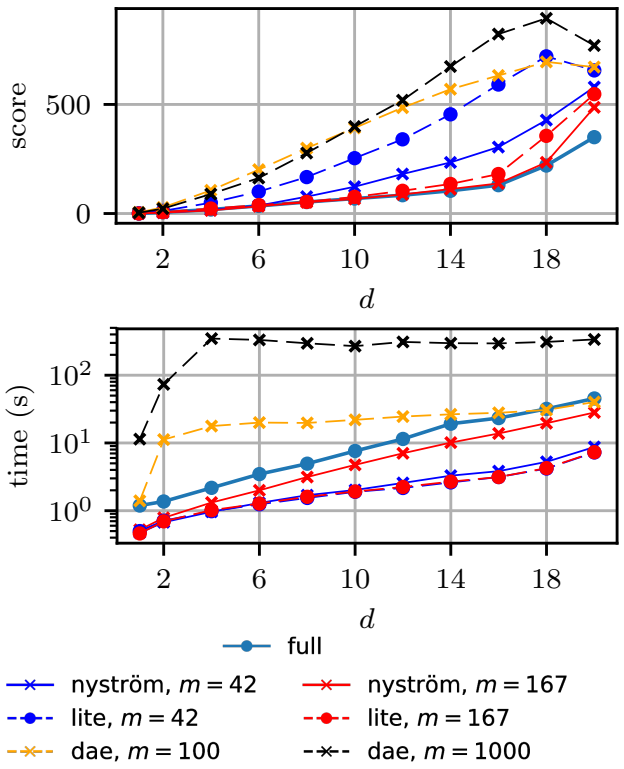
We first consider two synthetic densities, where the true score is available. The `ring` dataset takes inspiration

from the "spiral" dataset of Alain and Bengio (2014, Figure 5), being a similarly-shaped distribution but possessing a probability density for evaluation purposes. We sample points uniformly along three circles with radii $(1, 3, 5)$ in $\mathbb{R}^2$ and add $\mathcal{N}(0, 0.1^2)$ noise in the radial direction. We then add extra dimensions consisting of independent Gaussian noise with standard deviation 0.1. The `grid` dataset is a more challenging variant of the 2-component mixture example of Sriperumbudur et al. (2017, Figure 1). We fix d random vertices of a d-dimensional hypercube; the target is a mixture of normal distributions, one at each vertex.

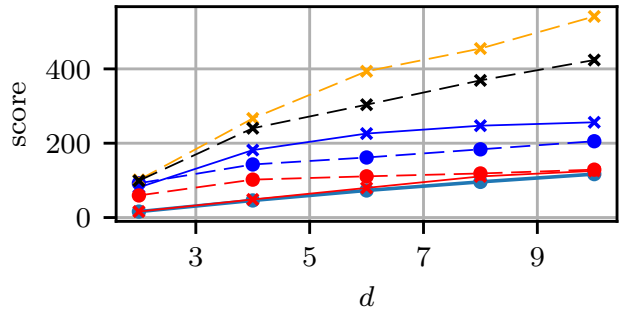
For each run, we generate $n = 500$ training points and estimate the score on 1500 (`grid`) or 5000 (`ring`) newly generated test points. We estimate the true score (1) on these test points to ensure a "best case" comparison of the models, though using (2) leads to indistinguishable parameter selections and performance. For lite and nyström, we independently evaluated the parameters for each subsampling level. We report performances for the best parameters found for each method. All experiments were conducted in a single CPU thread for timing comparisons, although multi-core parallelization is straightforward for each model.

Figure 1 shows convergence of the score as the dimension increases. On both the `ring` and `grid` datasets, nyström performs very close to the full solution, while showing large computational savings. With a reasonable drop in score at $m = 42$, we achieve a major reduction in cost and storage over the original $n = 500$ sample size. The lite performance is similar to that of nyström at comparable levels of data retention. As expected, the performance of nyström gets closer to that of full as m increases towards n. The autoencoder performs consistently worse than any of the kernel models, on both datasets. Autoencoder results are also strongly clustered, with only small performance improvements as the number of hidden units increases. As the `grid` data reaches 20 dimensions, all solutions start to converge to a similar score. None of the methods are able to learn the structure for this number of training points and dimensions; all solutions effectively revert to smooth, uninformative estimates.

The lite solution is fastest, followed by nyström for low to moderate m, with significant savings over the full solution even at $m = 167$ on `grid`, and across all m on `ring`. The additional cost of nyström over lite arises since it computes all derivatives at the retained samples. Autoencoder runtimes are longer than the other methods, although we point out that the settings of Alain and Bengio (2014) are not optimized for runtime. We observed, however, that replacing BFGS with stochastic gradient descent or avoiding the decreasing noise schedule both lead to instabilities in the solution.



(a) Scores and runtimes on the `grid` dataset.



(b) Scores on the `ring` dataset; same labels as for `grid`.

Figure 1: Convergence and timing on synthetic data.

5.1.1 Dimension subsampling

To quantify the effect of subsampling components of the Nyström basis in (7), we repeat the previous `grid` experiment with another version of our estimator: nyström D has the same number md of basis functions as nyström, but rather than using all d components of m uniformly chosen training points, we uniformly choose md of *all* available components. That is, we pick $Y = X$ in (7) and $\mathcal{I} \subset [n] \times [d], |\mathcal{I}| = md$. This equalizes the cost of (8) for nyström and nyström D.

Figure 2 shows that distributing the used components across all training data helps slightly when m is small. Yet this benefit comes at a cost: as mentioned in Section 3, nyström can discard training data not used in

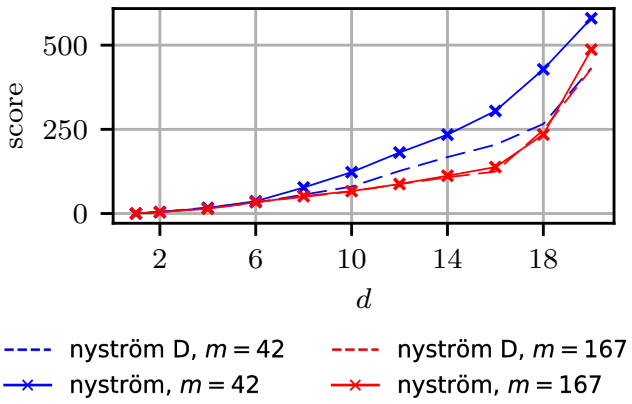


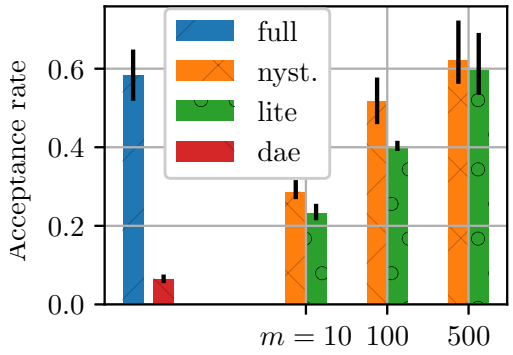
Figure 2: Dimension sub-sampling for `grid`.

the basis after fitting. For nyström D, however, we can only discard training data if *no* components were chosen, so we must retain many more points.

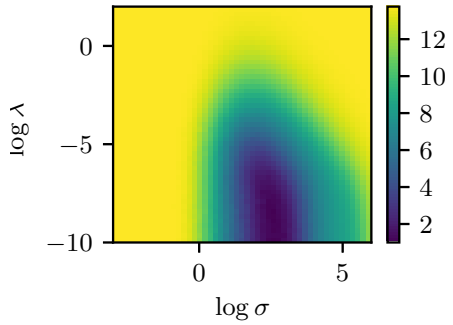
5.2 Gradient-free Hamiltonian Monte Carlo

Our final experiment follows methodology and code by Sejdinovic et al. (2014) and Strathmann et al. (2015) in constructing a gradient-free HMC sampler using score estimates learned on the previous MCMC samples. Our goal is to efficiently sample from the marginal posterior over hyperparameters of a Gaussian process (GP) classifier on the UCI Glass dataset (Lichman 2013). Closed-form expressions for the score (and therefore HMC itself) are unavailable, due to the intractability of the marginal data likelihood given the hyperparameters. But one can construct a Pseudo-Marginal MCMC method using an Expectation Propagation approximation to the GP posterior and importance sampling (Filippone and Girolami 2014). We compare all score estimators' ability to generate an HMC-like proposal as in Strathmann et al. (2015). An accurate score estimate would give proposals close to an HMC move, which would have high acceptance probability. Thus higher acceptance rates indicate better score estimates.

Our experiment assumes the idealized scenario where a burn-in is successfully completed. We run 40 random walk adaptive-Metropolis MCMC samplers for 30 000 iterations, discard the first 10 000 samples, and thin by a factor of 400. Merging these samples results in 2 000 posterior samples. We fit all score estimators on a random subset of $n = 500$ of these samples, and use the remaining 1500 samples to tune the model hyperparameters. The validation surface obtained for nyström by the estimated score objective on the held-out set is shown in Figure 3: it is smooth and easily optimized. For dae (not shown here), a well-tuned level of corruption noise is essential. Starting from a random point of the initial posterior sketch, we construct trajectories



(a) HMC acceptance rates, with 90% quantiles.



(b) Log scores for various hyperparameters, for nyström with $m = 42$.

Figure 3: Results for GP hyperparameter optimization on the UCI Glass dataset.

along the surrogate Hamiltonian using 100 steps of size 0.1, and a standard Gaussian momentum. We compute the hypothetical acceptance probability for each step, and average over the trajectory.

Figure 3 shows the results averaged over 200 repetitions. As before, nyström matches the performance of full for $m = n = 500$, while for $m = 100$ it attains a high acceptance rate at a considerably reduced computational cost. It also reliably outperforms lite for lower m, which might occur since lite sub-samples the data while nyström only sub-samples the basis. dae does relatively poorly, despite a large grid-search for its hyperparameters. For any of the models, untuned hyperparameters yield an acceptance rate close to zero.

6 CONCLUSION

We proposed a Nyström approximation for score matching in kernel exponential families. Theorem 2 establishes that the proposed algorithm can achieve the same or nearly the same bound on convergence as the full algorithm, with $m \ll n$. We also demonstrated the efficacy of the approach on challenging synthetic datasets and on an approximate HMC problem for optimizing GP hyperparameters. These cost reductions help make estimation in this rich family of distributions practical.

Acknowledgements

The authors would like to thank Mladen Kolar for productive discussions.

References

Alain, G. and Y. Bengio (2014). "What regularized auto-encoders learn from the data-generating distribution." In: *JMLR* 15.1, pp. 3563–3593. arXiv: 1211.4246.

Barron, A. and C.-H. Sheu (1991). "Approximation of density functions by sequences of exponential families." In: *Annals of Statistics* 19.3, pp. 1347–1369.

Berlinet, A. and C. Thomas-Agnan (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer.

Brown, L. D. (1986). *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Hayward, CA: IMS.

Canu, S. and A. J. Smola (2006). "Kernel methods and the exponential family." In: *Neurocomputing* 69.7, pp. 714–720.

Caponnetto, A. and E. De Vito (2007). "Optimal rates for regularized least-squares algorithm." In: *Foundations of Computational Mathematics* 7.3, pp. 331–368.

Cortes, C., M. Mohri, and A. Talwalkar (2010). "On the impact of kernel approximation on learning accuracy." In: *AISTATS*.

Drineas, P. and M. W. Mahoney (2005). "On the Nyström method for approximating a Gram matrix for improved kernel-based learning." In: *Journal of Machine Learning Research* 6, pp. 2153–2175.

El Alaoui, A. and M. W. Mahoney (2015). "Fast Randomized Kernel Methods With Statistical Guarantees." In: *NIPS*. arXiv: 1411.0306.

Filippone, M. and M. Girolami (2014). "Pseudo-marginal Bayesian inference for Gaussian Processes." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Fukumizu, K. (2009). "Exponential manifold by reproducing kernel Hilbert spaces." In: *Algebraic and Geometric Methods in Statistics*. Cambridge University Press, pp. 291–306.

Gu, C. and C. Qiu (1993). "Smoothing spline density estimation: Theory." In: *Annals of Statistics* 21.1, pp. 217–234.

Hyvärinen, A. (2005). "Estimation of non-normalized statistical models by score matching." In: *JMLR* 6.Apr, pp. 695–709.

LeCun, Y., S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang (2006). "A Tutorial on Energy-Based Learning." In: *Predicting Structured Data*. MIT Press, pp. 191–246.

Lichman, M. (2013). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Neal, R. (2011). "MCMC using Hamiltonian dynamics." In: *Handbook of Markov Chain Monte Carlo* 2.

Oates, C. J., M. Girolami, and N. Chopin (2017). "Control functionals for Monte Carlo integration." In: *J. R. Statist. Soc. B* 79.3, pp. 695–718. arXiv: 1410.2392.

Rahimi, A. and B. Recht (2007). "Random Features for Large-Scale Kernel Machines." In: *NIPS*.

Rudi, A., R. Camoriano, and L. Rosasco (2015). "Less is More: Nyström Computational Regularization." In: *NIPS*. arXiv: 1507.04717.

Rudi, A. and L. Rosasco (2017). "Generalization Properties of Learning with Random Features." In: *NIPS*. arXiv: 1602.04474.

Schölkopf, B., R. Herbrich, and A. J. Smola (2001). "A Generalized Representer Theorem." In: *COLT*.

Sejdinovic, D., H. Strathmann, M. Lomeli, C. Andrieu, and A. Gretton (2014). "Kernel Adaptive Metropolis-Hastings." In: *ICML*.

Smola, A. J. and B. Schölkopf (2000). "Sparse Greedy Matrix Approximation for Machine Learning." In: *ICML*.

Sriperumbudur, B. K., K. Fukumizu, R. Kumar, A. Gretton, A. Hyvärinen, and R. Kumar (2017). "Density Estimation in Infinite Dimensional Exponential Families." In: *Journal of Machine Learning Research* 18.57, pp. 1–59. arXiv: 1312.3516.

Sriperumbudur, B. K. and Z. Szábo (2015). "Optimal rates for random Fourier features." In: *NIPS*. arXiv: 1506.02155.

Strathmann, H., D. Sejdinovic, S. Livingstone, Z. Szábo, and A. Gretton (2015). "Gradient-free Hamiltonian Monte Carlo with Efficient Kernel Exponential Families." In: *NIPS*. arXiv: 1506.02564.

Sutherland, D. J. and J. Schneider (2015). "On the Error of Random Fourier Features." In: *UAI*. arXiv: 1506.02785.

Williams, C. K. I. and M. Seeger (2000). "Using the Nyström method to speed up kernel machines." In: *NIPS*.

Woodruff, D. P. (2014). "Sketching as a Tool for Numerical Linear Algebra." In: *Foundations and Trends in Theoretical Computer Science* 10.1–2, pp. 1–157. arXiv: 1411.4357.

Yang, Y., M. Pilanci, and M. J. Wainwright (2017). "Randomized Sketches for Kernels: Fast and Optimal Non-Parametric Regression." In: *Annals of Statistics* 45.3, pp. 991–1023. arXiv: 1501.06195.