Thesis Proposal:

Scalable, Active and Flexible Learning on Distributions

Dougal J. Sutherland

November 17, 2015

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jeff Schneider (Chair)
Maria-Florina Balcan
Barnabás Póczos
Arthur Gretton (UCL)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

A wide range of machine learning problems, including astronomical inference about galaxy clusters, natural image scene classification, parametric statistical inference, and predictions of public opinion, can be well-modeled as learning a function on (samples from) distributions. This thesis explores problems in learning such functions via kernel methods.

The first challenge is one of computational efficiency when learning from large numbers of distributions: the computation of typical methods scales between quadratically and cubically, and so they are not amenable to large datasets. We investigate the approach of approximate embeddings into Euclidean spaces such that inner products in the embedding space approximate kernel values between the source distributions. We present a new embedding for a class of information-theoretic distribution distances, and evaluate it and existing embeddings on several real-world applications. We also propose the integration of these techniques with deep learning models so as to allow the simultaneous extraction of rich representations for inputs with the use of expressive distributional classifiers.

In a related problem setting, common to astrophysical observations, autonomous sensing, and electoral polling, we have the following challenge: when observing samples is expensive, but we can choose where we would like to do so, how do we pick where to observe? We propose the development of a method to do so in the distributional learning setting (which has a natural application to astrophysics), as well as giving a method for a closely related problem where we search for instances of patterns by making point observations.

Our final challenge is that the choice of kernel is important for getting good practical performance, but how to choose a good kernel for a given problem is not obvious. We propose to adapt recent kernel learning techniques to the distributional setting, allowing the automatic selection of good kernels for the task at hand. Integration with deep networks, as previously mentioned, may also allow for learning the distributional distance itself.

Throughout, we combine theoretical results with extensive empirical evaluations to increase our understanding of the methods.

Contents

Chapter 1

Introduction

Traditional machine learning approaches focus on learning problems defined on vectors, mapping whatever kind of object we wish to model to a fixed number of real-valued attributes. Though this approach has been very successful in a variety of application areas, choosing natural and effective representations can be quite difficult.

In many settings, we wish to perform machine learning tasks on objects that can be viewed as a collection of lower-level objects or more directly as samples from a distribution. For example:

- Images can be thought of as a collection of local patches (Section 4.2); similarly, videos are collections of frames.
- The total mass of a galaxy cluster can be predicted based on the positions and velocities of individual galaxies (Section 4.3).
- Support for a political candidate among various demographic groups can be estimated by learning a regression model from electoral districts of individual voters to district-level support for political candidates (Flaxman et al. 2015).
- Documents are made of sentences, which are themselves composed of words, which themselves can be seen as being represented by sets of the contexts in which they appear (Section 6.2).
- Parametric statistical inference problems learn a function from sample sets to model parameters (Section 4.1).
- Expectation propagation techniques relay on maps from sample sets to messages normally computed via expensive numerical integration (Jitkrittum et al. 2015).
- Causal arrows between distributions can be estimated from samples (Lopez-Paz et al. 2015).

In order to use traditional techniques on these collective objects, we must create a single vector that represents the entire set. Though there are various ways to summarize a set as a vector, we can often discard less information and require less effort in feature engineering by operating directly on sets of feature vectors.

One method for machine learning on sets is to consider them as samples from some unknown underlying probability distribution over feature vectors. Each example then has its own distribution: if we are classifying images as sets of patches, each image is defined as a distribution over patch features, and each class of clusters is a set of patch-level feature distributions. We can then

define a kernel based on statistical estimates of a distance between probability distributions. Letting $\mathcal{X} \subseteq \mathbb{R}^d$ denote the set of possible feature vectors, we thus define a kernel $k : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$. This lets us perform classification, regression, anomaly detection, clustering, low-dimensional embedding, and any of many other applications with the well-developed suite of kernel methods. Chapter 2 discusses various such kernels and their estimators; Chapter 4 gives empirical results on several problems.

When used for a learning problem with N training items, however, typical kernel methods require operating on an $N \times N$ kernel matrix, which requires far too much computation to scale to datasets with a large number of instances. Chapter 3 discusses one way to avoid this problem: approximate embeddings $z : \mathcal{X} \to \mathbb{R}^D$, à la Rahimi and Recht (2007), such that $z(x)^\mathsf{T} z(y) \approx k(x, y)$. These embeddings are available for several distributional kernels, and are also evaluated empirically in Chapter 4.

Chapter 5 addresses the application of this type of complex functional classifier to an active search problem. Consider finding polluted areas in a body of water, based on point measurements. We wish to, given an observation budget, adaptively choose where we should make these observations in order to maximize the number of regions we can be confident are polluted. If our notion of "pollution" is defined simply by a threshold on the mean value of a univariate measurement, Ma, Garnett, et al. (2014) give a natural selection algorithm based on Gaussian process inference. If, instead, our sensors measure the concentrations of several chemicals, the vector flow of water current, or other such more complicated data, we can instead apply a classifier to a region and consider the problem of finding regions that the classifier marks as relevant.

One area of proposed work, discussed in Section 6.5, bridges the problems of learning on distributions in Chapters 2 to 4 with that of active pattern search in Chapter 5. Specifically, we would like to consider the problem of *active learning on distributions*. There are several possible avenues of incorporating active selection into distribution learning: given a noisy understanding of a distribution, which points should be selected for more careful measurement? Which distributions should be measured in order to best accomplish our objectives? This problem is intimately related to the setting of Chapter 5 when regions are independent of one another.

Other areas of future work, discussed in Chapter 6, propose integrating the distributional embeddings of Chapter 3 with deep learning models (Section 6.1) and kernel learning techniques (Section 6.3), applying them to word embeddings in natural language processing (Section 6.2), and developing scalable embeddings for other forms of distributional kernel (Section 6.4).

1.1 Summary of contributions

- Section 3.1 improves the theoretical understanding of the random Fourier features of Rahimi and Recht (2007). (Based on Sutherland and Schneider 2015.)
- Section 3.3.1 provides a method to scale the L_2 embedding of J. B. Oliva, Neiswanger, et al. (2014) to higher dimensions in some situations.
- Section 3.4 gives an approximate embedding for a new class of distributional distances. (Based on Sutherland, J. B. Oliva, et al. 2015.)
- Chapter 4 provides three empirical studies for the application of distributional distances to

practical problems. (Based on Póczos, Xiong, Sutherland, et al. 2012; Ntampaka, Trac, Sutherland, Battaglia, et al. 2014; Sutherland, J. B. Oliva, et al. 2015.)

- Chapter 5 presents and analyzes a method for the novel problem setting of *active pointillistic pattern search*. (Based on Ma, Sutherland, et al. 2015.)

Chapter 6 proposes further work related to these areas.

1.2 The `gestalt-learn` package

Efficient implementations of many of the methods for learning on distributions discussed in this thesis are available in the Python package `gestalt-learn`[1], and more will be available soon. This package integrates with the standard Python numerical ecosystem and presents an API compatible with that of `scikit-learn` (Pedregosa et al. 2011).

[1]Currently https://github.com/dougalsutherland/skl-groups, soon to be renamed to https://github.com/dougalsutherland/gestalt-learn.

Chapter 2

Learning on distributions

As discussed in Chapter 1, we consider the problem of learning on probability distributions. Specifically: let $\mathcal{X} \subseteq \mathbb{R}^d$ be the set of observable feature vectors, and $\mathcal{P}$ the set of probability distributions under consideration. We then perform machine learning on samples from distributions by:

1. Choosing a distance $\rho : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$.
2. Defining a Mercer kernel $k : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ based on ρ.
3. Estimating k based on the observed samples as $\hat{k} : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$, which should itself be a kernel on $2^{\mathcal{X}}$.
4. Using $\hat{k}$ in a standard kernel method, such as an SVM or a Gaussian Process, to perform classification, regression, collective anomaly detection, or other machine learning tasks.

Certainly, this is not the only approach to learning on distributions. Some distributional learning methods do not directly compare sample sets to one another, but rather compare their elements to a class-level distribution (Boiman et al. 2008). Given a distance ρ, one can naturally use k-nearest neighbor models (Póczos, Xiong, and Schneider 2011; Kusner et al. 2015), or Nadaraya-Watson–type local regression models (J. B. Oliva, Póczos, et al. 2013; Póczos, Rinaldo, et al. 2013) with respect to that distance. In this thesis, however, we focus on kernel methods as a well-studied, flexible, and empirically effective approach to a broad variety of learning problems.

We typically assume that every distribution in $\mathcal{P}$ is absolutely continuous with respect to the Lebesgue measure, and slightly abuse notation by using distributions and their densities interchangeably.

2.1 Distances on distributions

We will define kernels on distributions by first defining distances between them. We first present four general frameworks for such distances:

L_r metrics One natural way to compute distances between distributions is the L_r metric between their densities, for order $r \geq 1$:

$$L_r(p, q) \coloneqq \left(\int_{\mathcal{X}} |p(x) - q(x)|^r \, \mathrm{d}x \right)^{1/r}.$$

Note that the limit $r = \infty$ yields the distance $L_\infty(p, q) = \sup_{x \in \mathcal{X}} |p(x) - q(x)|$.

f-divergences For any convex function f with $f(1) = 0$, the f-divergence of P to Q is

$$D_f(P \| Q) \coloneqq \int_{\mathcal{X}} f\left(\frac{p(x)}{q(x)} \right) q(x) \, \mathrm{d}x.$$

This class is sometimes called "Csiszár f-divergences", after Csiszár (1963). Sometimes the requirement of convexity or that $f(1) = 0$ is dropped. Note that these functions are not in general symmetric or respecting of the triangle inequality. They do, however, satisfy $D_f(P \| P) = 0$, $D_f(P \| Q) \geq 0$, and are jointly convex:

$$D_f(\lambda P + (1 - \lambda) P' \| \lambda Q + (1 - \lambda) Q') \leq \lambda D_f(P \| Q) + (1 - \lambda) D_f(P' \| Q').$$

In fact, the only metric f-divergences are multiples of the total variation distance, discussed shortly (Khosravifard et al. 2007) — though e.g. the Hellinger distance is the square of a metric. For an overview, see e.g. Liese and Vajda (2006).

α-β divergences The following somewhat less-standard divergence family, defined e.g. by Póczos, Xiong, Sutherland, et al. (2012) generalizing the α-divergence of Amari (1985), is also useful. Given two real parameters α, β, $D_{\alpha,\beta}$ is defined as

$$D_{\alpha,\beta}(P \| Q) \coloneqq \int p^\alpha(x) \, q^\beta(x) \, p(x) \, \mathrm{d}x.$$

$D_{\alpha,\beta}(P \| Q) \geq 0$ for any α, β; $D_{\alpha,-\alpha}(P \| P) = 0$. Note also that $D_{\alpha,-\alpha}$ has the form of an f-divergence with $t \mapsto t^{\alpha+1}$, though this does not satisfy $f(1) = 0$ and is convex only if $\alpha \notin (-1, 0)$.

Integral probability metrics Many useful metrics can be expressed as *integral probability metrics* (IPMs, Müller 1997):

$$\rho_{\mathfrak{F}}(P, Q) \coloneqq \sup_{f \in \mathfrak{F}} \left| \int f \, \mathrm{d}P - \int f \, \mathrm{d}Q \right|,$$

where $\mathfrak{F}$ is some family of functions $f : \mathcal{X} \to \mathbb{R}$. Note that $\rho_{\mathfrak{F}}$ satisfies $\rho_{\mathfrak{F}}(P, P) = 0$, $\rho_{\mathfrak{F}}(P, Q) = \rho_{\mathfrak{F}}(Q, P)$, and $\rho_{\mathfrak{F}}(P, Q) \leq \rho_{\mathfrak{F}}(P, R) + \rho_{\mathfrak{F}}(R, Q)$ for any $\mathfrak{F}$; the only metric property which depends on $\mathfrak{F}$ is $(\rho_{\mathfrak{F}}(P, Q) = 0) \implies (P = Q)$. Sriperumbudur et al. (2009) give an overview.

The various distributional distances below can often be represented in one or more of these frameworks.

L_2 distance The L_2 distance is one of the most common metrics used on distributions. It can also be represented as $D_{1,0} - 2D_{0,1} + D_{-1,2}$.

Kullback-Leibler divergence The Kullback-Leibler (KL) divergence is defined as

$$\textsc{kl}(P\|Q) \coloneqq \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x.$$

For discrete distributions, the KL divergence bears a natural information theoretic interpretation as the expected excess code length required to send a message for P via the optimal code for Q. It is nonnegative, and zero iff $P = Q$ almost everywhere; however, $\textsc{kl}(P\|Q) \neq \textsc{kl}(Q\|P)$ in general. Note also that if there is any point with $p(x) > 0$ and $q(x) = 0$, $\textsc{kl}(P\|Q) = \infty$.

Applications often use a symmetrization by averaging with the dual:

$$\textsc{skl}(P, Q) \coloneqq \tfrac{1}{2} \left(\textsc{kl}(P\|Q) + \textsc{kl}(Q\|P)\right).$$

SKL, however, still does not satisfy the triangle inequality. It is also sometimes called Jeffrey's divergence, though that name is also sometimes used to refer to the Jensen-Shannon divergence (below), so we avoid it.

KL can be viewed as a f divergence, with one direction corresponding to $t \mapsto t \log t$ and the other to $t \mapsto -\log t$; SKL is thus an f divergence with $t \mapsto \frac{1}{2}(t-1)\log t$.

Jensen-Shannon divergence The Jensen-Shannon divergence is based on KL:

$$\textsc{js}(P, Q) \coloneqq \tfrac{1}{2}\,\textsc{kl}\left(P \middle\| \frac{P+Q}{2}\right) + \tfrac{1}{2}\,\textsc{kl}\left(Q \middle\| \frac{P+Q}{2}\right),$$

where $\frac{P+Q}{2}$ denotes an equal mixture between P and Q. JS is clearly symmetric, and in fact $\sqrt{\textsc{js}}$ satisfies the triangle inequality. Note also that $0 \leq \textsc{js}(P, Q) \leq \log 2$. It gets its name from the fact that it can be written as the Jensen difference of the Shannon entropy:

$$\textsc{js}(P, Q) = \mathrm{H}\left[\frac{P+Q}{2}\right] - \frac{\mathrm{H}[P] + \mathrm{H}[Q]}{2},$$

a view which allows a natural generalization to more than two distributions. Non-equal mixtures are also natural, but of course asymmetric. For more details, see e.g. Martins et al. (2009).

Rényi-α divergence The Rényi-α divergence generalizes KL as

$$\textsc{r}_\alpha(P\|Q) \coloneqq \frac{1}{\alpha - 1} \log \int p(x)^\alpha q(x)^{1-\alpha} \,\mathrm{d}x;$$

note that $\lim_{\alpha\to 1} \mathrm{R}_\alpha(P\|Q) = \mathrm{KL}(P\|Q)$. Like KL, R is asymmetric; we similarly define a symmetrization

$$\mathrm{SR}_\alpha(P,Q) \coloneqq \tfrac{1}{2}\left(\mathrm{R}_\alpha(P\|Q) + \mathrm{R}_\alpha(Q\|P)\right).$$

SR_α does not satisfy the triangle inequality.

R_α can be represented based on an α-β divergence: $\mathrm{R}(P\|Q) = \frac{1}{\alpha-1}\log D_{\alpha-1,1-\alpha}(P\|Q)$.

A Jensen-Rényi divergence, defined by replacing KL with R_α in the definition of JS, has also been studied (Martins et al. 2009), but we will not consider it here.

Total variation distance The total variation distance (TV) is such an important distance that it is sometimes referred to simply as "the stastitical distance." It can be defined as

$$\mathrm{TV}(P,Q) = \sup_A |P(A) - Q(A)|,$$

where A ranges over every event in the underlying sigma-algebra. It can also be represented as $\frac{1}{2}L_1(P,Q)$, as an f-divergence with $t \mapsto |t-1|$, and as an IPM with $\mathfrak{F} = \{f : \|f\|_\infty \le 1\}$. (Recall that $\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|$.) Note that TV is a metric, and $0 \le \mathrm{TV}(P,Q) \le 1$.

The total variation distance is closely related to the "intersection distance", most commonly used on histograms (Cha and Srihari 2002):

$$\int_{\mathcal{X}} \min(p(x), q(x))\,\mathrm{d}x = \int_{\mathcal{X}} \tfrac{1}{2}\left(p(x) + q(x) - |p(x) - q(x)|\right)\mathrm{d}x = 1 - \mathrm{TV}(P,Q).$$

Hellinger distance The square of the Hellinger distance H is defined as

$$\mathrm{H}^2(P,Q) \coloneqq \tfrac{1}{2}\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x = 1 - \int \sqrt{p(x)\,q(x)}\,\mathrm{d}x.$$

H^2 can be expressed as an f-divergence with either $t \mapsto \frac{1}{2}(\sqrt{t}-1)^2$ or $t \mapsto 1 - \sqrt{t}$; it is also closely related to an α-β divergence as $\mathrm{H}^2(P,Q) = 1 - D_{-1/2,1/2}$. H is a metric, and is bounded in $[0,1]$. It is proportional to the L_2 difference between $\sqrt{p}$ and $\sqrt{q}$, which yields the bounds $\mathrm{H}^2(P,Q) \le \mathrm{TV}(P,Q) \le \sqrt{2}\,\mathrm{H}(P,Q)$.

Earth mover's distance The earth mover's distance (EMD_ρ) is defined for a metric ρ as

$$\mathrm{EMD}_\rho(P,Q) \coloneqq \inf_{R\in\Gamma(P,Q)} \mathbb{E}_{(X,Y)\sim R}\left[\rho(X,Y)\right],$$

where $\Gamma(P,Q)$ is the set of joint distributions with marginals P and Q. It is also called the first *Wasserstein distance*, or the *Mallows distance*. When $(\mathcal{X},\rho)$ is separable, it is also equal to the *Kantorovich metric*, which is the IPM with $\mathfrak{F} = \{f : \|f\|_L \le 1\}$, where $\|f\|_L \coloneqq \sup\{|f(x) - f(y)|/\rho(x,y) \mid x \ne y \in \mathcal{X}\}$ is the Lipschitz semi-norm.

For discrete distributions, EMD can be computed via linear programming, and is popular in the computer vision community.

Maximum mean discrepancy The maximum mean discrepancy (MMD, Sriperumbudur, Gretton, et al. 2010; Gretton, Borgwardt, et al. 2012) is defined by embedding distributions into a reproducing kernel Hilbert space (RKHS; for an overview see Berlinet and Thomas-Agnan 2004). Let k be the kernel associated with some RKHS $\mathcal{H}$ with feature map $\varphi : \mathcal{X} \to \mathcal{H}$, such that $\langle \varphi(x), \varphi(y) \rangle = k(x, y)$. We can then map a distribution P to its mean embedding $\mu(P) \coloneqq \mathbb{E}_{X \sim P}[\varphi(X)]$, and define the distance between distributions as the distance between their mean embeddings:

$$\mathrm{MMD}_k(P, Q) \coloneqq \|\mu(P) - \mu(Q)\|.$$

MMD_k can also be viewed as an IPM with $\mathfrak{F} = \{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \leq 1\}$, where $\|f\|_{\mathcal{H}}$ is the norm in $\mathcal{H}$. (If $f \in \mathcal{H}$, $f(\cdot) = \sum_{i=1}^{\infty} \alpha_i k(x_i, \cdot)$ for some points $x_i \in \mathcal{X}$ and weights $\alpha_i \in \mathbb{R}$; $\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{\infty} \alpha_i f(x_i)$.)

The mean embedding always exists when the base kernel k is bounded; ρ is metric for when it is characteristic. See Sriperumbudur, Gretton, et al. (2010) and Gretton, Borgwardt, et al. (2012) for details.

Szabó et al. (2014) proved learning-theoretic bounds on the use of ridge regression with MMD.

2.2 Estimators of distributional distances

We now discuss methods for estimating different distributional distances ρ.

The most obvious estimator of most distributional distances is perhaps to first perform density estimation, and then compute distances between the density estimates: the plug-in approach. These approaches suffer from the problem that the density is in some sense a nuisance parameter for the problem of distance estimation, and density estimation is quite difficult, particularly in higher dimensions.

Some of the methods below are plug-in methods; others correct a plug-in estimate, or use inconsistent density estimates in such a way that the overall divergence estimate is consistent.

Parametric models Closed forms of some distances are available for certain distributions:

- For members of the same exponential family, closed forms of the Bhattacharyya kernel (corresponding to Hellinger distance), and certain other kernels of the form $D_{\alpha-1,\alpha}$ were computed by Jebara et al. (2004). Nielsen and Nock (2011) give closed forms for all $D_{\alpha-1,1-\alpha}$, allowing the computation of R, KL, and related divergences in addition to H.
- For Gaussian distributions, Muandet, Schölkopf, et al. (2012) compute the closed form of MMD for a few base kernels. The Euclidean EMD is also available.[1]
- For mixture distributions, L_2 and MMD can be computed based on the inner products between the components by simple linearity arguments. For mixtures specifically of Gaussians, F. Wang et al. (2009) obtain the quadratic (R_2) entropy, which allows the computation of Jensen-Rényi divergences for $\alpha = 2$.

For cases when a closed form does not exist, numerical integration may be necessary, often obviating the computational advantages of this approach.

[1] http://stats.stackexchange.com/a/144896/9964

It is thus possible to fit a parametric model to each distribution and compute distances between the fits; this is done for machine learning applications by e.g. Jebara et al. (2004) and Moreno et al. (2004). In practice, however, we rarely know that a given parametric family is appropriate, and so the use of parametric models introduces unavoidable approximation error and bias.

Histograms One common method for representing distributions is the use of histograms; many distances ρ are then simple to compute. The prominent exception to that is EMD, which requires $O(m^3 \log m)$ time for exact computation with m-bin histograms, though in some settings $O(m)$ approximations are available (Shirdhonkar and Jacobs 2008). MMD also requires approximately $O(m^2)$ computation for typical histograms.

The main disadvantages of histograms are their poor performance in even moderate dimensions, and the fact that (for most ρs) choosing the right bin size is both quite important and quite difficult, since nearby bins do not affect one another.

Vector quantization An improvement over histograms popular in computer vision is to instead quantize distributions to group points by their nearest *codeword* from a dictionary, often learned via k-means or a similar algorithm. This method is known as the *bag of words* (BOW) approach and was popularized by Leung and Malik (2001). This method scales to much higher dimensions than the histogram approach, but suffers from similar problems related to the hard assignment of sample points to bins.

Grauman and Darrell (2007) uses multiple resolutions of histograms to compute distances, helping somewhat with the issue of choosing bin sizes.

Kernel density estimation Probably the most popular form of general-purpose nonparametric density estimation is kernel density estimation (KDE). KDE results in a mixture distribution, which allow $O(n^2)$ exact computation of plug-in MMD and L_2 for certain density kernels.

Singh and Póczos (2014) show exponential concentration for a particular plug-in estimator for a broad class of functionals including L_p, α-β and f-divergences as well as JS, though they don't discuss computational issues of the estimator, which in general requires numerical integration.

Krishnamurthy et al. (2014) correct a plug-in estimator for L_2 and R_α divergences by estimating higher order terms in the von Mises expansion; one of their estimators is computationally attractive and optimal for smooth distributions, while another is optimal for a broader range of distributions but requires numerical integration.

k-NN density estimator The k-NN density estimator provides the basis for another family of estimators. These estimators typically require k-nearest neighbor distances within and between the sample sets; much research has been put into data structures for approximate nearest neighbor computation (e.g. Beygelzimer et al. 2006; Muja and Lowe 2009; Andoni and Razenshteyn 2015), though in high dimensions the problem is quite difficult and brute-force pairwise computation may be the most efficient method. Plug-in methods require k to grow with sample size for consistency, which makes computation more difficult.

Q. Wang et al. (2009) give a simple, consistent k-NN KL divergence estimator. Póczos and Schneider (2011) give a similar estimator for $D_{\alpha-1,1-\alpha}$ and show consistency; Póczos, Xiong,

Sutherland, et al. (2012) generalize to $D_{\alpha,\beta}$. This family of estimators is consistent with a fixed k, though convergence rates are not known.

Moon and Hero (2014a,b) propose an f-divergence estimator based on ensembles of plug-in estimators, and show the distribution is asymptotically Gaussian. (Their estimator requires neither convex f nor $f(1) = 0$.)

Mean map estimators A natural estimator of MMK is simply the mean of the pairwise kernel evaluations between the two sets; this corresponds to estimating the mean embedding by the empirical mean of the embedding for each point (Gretton, Borgwardt, et al. 2012). More recently, Muandet, Fukumizu, et al. (2014) proposed biased estimators with smaller variance via the idea of Stein shrinkage (1956). Ramdas and Wehbe (2014) showed the efficacy of this approach for independence testing.

Other approaches Nguyen et al. (2010) provide an estimator for f-divergences (requiring convex f but not $f(1) = 0$) by solving a convex program. When an RKHS structure is imposed, it requires solving a general convex program with dimensionality equal to the number of samples, so the estimator is quite computationally expensive.

Sriperumbudur et al. (2012) estimate the L_1-EMD via a linear program.

K. Yang et al. (2014) estimate f- and R_α divergences by adaptively partitioning both distributions simultaneously. Their Bayesian approach requires MCMC and is computationally expensive, though it does provide a posterior over the divergence value which can be useful in some settings.

2.3 Kernels on distributions

We consider two methods for defining kernels based on distributional distances ρ. Proposition 1 of Haasdonk and Bahlmann (2004) shows that both methods always create positive definite kernels iff ρ is isometric to an L_2 norm, i.e. there exist a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that $\rho(P, Q) = \|\Phi(P) - \Phi(Q)\|$. Such metrics are also called *Hilbertian*.[2]

For distances that do not satisfy this property, we will instead construct an indefinite kernel as below and then "correct" it, as discussed in Section 2.4.1.

The first method is to create a "linear kernel" k such that $\rho(P, Q) = k(P, P)^2 + k(Q, Q)^2 - 2k(P, Q)$, so that the RKHS with inner product k has metric ρ. Note that, while distances are translation-invariant, inner products are not; we must thus first choose some origin O. Then

$$k_{\text{lin}}^{(O)}(P, Q) := \tfrac{1}{2}\left(\rho^2(P, 0) + \rho^2(Q, O) - \rho^2(P, Q)\right) \tag{2.1}$$

is a valid kernel for any O iff ρ is Hilbertian. If ρ is defined for the zero measure, it is often most natural to use that as the origin.

We can also use ρ in a *generalized RBF kernel*: for a bandwidth parameter $\sigma > 0$,

$$k_{\text{RBF}}^{(\sigma)}(x, y) := \exp\left(-\frac{1}{2\sigma^2}\rho^2(p, q)\right). \tag{2.2}$$

[2]Note that if ρ is Hilbertian, Proposition 1 (ii) of Haasdonk and Bahlmann (2004) shows that $-\rho^{2\beta}$ is conditionally positive definite for any $0 \le \beta \le 1$; by a classic result of Schoenberg (1938), this implies that ρ^β is also Hilbertian. We will use this fact later.

The L_2 distance is clearly Hilbertian; $k^{(0)}_{\text{lin}}(P, Q) = \int p(x)q(x)\,\mathrm{d}x$.

Fuglede (2005) shows that $\sqrt{\text{JS}}$, TV, and H are Hilbertian.

- For $\sqrt{\text{JS}}$, $k^{(O)}_{\text{lin}}(P, Q) = \frac{1}{2}\left(\text{H}\left[\frac{P+O}{2}\right] + \text{H}\left[\frac{Q+O}{2}\right] - \text{H}\left[\frac{P+Q}{2}\right] - \text{H}[O]\right)$.
- For TV, $k^{(0)}_{\text{lin}}(P, Q) = \frac{1}{4}\left(1 - \text{TV}(P, Q)\right)$.
- For H, $k^{(0)}_{\text{lin}}(P, Q) = 1 - \frac{1}{2}\,\text{H}^2(P, Q) = \frac{1}{2} + \int \sqrt{p(x)\,q(x)}\mathrm{d}x$, but the halved Bhattacharyya affinity $k(P, Q) = \frac{1}{2}\int \sqrt{p(x)\,q(x)}\mathrm{d}x$ is more natural.

Gardner et al. (2015) shows that EMD is Hilbertian for the 0-1 distance (an unusual choice of ground metric for EMD). EMD is probably not Hilbertian in most cases for Euclidean base distance: Naor and Schechtman (2007) show that EMD on distributions supported on a grid in $\mathbb{R}^2$ does not embed in L_1, which since L_2 embeds into L_1 (Bretagnolle et al. 1966) means that EMD on that grid does not embed in L_2. It is likely, though the details remain to be checked, that this also implies L_1-EMD on continuous distributions over $\mathbb{R}^d$ for $d \geq 2$ is not Hilbertian. The most common kernel based on EMD, however, is actually $\exp\left(-\gamma\,\text{EMD}(P, Q)\right)$. Whether that kernel is positive definite seems to remain an open question, defined by whether $\sqrt{\text{EMD}}$ is Hilbertian; studies that have used it in practice have not reported finding any instance of an indefinite kernel matrix (Zhang et al. 2006).

The MMD is Hilbertian by definition. The natural associated linear kernel is $k^{(0)}_{\text{lin}}(P, Q) = \langle \mu(P), \mu(Q) \rangle$, which we term the *mean map kernel* (MMK). (We can easily verify that this is a valid kernel inducing MMD despite some technical issues with considering it as $k^{(0)}_{\text{lin}}$.)

2.4 Kernels on sample sets

As discussed previously, in practice we rarely directly observe a probability distribution; rather, we observe samples from those distributions. We will instead construct a kernel on sample sets, based on an estimate of a kernel on distributions using an estimate of the base distance ρ.

We wish to estimate a kernel on N distributions $\{P_i\}_{i=1}^N$ based on an iid sample from each distribution $\{X^{(i)}\}_{i=1}^N$, where $X^{(i)} = \{X^{(i)}_j\}_{j=1}^{n_i}$, $X^{(i)}_j \in \mathbb{R}^d$. Given an estimator $\hat{\rho}(X^{(i)}, X^{(j)})$ of $\rho(P_i, P_j)$, we estimate $k(P_i, P_j)$ with $\hat{k}(X^{(i)}, X^{(j)})$ by substituting $\hat{\rho}(X^{(i)}, X^{(j)})$ for $\rho(P_i, P_j)$ in (2.1) or (2.2). We thus obtain an estimate $\hat{K}$ of the true kernel matrix K, where $\hat{K}_{i,j} = \hat{k}(X^{(i)}, X^{(j)})$.

2.4.1 Handling indefinite kernel matrices

Section 2.3 established that K is positive semidefinite for many distributional distances ρ, but for some, particularly SKL and SR, K is indefinite. Even if K is PSD, however, depending on the form of the estimator $\hat{K}$ is likely to be indefinite.

In this case, for many downstream learning tasks we must modify $\hat{K}$ to be positive semidefinite. Chen et al. (2009) study this setting, presenting four methods to make $\hat{K}$ PSD:

- Spectrum clip: Set any negative eigenvalues in the spectrum of $\hat{K}$ to zero. This yields the nearest PSD matrix to $\hat{K}$ in Frobenius norm, and corresponds to the view where negative eigenvalues are simply noise.
- Spectrum flip: Replace any negative eigenvalues in the spectrum with their absolute value.

- Spectrum shift: Increase each eigenvalue in the spectrum by the magnitude of the smallest eigenvalue, by taking $\hat{K} + |\lambda_{\min}|I$. When $|\lambda_{\min}|$ is small, this is computationally simpler – it is easier to find $\lambda_{\min}$ than to find all negative eigenvalues, and requires modifying only the diagonal elements — but can change $\hat{K}$ more drastically.
- Spectrum square: Square the eigenvalues, by using $\hat{K}\hat{K}^T$. This is equivalent to using the kernel estimates as features.

We denote this operation by Π.

When test values are available at training time, i.e. in a transductive setting, it is best to perform these operations on the full kernel matrix containing both training and test points: that is, to use $\Pi\left(\begin{bmatrix} \hat{K}_{\text{train}} & \hat{K}_{\text{train,test}} \\ \hat{K}_{\text{test,train}} & \hat{K}_{\text{test}} \end{bmatrix}\right)$. (Note that $\hat{K}_{\text{test}}$ is not actually used by e.g. an SVM.) If the changes are performed only on the training matrix, i.e. using $\begin{bmatrix} \Pi\left(\hat{K}_{\text{train}}\right) & \hat{K}_{\text{train,test}} \\ \hat{K}_{\text{test,train}} & \hat{K}_{\text{test}} \end{bmatrix}$, which is necessary in the typical inductive setting, the resulting full kernel matrix may not be PSD, and the kernel estimates may be treated inconsistently between training and test points. This is more of an issue for a truly-indefinite kernel, e.g. one based on KL or R, where the changes due to Π may be larger.

When the test values are not available, Chen et al. (2009) propose a heuristic to account for the effect of Π: they find the linear transformation which maps $\hat{K}_{\text{train}}$ to $\Pi\hat{K}_{\text{train}}$, based on the eigendecomposition of $\hat{K}_{\text{train}}$, and apply it to $\hat{K}_{\text{test,train}}$. We compare this method to the transductive method as well as the method where test evaluations are unaltered in experiments. In general, the transductive method is better than the heuristic approach, which is better than ignoring the problem, but the size of these gaps is problem-specific: for some problems, the gap is substantial, but for others it matters little.

When performing bandwidth selection for a generalized Gaussian RBF kernel, this approach requires separately eigendecomposing each $\hat{K}_{\text{train}}$. Xiong (2013, Chapter 6) considers a different solution: rank-penalized metric multidimensional scaling according to $\hat{\rho}$, so that standard Gaussian RBF kernels may be applied to the embedded points. That work does not consider the inductive setting, though an approach similar to that of Bengio et al. (2004) is probably applicable.

Chapter 3

Scalable distribution learning with approximate kernel embeddings

The kernel methods of Chapter 2 share a common drawback: solving learning problems with N distributions typically requires computing all or most of the $N \times N$ kernel matrix; further, many of the methods of Section 2.4.1 to deal with indefinite kernels require $O(N^3)$ eigendecompositions. For large N, this quickly becomes impractical.

Rahimi and Recht (2007) spurred recent interest in a method to avoid this: approximate embeddings $z : \mathcal{X} \to \mathbb{R}^D$ such that $k(x, y) \approx z(x)^\mathsf{T} z(y)$. Learning primal models in $\mathbb{R}^D$ using the z features can then usually be accomplished in time linear in n, with the models on z approximating the models on k.

This chapter first reviews the method of Rahimi and Recht (2007), providing some additional theoretical understanding, and then shows how to find embeddings z for various distributional kernels.

3.1 Random Fourier features

Rahimi and Recht (2007) considered continuous shift-invariant kernels on $\mathbb{R}^d$, i.e. those that can be written $k(x, y) = \underline{k}(\Delta)$, where we will use $\Delta := x - y$ throughout. In this case, Bochner's theorem (1959) guarantees that the Fourier transform $\Omega(\cdot)$ of $\underline{k}$ will be a nonnegative measure; if $\underline{k}(0) = 1$, it will be properly normalized. Thus if we define

$$\tilde{z}(x) := \sqrt{\frac{2}{D}} \begin{bmatrix} \sin(\omega_1^\mathsf{T} x) & \cos(\omega_1^\mathsf{T} x) & \dots & \sin(\omega_{D/2}^\mathsf{T} x) & \cos(\omega_{D/2}^\mathsf{T} x) \end{bmatrix}^\mathsf{T}, \ \{\omega_i\}_{i=1}^{D/2} \sim \Omega^{D/2}$$

and let $\tilde{s}(x, y) := \tilde{z}(x)^\mathsf{T} \tilde{z}(y)$, we have that

$$\tilde{s}(x, y) = \frac{2}{D} \sum_{i=1}^{D/2} \sin(\omega_i^\mathsf{T} x) \sin(\omega_i^\mathsf{T} y) + \cos(\omega_i^\mathsf{T} x) \cos(\omega_i^\mathsf{T} y) = \frac{1}{D/2} \sum_{i=1}^{D/2} \cos(\omega_i^\mathsf{T} \Delta).$$

Noting that $\mathbb{E} \cos(\omega^\mathsf{T} \Delta) = \int \Re e^{\omega^\mathsf{T} \Delta \mathrm{i}} \mathrm{d}\Omega(\omega) = \Re k(\Delta)$, we therefore have $\mathbb{E}\, \tilde{s}(x, y) = k(x, y)$.

Note that $\underline{k}$ is the characteristic function of Ω, and $\underline{\tilde{s}}$ the empirical characteristic function corresponding to the samples $\{\omega_i\}$.

Rahimi and Recht (2007) alternatively proposed

$$\check{z}(x) := \sqrt{\frac{2}{D}}\left[\cos(\omega_1^\mathsf{T} x + b_1) \quad \dots \quad \cos(\omega_D^\mathsf{T} x + b_D)\right]^\mathsf{T},\ \{\omega_i\}_{i=1}^D \sim \Omega^D,\ \{b_i\}_{i=1}^D \overset{iid}{\sim} \mathrm{Unif}_{[0,2\pi]}^D.$$

Letting $\check{s}(x,y) := \check{z}(x)^\mathsf{T}\check{z}(y)$, we have

$$\check{s}(x,y) = \frac{1}{D}\sum_{i=1}^D \cos(\omega_i^\mathsf{T} x + b_i)\cos(\omega_i^\mathsf{T} y + b_i) = \frac{1}{D}\sum_{i=1}^D \cos(\omega_i^\mathsf{T}(x-y)) + \cos(\omega_i^\mathsf{T}(x+y) + 2b_i).$$

Let $t := x + y$ throughout. Since $\mathbb{E}\cos(\omega^\mathsf{T} t + 2b) = \mathbb{E}_\omega\left[\mathbb{E}_b \cos(\omega^\mathsf{T} t + 2b)\right] = 0$, we also have $\mathbb{E}\,\check{s}(x,y) = k(x,y)$.

Thus, in expectation, both $\tilde{z}$ and $\check{z}$ work; they are each the average of bounded, independent terms with the correct mean. For a given embedding dimension, $\tilde{z}$ has half as many terms as $\check{z}$, but each of those terms has lower variance; which embedding is superior is, therefore, not obvious. We will answer this question, as well as giving uniform convergence bounds for each embedding.[1]

3.1.1 Reconstruction variance

We can in fact find the covariance of the reconstructions:

$$\begin{aligned}
\mathrm{Cov}\left(\underline{\tilde{s}}(\Delta), \underline{\tilde{s}}(\Delta')\right) &= \frac{2}{D}\mathrm{Cov}\left(\cos(\omega^\mathsf{T}\Delta), \cos(\omega^\mathsf{T}\Delta')\right) \\
&= \frac{1}{D}\left[\mathbb{E}\left[\cos\left(\omega^\mathsf{T}(\Delta - \Delta')\right) + \cos\left(\omega^\mathsf{T}(\Delta + \Delta')\right)\right] - 2\,\mathbb{E}\left[\cos\left(\omega^\mathsf{T}\Delta\right)\right]\mathbb{E}\left[\cos\left(\omega^\mathsf{T}\Delta\right)\right]\right] \\
&= \frac{1}{D}\left[\underline{k}(\Delta - \Delta') + \underline{k}(\Delta + \Delta') - 2\underline{k}(\Delta)\underline{k}(\Delta')\right],
\end{aligned}$$

so that

$$\mathrm{Var}\,\underline{\tilde{s}}(\Delta) = \frac{1}{D}\left[1 + \underline{k}(2\Delta) - 2\underline{k}(\Delta)^2\right].$$

Similarly,

$$\begin{aligned}
\mathrm{Cov}\left(\check{s}(x,y), \check{s}(x',y')\right) &= \frac{1}{D}\mathrm{Cov}\left(\cos(\omega^\mathsf{T}\Delta) + \cos(\omega^\mathsf{T} t + 2b), \cos(\omega^\mathsf{T}\Delta') + \cos(\omega^\mathsf{T} t' + 2b)\right) \\
&= \frac{1}{D}\Big[\mathrm{Cov}\left(\cos(\omega^\mathsf{T}\Delta), \cos(\omega^\mathsf{T}\Delta')\right) + \mathrm{Cov}\left(\cos(\omega^\mathsf{T} t + 2b), \cos(\omega^\mathsf{T} t' + 2b)\right) \\
&\qquad + \underbrace{\mathrm{Cov}\left(\cos(\omega^\mathsf{T}\Delta), \cos(\omega^\mathsf{T} t' + 2b)\right)}_{0} + \underbrace{\mathrm{Cov}\left(\cos(\omega^\mathsf{T} t + 2b), \cos(\omega^\mathsf{T}\Delta')\right)}_{0}\Big] \\
&= \frac{1}{D}\left[\tfrac{1}{2}\underline{k}(\Delta - \Delta') + \tfrac{1}{2}\underline{k}(\Delta + \Delta') - \underline{k}(\Delta)\underline{k}(\Delta') + \tfrac{1}{2}\underline{k}(t - t')\right],
\end{aligned}$$

[1] Most of the remainder of Section 3.1 is based on Sutherland and Schneider (2015).

and so

$$\operatorname{Var} \check{s}(x, y) = \frac{1}{D}\left[1 + \tfrac{1}{2}\underline{k}(2\Delta) - \underline{k}(\Delta)^2\right].$$

Thus $\tilde{s}$ has lower variance than $\check{s}$ when $\underline{k}(2\Delta) < 2\underline{k}(\Delta)^2$, i.e.

$$\operatorname{Var} \cos(\omega^\mathsf{T}\Delta) = \frac{1}{2} + \frac{1}{2}\underline{k}(2\Delta) - \underline{k}(\Delta)^2 \leq \frac{1}{2}.$$

Exponentiated norms Consider a kernel of the form $\underline{k}(\Delta) = \exp(-\gamma\|\Delta\|^\beta)$ for any norm and some $\beta \geq 1$. For example, the Gaussian kernel uses $\|\cdot\|_2$ and $\beta = 2$, and the Laplacian kernel uses $\|\cdot\|_1$ and $\beta = 1$. Then

$$\begin{aligned}
2\underline{k}(\Delta)^2 - \underline{k}(2\Delta) &= 2\exp\left(-\gamma\|\Delta\|^\beta\right)^2 - \exp\left(-\gamma\|2\Delta\|^\beta\right) \\
&= 2\exp\left(-2\gamma\|\Delta\|^\beta\right) - \exp\left(-2^\beta\gamma\|\Delta\|^\beta\right) \\
&\geq 2\exp\left(-2\gamma\|\Delta\|^\beta\right) - \exp\left(-2\gamma\|\Delta\|^\beta\right) = \exp\left(-2\gamma\|\Delta\|^\beta\right) > 0.
\end{aligned}$$

Matérn kernel The Matérn kernel for half-integer orders also yields $\tilde{s}$ uniformly lower-variance than $\check{s}$. The kernel has two hyperparameters, a length-scale ℓ and an order ν. If we let $r \coloneqq \frac{1}{\ell}\|\Delta\|$ and $\nu = \eta + \frac{1}{2}$ for η a nonnegative integer; then the kernel can be written (Rasmussen and Williams 2006, equation 4.16):

$$\underline{k}(r) = \exp\left(-\sqrt{2\eta+1}r\right)\sum_{i=0}^{\eta}\underbrace{\frac{\eta!\,(2\eta-i)!}{(2\eta)!\,i!\,(\eta-i)!}\left(2\sqrt{2\eta+1}\right)^i}_{a_i} r^i.$$

Then we have

$$\begin{aligned}
2\underline{k}(r)^2 - \underline{k}(2r) &= 2\exp\left(-\sqrt{2\eta+1}r\right)^2\sum_{i=0}^{\eta}\sum_{j=0}^{\eta}a_i a_j r^{i+j} - \exp\left(-2\sqrt{2\eta+1}r\right)\sum_{i=0}^{\eta}a_i(2r)^i \\
&= \exp\left(-2\sqrt{2\eta+1}r\right)\left(2\sum_{m=0}^{2\eta}\sum_{i=0}^{m}a_i a_{m-i}r^m - \sum_{m=0}^{\eta}2^m a_m r^m\right) \\
&\geq \exp\left(-2\sqrt{2\eta+1}r\right)\sum_{m=0}^{\eta}\left(2\sum_{i=0}^{m}\frac{a_i a_{m-i}}{a_m} - 2^m\right)a_m r^m.
\end{aligned}$$

Now,

$$\begin{aligned}
\sum_{i=0}^{m}\frac{a_i a_{m-i}}{a_m} &= \sum_{i=0}^{m}\frac{m!}{i!\,(m-i)!}\frac{\eta!}{(\eta-i)!}\frac{(\eta-m)!}{(\eta-m+i)!}\frac{(2\eta-m+i)!}{(2\eta-m)!}\frac{(2\eta-i)!}{(2\eta)!} \\
&= \sum_{i=0}^{m}\binom{m}{i}\prod_{j=1}^{i}\frac{\eta-i+j}{\eta-m+j}\frac{2\eta-m+j}{2\eta-i+j} \geq \sum_{i=0}^{m}\binom{m}{i} = 2^m
\end{aligned}$$

because, since $\eta - i + j \geq \eta - m + j$, each factor in the product is at least 1. Thus $2\underline{k}(r)^2 > \underline{k}(2r)$.

3.1.2 Convergence bounds

Let $\tilde{f}(x,y) \coloneqq \tilde{s}(x,y) - k(x,y)$, and $\breve{f}(x,y) \coloneqq \breve{s}(x,y) - k(x,y)$. We know that $\mathbb{E}\, f(x,y) = 0$ and have a closed form for $\operatorname{Var} f(x,y)$, but to better understand the error behavior across inputs, we wish to bound $\|f\|$ for various norms.

L_2 bound If μ is a finite measure on $\mathcal{X} \times \mathcal{X}$ ($\mu(\mathcal{X}^2) < \infty$), the $L_2(\mathcal{X}^2, \mu)$ norm of f is

$$\|f\|_\mu^2 \coloneqq \int_{\mathcal{X}^2} f(x,y)^2 \,\mathrm{d}\mu(x,y).$$

We know (via Tonelli's theorem) that

$$\begin{aligned}
\mathbb{E}\|\tilde{f}\|_\mu^2 &= \int_{\mathcal{X}^2} \mathbb{E}\,\tilde{f}(x,y)^2 \,\mathrm{d}\mu(x,y) \\
&= \frac{1}{D} \int_{\mathcal{X}^2} \left[1 + k(2x,2y) - 2k(x,y)^2\right] \mathrm{d}\mu(x,y) \\
\mathbb{E}\|\tilde{f}\|_\mu^2 &= \frac{1}{D} \int_{\mathcal{X}^2} \left[1 + \tfrac{1}{2}k(2x,2y) - k(x,y)^2\right] \mathrm{d}\mu(x,y)
\end{aligned}$$

so that, for the kernels considered above, the expected $L_2(\mu)$ error for $\tilde{z}$ is less than that of $\breve{z}$. Note that if $\mu = \mu_X \times \mu_Y$ is a joint distribution of independent variables, then

$$\begin{aligned}
\mathbb{E}\|\tilde{f}\|_\mu^2 &= \frac{1}{D}\left[1 + \operatorname{MMK}_k(\mu_{2X}, \mu_{2Y}) - 2\operatorname{MMK}_{k^2}(\mu_X, \mu_y)\right] \\
\mathbb{E}\|\breve{f}\|_\mu^2 &= \frac{1}{D}\left[1 + \tfrac{1}{2}\operatorname{MMK}_k(\mu_{2X}, \mu_{2Y}) - \operatorname{MMK}_{k^2}(\mu_X, \mu_y)\right].
\end{aligned}$$

Propositions 7 and 8 of Sutherland and Schneider (2015) further bound the deviation from this expectation via McDiarmid's inequality:

$$\begin{aligned}
\Pr\left(\left|\|\tilde{f}\|_\mu^2 - \mathbb{E}\|\tilde{f}\|_\mu^2\right| \geq \varepsilon\right) &\leq 2\exp\left(\frac{-D^3\varepsilon^2}{8(4D+1)^2\mu(\mathcal{X}^2)^2}\right) \leq 2\exp\left(\frac{-D\varepsilon^2}{200\mu(\mathcal{X}^2)^2}\right) \\
\Pr\left(\left|\|\breve{f}\|_\mu^2 - \mathbb{E}\|\breve{f}\|_\mu^2\right| \geq \varepsilon\right) &\leq 2\exp\left(\frac{-D^3\varepsilon^2}{512(D+1)^2\mu(\mathcal{X}^2)^2}\right) \leq 2\exp\left(\frac{-D\varepsilon^2}{2048\mu(\mathcal{X}^2)^2}\right).
\end{aligned}$$

Sriperumbudur and Szabó (2015) independently bounded the deviation of f in the L_r norm for any $r \in [1, \infty)$ but only for μ the Lebesgue measure.

Uniform bound Rahimi and Recht (2007) showed a uniform convergence bound for $\tilde{s}$. Propositions 1 and 2 of Sutherland and Schneider (2015) tightened that bound, and showed an analogous one for $\breve{s}$, which we reproduce here.

When $\nabla^2 k(0)$ exists and $\mathcal{X} \subset \mathbb{R}^d$ is compact with diameter ℓ, let $\sigma_\Omega^2 \coloneqq \mathbb{E}_\Omega\|\omega\|^2$ and

$$\alpha_\varepsilon \coloneqq \min\left(1, \sup_{x,y\in\mathcal{X}} \tfrac{1}{2} + \tfrac{1}{2}k(2x,2y) - k(x,y)^2 + \tfrac{1}{3}\varepsilon\right), \quad \beta_d \coloneqq \left(\left(\tfrac{d}{2}\right)^{-\frac{d}{d+2}} + \left(\tfrac{d}{2}\right)^{\frac{2}{d+2}}\right) 2^{\frac{6d+2}{d+2}}.$$

Then, assuming only for the second statement that $\varepsilon \le \sigma_p \ell$,

$$\Pr\left(\|\tilde{f}\|_\infty \ge \varepsilon\right) \le \beta_d \left(\frac{\sigma_p \ell}{\varepsilon}\right)^{\frac{2}{1+\frac{2}{d}}} \exp\left(-\frac{D\varepsilon^2}{8(d+2)\alpha_\varepsilon}\right) \le 66\left(\frac{\sigma_p \ell}{\varepsilon}\right)^2 \exp\left(-\frac{D\varepsilon^2}{8(d+2)}\right).$$

For $\breve{f}$, define

$$\alpha'_\varepsilon := \min\left(1, \sup_{x,y\in\mathcal{X}} \tfrac{1}{4} + \tfrac{1}{8}k(2x,2y) - \tfrac{1}{4}k(x,y)^2 + \tfrac{1}{6}\varepsilon\right), \quad \beta'_d := \left(d^{-\frac{d}{d+1}} + d^{\frac{1}{d+1}}\right) 2^{\frac{5d+1}{d+1}} 3^{\frac{d}{d+1}}.$$

Then, again assuming only for the second statement that $\varepsilon \le \sigma_p \ell$,

$$\Pr\left(\|\tilde{f}\|_\infty \ge \varepsilon\right) \le \beta'_d \left(\frac{\sigma_p \ell}{\varepsilon}\right)^{\frac{2}{1+\frac{1}{d}}} \exp\left(-\frac{D\varepsilon^2}{32(d+1)\alpha'_\varepsilon}\right) \le 98\left(\frac{\sigma_p \ell}{\varepsilon}\right)^2 \exp\left(-\frac{D\varepsilon^2}{32(d+1)}\right).$$

For the kernels for which $\underline{k}(2\Delta) < 2\underline{k}(\Delta)^2$, note that $\alpha_\varepsilon \le \frac{1}{2} + \frac{1}{3}\varepsilon$ and $\alpha'_\varepsilon \le \frac{1}{4} + \frac{1}{6}\varepsilon$.

Propositions 3–4 of Sutherland and Schneider (2015) give an upper bound on $\mathbb{E}\|f\|_\infty$, and Propositions 5–6 bound $\Pr\left(\|f\|_\infty - \mathbb{E}\|f\|_\infty \ge \varepsilon\right)$. The former bound is quite loose in practice; the latter, when used with the true value of the expectation, is sometimes tighter than the previous bounds and sometimes not.

Sriperumbudur and Szabó (2015) later proved a rate-optimal $O_P(n^{-1/2})$ bound on $\|\tilde{f}\|_\infty$; in practice, the constants are often worse than the non-optimal bound above.

3.2 Mean map kernels

Armed with an approximate embedding for shift-invariant kernels on $\mathbb{R}^d$, we now develop our first embedding for a distributional kernel, MMK. Recall that, given samples $\{X_i\}_{i=1}^n \sim P^n$ and $\{Y_j\}_{j=1}^m \sim Q^m$, $\text{MMK}(P,Q)$ can be estimated as

$$\text{MMK}(X,Y) = \frac{1}{nm}\sum_{i=1}^n \sum_{j=1}^m k(X_i, Y_j).$$

Simply plugging in an approximate embedding $z(x)^\mathsf{T} z(y) \approx k(x,y)$ yields

$$\text{MMK}(X,Y) \approx \frac{1}{nm}\sum_{i=1}^n \sum_{j=1}^m z(X_i)^\mathsf{T} z(Y_j) = \left[\frac{1}{n}\sum_{i=1}^n z(X_i)\right]^\mathsf{T} \left[\frac{1}{m}\sum_{j=1}^m z(Y_j)\right] = \bar{z}(X)^\mathsf{T}\bar{z}(Y),$$

where we defined $\bar{z}(X) := \frac{1}{n}\sum_{i=1}^n z(X_i)$. This additionally has a natural interpretation as the direct estimate of MMD in the Hilbert space induced by the feature map z, which approximates the Hilbert space associated with k.

Note that $e^{-\gamma\,\text{MMD}^2}$ can be approximately embedded with $z(\bar{z}(\cdot))$.

This natural approximation, or its equivalents, have been considered many times quite recently (Mehta and Gray 2010; Li and Tsang 2011; Zhao and Meng 2014; Chwialkowski et al. 2015; Flaxman et al. 2015; Jitkrittum et al. 2015; Lopez-Paz et al. 2015; Sutherland, J. B. Oliva, et al. 2015; Sutherland and Schneider 2015).

3.3 L_2 distances

J. B. Oliva, Neiswanger, et al. (2014) gave an embedding for $e^{-\gamma L_2^2}$, by first embedding L_2 with orthonormal projections and then applying random Fourier features.

Suppose that $\mathcal{X} \subseteq [0,1]^d$. Let $\{\varphi_i\}_{i\in\mathbb{Z}}$ be an orthonormal basis for $L_2([0,1])$. Then, we can construct an orthonormal basis for $L_2([0,1]^d)$ by the tensor product: letting $\varphi_\alpha(x) \coloneqq \prod_{i=1}^d \varphi_{\alpha_i}(x_i)$, $\{\varphi_\alpha\}_{\alpha\in\mathbb{Z}^d}$ is an orthonormal basis, and any function $f \in L_2([0,1])$ can be represented as $f(x) = \sum_{\alpha\in\mathbb{Z}^d} a_\alpha(f)\varphi_\alpha(x)$, where $a_\alpha(f) = \langle\varphi_\alpha, f\rangle = \int_{[0,1]^d} \varphi_\alpha(t)f(t)\,\mathrm{d}t$. Thus for any $f, g \in L_2([0,1]^d)$,

$$\begin{aligned}\langle f, g\rangle &= \left\langle \sum_{\alpha\in\mathbb{Z}^d} a_\alpha(f)\varphi_\alpha, \sum_{\beta\in\mathbb{Z}^d} a_\beta(g)\varphi_\beta \right\rangle \\ &= \sum_{\alpha\in\mathbb{Z}^d}\sum_{\beta\in\mathbb{Z}^d} a_\alpha(f)a_\beta(g)\langle\varphi_\alpha,\varphi_\beta\rangle \\ &= \sum_{\alpha\in\mathbb{Z}^d} a_\alpha(f)a_\alpha(g)\end{aligned}$$

Let $V \subset \mathbb{Z}^d$ be an appropriately chosen finite set of indices $\{\alpha_1, \ldots, \alpha_{|V|}\}$. Define $\vec{a}(f) = (a_{\alpha_1}(f), \ldots, a_{\alpha_{|V|}}(f))^\mathsf{T} \in \mathbb{R}^{|V|}$. If f and g are smooth with respect to V, i.e. they have only small contributions from basis functions not in V, we have

$$\langle f, g\rangle = \sum_{\alpha\in\mathbb{Z}^d} a_\alpha(f)\, a_\alpha(g) \approx \sum_{\alpha\in V} a_\alpha(f)\, a_\alpha(g) = \vec{a}(f)^\mathsf{T}\, \vec{a}(g).$$

Now, given a sample $X = \{X_1, \ldots, X_n\} \sim P^n$, let $\hat{P}(x) = \frac{1}{n}\sum_{i=1}^n \delta(X_i - x)$ be the empirical distribution of X. J. B. Oliva, Neiswanger, et al. (2014) estimate the density p as

$$\hat{p}(x) = \sum_{\alpha\in V} a_\alpha(\hat{P})\,\varphi_\alpha(x) \qquad \text{where } a_\alpha(\hat{P}) = \int_{[0,1]^d} \varphi_\alpha(t)\,\mathrm{d}\hat{P}(t) = \frac{1}{n}\sum_{i=1}^n \varphi_\alpha(t).$$

Note that technically this is an extension of a_α to a broader domain than $L_2([0,1])$. Assuming the distributions lie in a certain Sobolev ellipsiod with respect to V, we thus have that

$$\langle p, q\rangle \approx \langle \hat{p}, \hat{q}\rangle \approx \vec{a}(\hat{P})^\mathsf{T}\, \vec{a}(\hat{Q})$$

and so

$$z(\vec{a}(\hat{P}))^\mathsf{T} z(\vec{a}(\hat{Q})) \approx \exp\left(-\frac{1}{2\sigma^2}\|P - Q\|_2^2\right).$$

For the Sobolev assumption to hold on a fairly general class of distributions, however, we need $|V|$ to be $\Omega(T^d)$ for some constant T. Thus this method is limited in practice to fairly low dimensions d.

3.3.1 Direct random Fourier features with Gaussian processes

When we apply $z(\vec{a}(f))$ for the Gaussian kernel with bandwidth σ, we draw $\omega \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^{-2} I\right)$ and then take trigonometric functions of multiples of

$$\omega^\mathsf{T} \vec{a}(f) = \sum_\alpha \omega_\alpha a_\alpha(f) = \left\langle \sum_\alpha \omega_\alpha \varphi_\alpha(\cdot), \sum_\beta \alpha_\beta(f) \varphi_\beta(\cdot) \right\rangle = \langle g, f \rangle,$$

where $g(\cdot) \coloneqq \sum_\alpha \omega_\alpha \varphi_\alpha(\cdot)$ is a random function distributed according to a process G. G is in fact a Gaussian process: any m points have the distribution

$$\begin{bmatrix} G(x_1) \\ \vdots \\ G(x_m) \end{bmatrix} = \begin{bmatrix} \varphi_1(x_1) & \cdots & \varphi_{|V|}(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(x_m) & \cdots & \varphi_{|V|}(x_m) \end{bmatrix} \omega \sim \mathcal{N}\left(0, \frac{1}{\sigma^2} \left[\sum_\alpha \varphi_\alpha(x_i) \varphi_\alpha(x_j) \right]_{ij} \right).$$

Thus, we can avoid explicitly computing $\vec{a}(f)$ — which typically grows exponentially in d — if we can otherwise compute $\langle g, f \rangle$. When $f = \hat{p}$ as above, this is simply $\frac{1}{n} \sum_{i=1}^n g(X_i)$. Thus for each dimension of z, we need to sample from a GP at every point contained in any of the samples X.

Doing so with the standard GP machinery requires quadratic storage and cubic computation in the total number of points contained in any sample, which is typically unacceptable in our setting: the only case in which it would be reasonable is if the distributions are defined only over a relatively small number of distinct points, in which case we should probably be using histogram techniques anyway. If the set of observed points happen to lie on a d-dimensional lattice, we could use the fast Kronecker inference techniques of Saatçi (2011); however, as these techniques require a *full* grid, it is more practical than simply computing $\vec{a}(f)$ only in very special cases.

Suppose, however, there exists some $\psi : \mathcal{X} \to \mathbb{R}^B$ such that $\psi(x)^\mathsf{T} \psi(y) \approx \mathrm{Cov}(G(x), G(y))$, and let $\Psi = \begin{bmatrix} \psi(x_1) & \cdots & \psi(x_m) \end{bmatrix} \in \mathbb{R}^{B \times m}$. Then we can just sample $b \sim \mathcal{N}(0, I_B)$ and use $\begin{bmatrix} G(x_1) & \cdots & G(x_m) \end{bmatrix}^\mathsf{T} = \Psi^\mathsf{T} b \sim \mathcal{N}(0, \Psi^\mathsf{T} \Psi)$, which can be computed on-demand as $G(x) = \psi(x)^\mathsf{T} b$. Not counting the evaluation of ψ, this takes $O(A)$ storage and $O(mA)$ time. Regression with this technique is known as *sparse spectrum Gaussian process regression* (Lázaro-Gredilla et al. 2010).

Suppose that $V = \{1, \ldots, T\}^d$. Then

$$\mathrm{Cov}(G(x), G(y)) = \sigma^{-2} \sum_{\alpha \in \{1, \ldots, T\}^d} \prod_{i=1}^d \varphi_{\alpha_i}(x_i) \varphi_{\alpha_i}(y_i) = \sigma^{-2} \prod_{i=1}^d \sum_{j=1}^T \varphi_j(x_i) \varphi_j(y_i).$$

Further suppose that $\{\varphi_i\}$ is the trigonometric basis for $L_2([0, 1])$:

$$\varphi_1(x) = 1 \qquad \varphi_{2m}(x) = \sqrt{2} \cos(2\pi\, m\, x) \qquad \varphi_{2m+1}(x) = \sqrt{2} \sin(2\pi\, m\, x)$$

and that T is odd. Then $\kappa(x_i, y_i) \coloneqq \sum_{j=1}^T \varphi_j(x_i) \varphi_j(y_i) = \frac{\sin(T\pi(x_i - y_i))}{\sin(\pi(x_i - y_i))}$ is known as the *Dirichlet kernel* and can be evaluated in $O(1)$. It also has a simple spectral representation: $\frac{1}{T} \underline{\kappa}(0) = 1$, and $\frac{1}{T} \underline{\kappa}$ has Fourier transform

$$\xi \sim \mathrm{Unif}\left(\{-\pi(T-1), -\pi(T-3), \ldots, \pi(T-3), \pi(T-1)\}\right). \tag{3.1}$$

Ψ is thus obtained via random Fourier features with each dimension of ξ independently distributed as (3.1).

If the approximation via Ψ is sufficient with $A \ll T^d$, this allows us to scale the L_2 embedding to higher dimensions.

3.4 Information-theoretic distances

We will now show how to extend this general approach to a class of information theoretic distances that includes TV, JS, and squared Hellinger.[2]

We consider a class of metrics that we term *homogeneous density distances* (HDDS):

$$\rho^2(p, q) = \int_{[0,1]^d} \kappa(p(x), q(x))\, \mathrm{d}x$$

where $\kappa : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is a 1-homogenous negative-type kernel. That is, $\kappa(tx, ty) = t\kappa(x, y)$ for all $t > 0$, and there exists some Hilbert space with $\|x - y\|^2 = \kappa(x, y)$. Table 3.1 shows some important instances.

Name	$\kappa(p(x), q(x))$	$\mathrm{d}\mu(\lambda)$
Jensen-Shannon (JS)	$\frac{p(x)}{2} \log\left(\frac{2p(x)}{p(x)+q(x)}\right) + \frac{q(x)}{2} \log\left(\frac{2q(x)}{p(x)+q(x)}\right)$	$\frac{\mathrm{d}\lambda}{\cosh(\pi\lambda)(1+\lambda^2)}$
Squared Hellinger (H^2)	$\frac{1}{2}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2$	$\frac{1}{2}\delta(\lambda = 1)\mathrm{d}\lambda$
Total Variation (TV)	$\lvert p(x) - q(x)\rvert$	$\frac{2}{\pi}\frac{\mathrm{d}\lambda}{1+4\lambda^2}$

Table 3.1: Various squared HDDs.

Vedaldi and Zisserman (2012) studied embeddings of a similar class of kernels, also using the key result of Fuglede (2005), but for discrete distributions only.

Fuglede (2005) shows that κ corresponds to a bounded measure $\mu(\lambda)$ by

$$\kappa(x, y) = \int_{\mathbb{R}_{\geq 0}} |x^{\frac{1}{2}+\mathrm{i}\lambda} - y^{\frac{1}{2}+\mathrm{i}\lambda}|^2 \,\mathrm{d}\mu(\lambda).$$

Let $Z \coloneqq \mu(\mathbb{R}_{\geq 0})$ and $c_\lambda \coloneqq (-\frac{1}{2} + \mathrm{i}\lambda)/(\frac{1}{2} + \mathrm{i}\lambda)$; then

$$\kappa(x, y) = \mathbb{E}_{\lambda \sim \frac{\mu}{Z}} |g_\lambda(x) - g_\lambda(y)|^2 \qquad \text{where } g_\lambda(x) \coloneqq \sqrt{Z} c_\lambda \left(x^{\frac{1}{2}+\mathrm{i}\lambda} - 1\right).$$

We approximate the expectation with an empirical mean. Let $\lambda_j \overset{iid}{\sim} \frac{\mu}{Z}$ for $j \in \{1, \ldots, M\}$; then

$$\kappa(x, y) \approx \frac{1}{M} \sum_{j=1}^{M} |g_{\lambda_j}(x) - g_{\lambda_j}(y)|^2.$$

[2]The method of Section 3.4 is currently in submission as Sutherland, J. B. Oliva, et al. (2015).

Hence, the squared HDD is, letting $\Re, \Im$ denote the real and imaginary parts:

$$\begin{aligned}
\rho^2(p,q) &= \int_{[0,1]^d} \kappa(p(x), q(x))\,\mathrm{d}x \\
&= \int_{[0,1]^d} \mathbb{E}_{\lambda\sim\frac{\mu}{Z}} |g_\lambda(p(x)) - g_\lambda(q(x))|^2\,\mathrm{d}x \\
&\approx \frac{1}{M}\sum_{j=1}^{M} \int_{[0,1]^d} \left(\left(\Re(g_{\lambda_j}(p(x))) - \Re(g_{\lambda_j}(q(x)))\right)^2 + \left(\Im(g_{\lambda_j}(p(x))) - \Im(g_{\lambda_j}(q(x)))\right)^2\right)\mathrm{d}x \\
&= \frac{1}{M}\sum_{j=1}^{M} \|p^R_{\lambda_j} - q^R_{\lambda_j}\|^2 + \|p^I_{\lambda_j} - q^I_{\lambda_j}\|^2,
\end{aligned}$$

where

$$p^R_\lambda(x) \coloneqq \Re(g_\lambda(p(x))), \qquad p^I_\lambda(x) \coloneqq \Im(g_\lambda(p(x))).$$

Each p_λ function is in $L_2([0,1]^d)$, so we can approximate $e^{-\gamma\rho^2(p,q)}$ as in Section 3.3: let

$$A(P) \coloneqq \frac{1}{\sqrt{M}}\left(\vec{a}(p^R_{\lambda_1})^\mathsf{T}, \vec{a}(p^I_{\lambda_1})^\mathsf{T}, \ldots, \vec{a}(p^R_{\lambda_M})^\mathsf{T}, \vec{a}(p^I_{\lambda_M})^\mathsf{T}\right)^\mathsf{T}$$

so that the kernel is estimated by $z(A(P))$.

However, the projection coefficients of the p_λ functions do not have simple forms as before; instead, we directly estimate the density as $\hat{p}$ using a technique such as kernel density estimation (KDE), and then estimate $\vec{a}(\hat{p}_\lambda)$ for each λ with numerical integration. Denote the estimated features as $\hat{A}(\hat{p})$.

For small d, simple Monte Carlo integration is sufficient.

In higher dimensions, three problems arise: (i) the embedding dimension increases exponentially, (ii) density estimation becomes statistically difficult, and (iii) accurate numerical integration becomes expensive. We can attempt to address (i) with the approach of Section 3.3.1, (ii) with sparse nonparametric graphical models (Lafferty et al. 2012), and (iii) with MCMC integration. High-dimensional multimodal integrals remain particularly challenging to current MCMC techniques, though some progress is being made (Betancourt 2015; Lan et al. 2014 give a heuristic algorithm).

Sutherland, J. B. Oliva, et al. (2015) bound the error probability for this estimator for a pair of distributions P, Q satisfying certain smoothness properties.

Chapter 4

Applications of distribution learning

We now turn to case studies of the application of distributional kernels to real machine learning tasks.

4.1 Mixture estimation

Statistical inference procedures can be viewed as functions from distributions to the reals; we can therefore consider learning such procedures. Jitkrittum et al. (2015) trained MMD-based GP regression for the messages computed by numerical integration in an expectation propagation system, and saw substantial speedups by doing so. We, inspired by J. B. Oliva, Neiswanger, et al. (2014), consider a problem where we not only obtain speedups over traditional algorithms, but actually see far superior results.[1] Specifically, we consider predicting the number of components in a Gaussian mixture. We generate mixtures as follows:

1. Draw the number of components Y_i for the ith distribution as $Y_i \sim \text{Unif}\{1, \dots, 10\}$.
2. For each component, select a mean $\mu_k^{(i)} \sim \text{Unif}[-5, 5]^2$ and covariance $\Sigma_k^{(i)} = a_k^{(i)} A_k^{(i)} A_k^{(i)\mathsf{T}} + B_k^{(i)}$, where $a \sim \text{Unif}[1, 4]$, $A_k^{(i)}(u, v) \sim \text{Unif}[-1, 1]$, and $B_k^{(i)}$ is a diagonal 2×2 matrix with $B_k^{(i)}(u, u) \sim \text{Unif}[0, 1]$.
3. Draw a sample $X^{(i)}$ from the equally-weighted mixture of these components.

An example distribution and sample from it is shown in Figure 4.1; predicting the number of components is difficult even for humans.

We compare generalized RBF kernels based on the MMD, L_2, and HDD embeddings of Sections 3.2 to 3.4 as well as the JS embedding of Vedaldi and Zisserman (2012) and the full Gram matrix techniques of Section 2.4 applied to the SKL estimator of Q. Wang et al. (2009).

Figure 4.2 presents results for predicting with ridge regression the number of mixture components Y_i, given a varying number of sample sets χ_i, with $|\chi_i| \in \{200, 800\}$; we use $D = 5\,000$. The HDD-based kernels achieve substantially lower error than the L_2 and MMD kernels in both cases. They also outperform the histogram kernels, especially with $|\chi_i| = 200$, and the KL kernel. Note that fitting mixtures with EM and selecting a number of components using AIC (Akiake 1973) or BIC (Schwarz 1978) performed much worse than regression; only AIC with $|\chi_i| = 800$

[1] These results are from Sutherland, J. B. Oliva, et al. (2015).

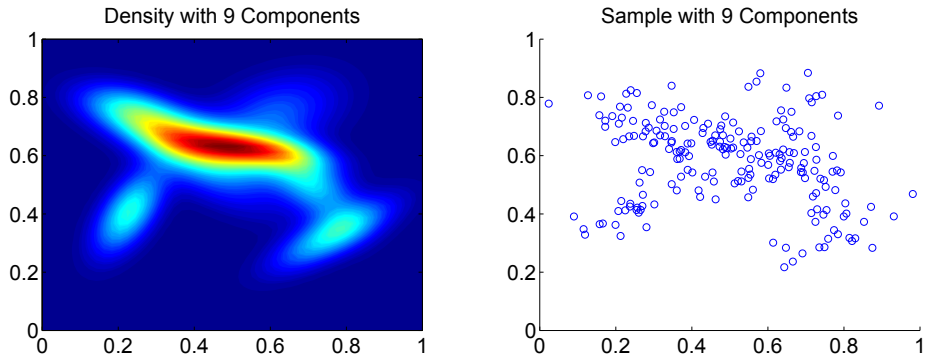
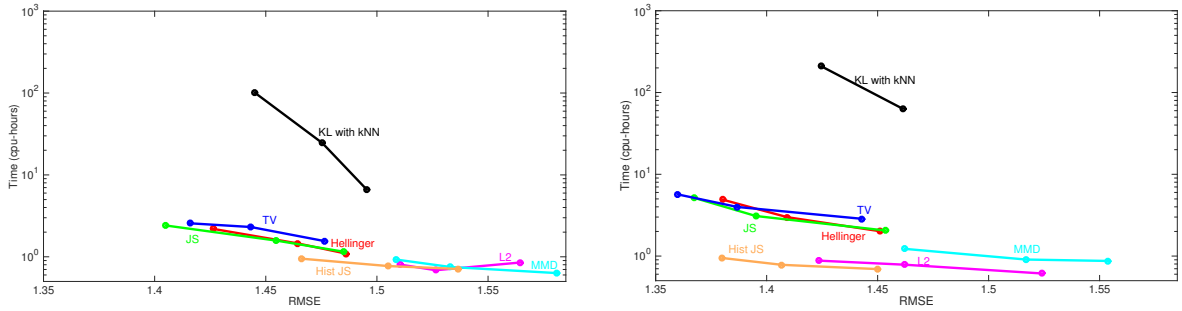


Figure 4.1: Example of a mixture with 9 components and a sample of size $n = 200$ drawn from it.



(a) Samples of size 200.

(b) Samples of size 800.

Figure 4.2: Error and computation time for estimating the number of mixture components. The three points on each line correspond to training set sizes of 4K, 8K, and 16K; error is on the fixed test set of size 2K. Note the logarithmic scale on the time axis. The KL kernel for sets of size 800 and 16K training sets was too slow to run. AIC-based predictions achieved RMSEs of 2.7 (for 200 samples) and 2.3 (for 800); BIC errors were 3.8 and 2.7; a constant predictor of 5.5 had RMSE of 2.8.

outperformed a constant predictor of 5.5. Linear versions of the L_2 and MMD kernels were also no better than the constant predictor.

The HDD embeddings were more computationally expensive than the other embeddings, but much less expensive than the KL kernel, which grows at least quadratically in the number of distributions. Note that the histogram embeddings used an optimized C implementation by the paper's authors (Vedaldi and Fulkerson 2008), and the KL kernel used the fairly optimized implementation of `gestalt-learn`, whereas the HDD embeddings used a simple Matlab implementation.

4.2 Scene recognition

Representing images as a collection of local patches has a long and successful history in computer vision.

4.2.1 SIFT features

The traditional approach selects a grid of patches, computes a hand-designed feature vector such as SIFT (Lowe 2004) for each patch, possibly appends information about the location of the patch, and then uses the BOW representation for this set of features. We will first consider the use of distributional distance kernels for this feature representation.[2]

We present here results on the 8-class OT scene recognition dataset (A. Oliva and Torralba 2001); the original papers show results on additional image datasets. This dataset contains 8 outdoor scene categories, illustrated in Figure 4.3. There are 2 688 total images, each about 256×256 pixels.

Figure 4.3: The 8 OT categories: *coast, forest, highway, inside city, mountain, open country, street, tall building*.

We extracted dense color SIFT features (Anna Bosch et al. 2008) at six different bin sizes using VLFEAT (Vedaldi and Fulkerson 2008), resulting in about 1 815 feature vectors per image, each of dimension 384. We used PCA to reduce these to 53 dimensions, preserving 70% of the variance, appended relative y coordinates, and standardized each dimension. (The paper contains precise details.)

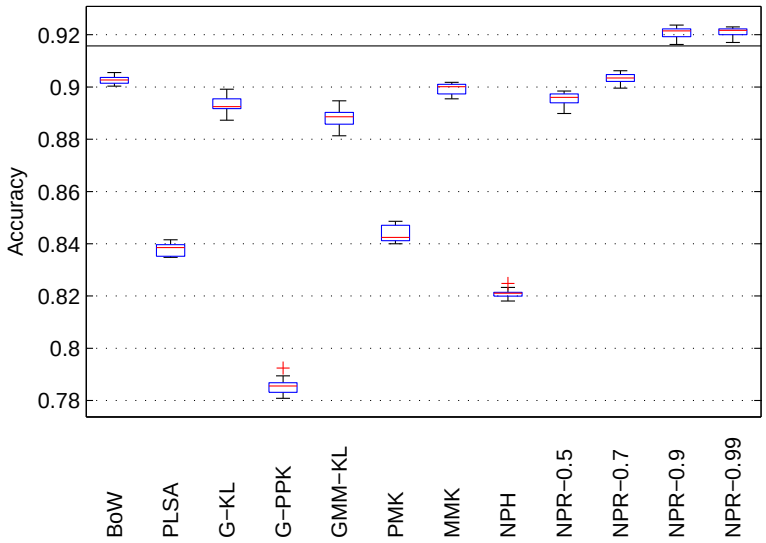


Figure 4.4: Accuracies on the OT dataset.

The results of 10 repeats of 10-fold cross-validation are shown in Figure 4.4. Each approach uses a generalized RBF kernel. Here BOW refers to vector quantization with k-means ($k = 1\,000$),

[2]These results appear in Póczos, Xiong, Sutherland, et al. (2012) and Sutherland, Xiong, et al. (2012).

PLSA to the approach of A. Bosch et al. (2006), G-KL and G-PPK to the KL and Hellinger divergences between Gaussians fit to the data, GMM-KL to the KL between Gaussian mixtures (computing via Monte Carlo), PMK to the pyramid matching kernel of Grauman and Darrell (2007), MMK to the MMK with a Gaussian base kernel, NPH to the nonparametric Hellinger estimate of Póczos, Xiong, Sutherland, et al. (2012), and NPR- to the R_α estimates. The horizontal line shows the best previously reported result (Qin and Yung 2010), though others have since slightly surpassed our results here.

4.2.2 Deep features

For the last several years, however, modern computer vision has become overwhelmingly based on deep neural networks. Image classification networks typically broadly follow the architecture of Krizhevsky et al. (2012), i.e. several convolutional and pooling layers to extract complex features of input images followed by one or two fully-connected layers to classify the images.

The activations are of shape $n \times h \times w$, where n is the number of filters; each unit corresponds to an overlapping patch of the original image. We can therefore treat the activations as a sample of size hw from an n-dimensional distribution. Wu et al. (2015) set accuracy records on several scene classification datasets with a particular method of extracting features from distributions. That method, however, resorts to ad-hoc statistics; we compare to our more principled alternatives here.[3]

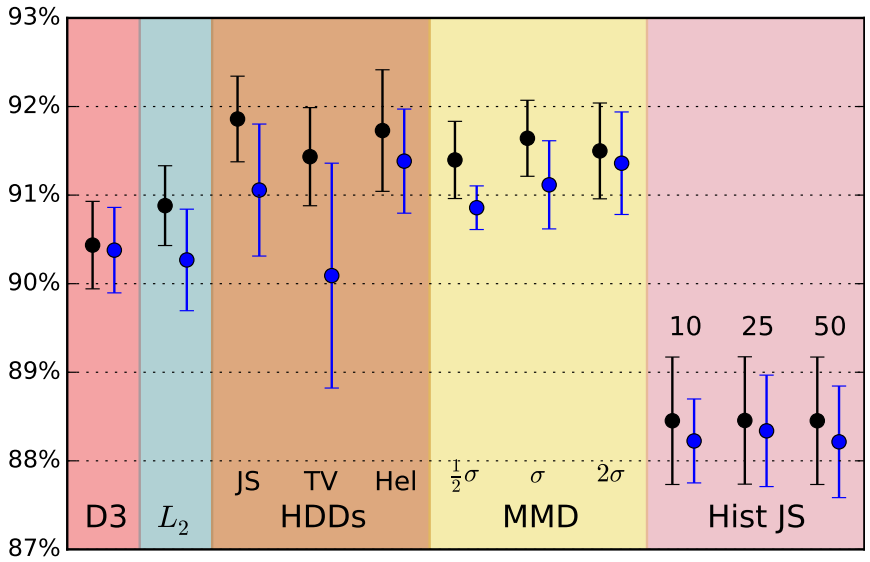


Figure 4.5: Mean and standard deviation accuracies on the Scene-15 dataset. The left, black lines show performance with linear features; the right, blue lines show generalized RBF embedding features. D3 refers to the method of Wu et al. (2015). MMD bandwidths are relative to σ, the median of pairwise distances; histogram methods use varying numbers of bins.

We consider here the Scene-15 dataset (Lazebnik et al. 2006), which contains 4 485 natural images in 15 categories based on location. (It is a superset of the OT dataset previously considered,

[3]These experiments appear in Sutherland, J. B. Oliva, et al. (2015).

but is available only in grayscale.) We follow Wu et al. (2015) in extracting features from the last convolutional layer of the `imagenet-vgg-verydeep-16` model (Simonyan and Zisserman 2015). We replace that layer's rectified linear activations with sigmoid squashing to $[0, 1]$.[4] After resizing the images as did Wu et al. (2015), hw ranges from 400 to 1 000. There are 512 filter dimensions; we concatenate features $\hat{A}(\hat{p}_i)$ extracted from each independently.

We select 100 images from each class for training, and test on the remainder; Figure 4.5 shows the results of 10 random splits. We do not add any spatial information to the model, unlike Wu et al. (2015); still, we match the best prior published performance of 91.59 ± 0.48, using a deep network trained on a large scene classification dataset (Zhou et al. 2014). Adding spatial information brought the D3 method of Wu et al. (2015) slightly above 92% accuracy; their best hybrid method obtained 92.9%. Using these features, however, our methods match or beat MMD and substantially outperform D3, L_2, and the histogram embeddings.

4.3 Dark matter halo mass prediction

Galaxy clusters are the most massive gravitationally bound system in the universe, containing up to hundreds of galaxies embedded in dark matter halos. Their properties, especially total mass, are extremely useful for making inferences about fundamental cosmological parameters, but because they are composed largely of dark matter, measuring that mass is difficult.

One classical method is that of Zwicky (1933). The virial theorem implies that the dispersion of velocities in a stable system should be approximately related to the halo mass as a power law; by measuring the Doppler shift of spectra from objects in the cluster, we can estimate the dispersion of velocities in the direction along our line of sight, and thus predict the total mass. He did so for the Coma cluster and concluded that dark matter outweighed luminous matter.

Experimental evidence, however, implies points towards various complicating factors that disturb this relationship, and indeed results based on numerical simulation have shown that the predictions from this power law relationship are not as accurate as we would hope. We can therefore consider using all information available in the line-of-sight velocity distribution by directly learning a regression function from that distribution to total masses, based on data from simulation.[5]

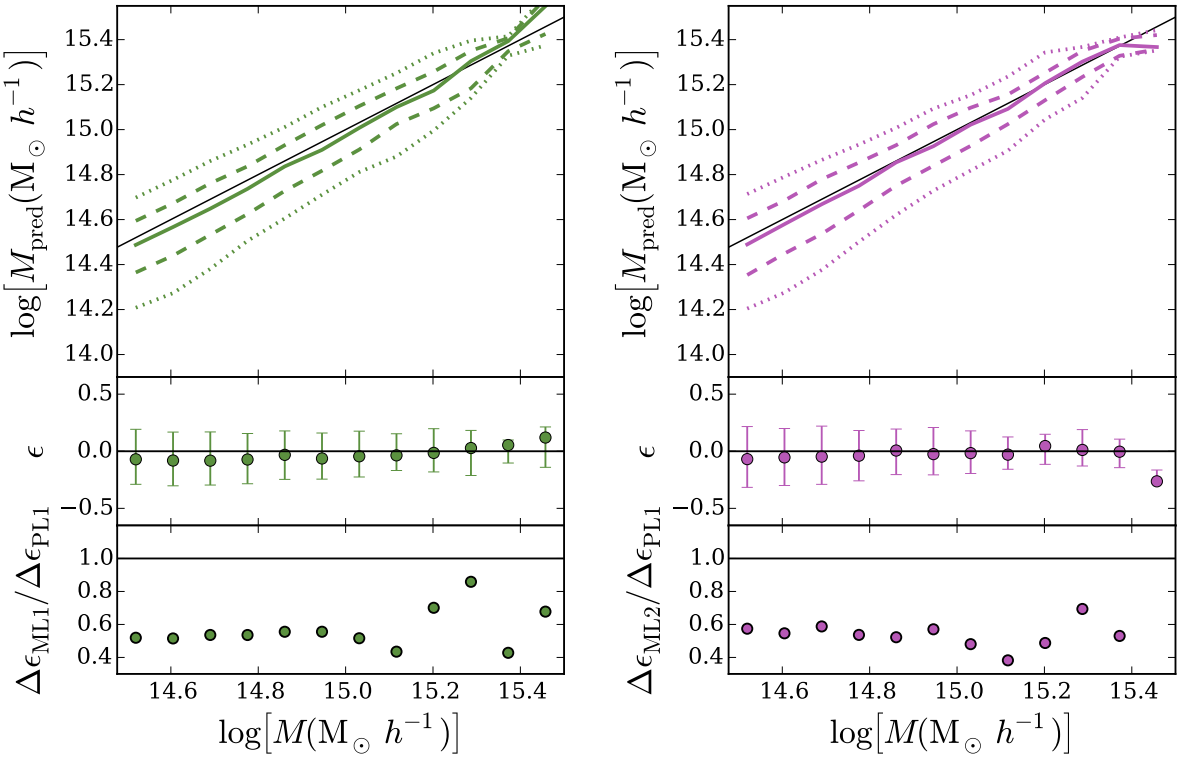
We assembled a catalog of massive halos from the MultiDark MDPL simulation (Klypin et al. 2014).The catalog contains 5 028 unique halos. Since we use only line-of-sight velocities, however, we can view each halo from multiple directions. For hyperparameter selection and testing, we use lines of sight corresponding to three perpendicular directions; for training, we additionally use projections sampled randomly from the unit sphere so as to oversample the rare high-mass halos. Different projections of the same halo are always assigned to the same fold for cross-validation. Ntampaka, Trac, Sutherland, Battaglia, et al. (2014) give a detailed description.

We then use the SKL estimator of Q. Wang et al. (2009) in a generalized RBF kernel on two sets of features: a one-dimensional feature set containing only the magnitude of the line-of-sight velocity, and a two-dimensional set adding $|v_{\text{los}}|/\sigma$, where σ is the standard deviation of that

[4] We used piecewise-linear weights such that 0 maps to 0.5, the 90th percentile of the positive observations maps to 0.9, and the 10th percentile of the negative observations to 0.1, for each filter.

[5] These results appear in Ntampaka, Trac, Sutherland, Battaglia, et al. (2014).

halo's $v_{\rm los}$ values. Thus, for the two-dimensional features, each halo's features lie on a line whose slope varies across halos.



(a) One-dimensional features.

(b) Two-dimensional features.

Figure 4.6: Performance for halo mass prediction for the two featurizations. Top panel: halo predicted mass vs. actual mass: median, 68% scatter, and 95% scatter. Middle panel: fractional mass error as a function of halo mass. Points are the median error; bars show 68% scatter. Bottom panel: error 68% width relative to that of the power law approach.

The SKL kernel substantially outperforms the power law approach.

Future work will compare different distributional regression methods, as well as more non-distributional methods (e.g. nonlinear regression directly from the velocity dispersion). Work in preparation (Ntampaka, Trac, Sutherland, Fromenteau, et al. 2015) additionally considers more realistic catalog construction, where "interloper" galaxies may appear to be part of the cluster, in which case the distributional regression approach is far more robust than the power law. Under these assumptions, we can also use features based on the absolute position, which is found to perform quite well.

Chapter 5

Active search for patterns

We will now change focuses slightly, and consider another problem setting in which collections of data play a key role.[1]

Consider a function containing interesting patterns that are defined only over a region of space. For example, if you view the direction of wind as a function of geographical location, it defines fronts, vortices, and other weather patterns, but those patterns are defined only in the aggregate. If we can only measure the direction and strength of the wind at point locations, we then need to infer the presence of patterns over broader spatial regions.

Many other real applications also share this feature. For example, an autonomous environmental monitoring vehicle with limited onboard sensors needs to strategically plan routes around an area to detect harmful plume patterns on a global scale (Valada et al. 2012). In astronomy, projects like the Sloan Digital Sky Survey (Eisenstein et al. 2011) search the sky for large-scale objects such as galaxy clusters. Biologists investigating rare species of animals must find the ranges where they are located and their migration patterns (Brown et al. 2014). We aim to use active learning to search for such global patterns using as few local measurements as possible.

This bears some resemblance to the artistic technique known as pointillism, where the painter creates small and distinct dots each of a single color, but when viewed as a whole they reveal a scene. Pointillist paintings typically use a denser covering of the canvas, but in our setting, "observing a dot" is expensive. Where should we make these observations in order to uncover interesting regions as quickly as possible?

We propose a probabilistic solution to this problem, known as *active pointillistic pattern search* (APPS). We assume we are given a predefined list of candidate regions and a classifier that estimates the probability that a given region fits the desired pattern. Our goal is then to find as many regions that are highly likely to match the pattern as we can. We accomplish this by sequentially selecting point locations to observe so as to approximately maximize expected reward.

[1]This chapter was previously published in longer form as Ma, Sutherland, et al. 2015.

5.1 Related Work

Our concept of active pattern search falls under the broad category of *active learning* (Settles 2012), where we seek to sequentially build a training set to achieve some goal as fast as possible. Our focus solely on finding positive ("interesting") regions, rather than attempting to learn to discriminate accurately between positives and negatives, is similar to the problem previously described as *active search* (Garnett et al. 2012). In previous work on active search, however, it has been assumed that the labels of interest can be revealed directly. In active pattern search, on the other hand, the labels are never revealed but must be inferred via a provided classifier. This indirection increases the difficulty of the search task considerably.

In *Bayesian optimization* (Osborne et al. 2009; Brochu et al. 2010), we seek to find the global optimum of an expensive black-box function. Bayesian optimization provides a model-based approach where a Gaussian process (GP) prior is placed on the objective function, from which a simpler acquisition function is derived and optimized to drive the selection procedure. Tesch et al. (2013) extend this idea to optimizing a latent function from binary observations. Our proposed active pattern search also uses a Gaussian process prior to model the unknown underlying function and derives an acquisition function from it, but differs in that we seek to identify entire *regions* of interest, rather than finding a single optimal value.

Another intimately related problem setup is that of *multi-arm bandits* (Auer et al. 2002), with more focus on analysis of the cumulative reward over all function evaluations. Originally, the goal was to maximize the expectation of a random function on a discrete set; a variant considers the optimization in continuous domains (Kroemer et al. 2010; Niranjan et al. 2010). However, like Bayesian optimization, multi-arm bandit problems usually do not consider discriminating a regional pattern.

Level set estimation (Low et al. 2012; Gotovos et al. 2013), rather than finding optima of a function, seeks to select observations so as to best discriminate the portions of a function above and below a given threshold. This goal, though related to ours, aims to directly map a portion of the function on the input space rather than seeking out instances of patterns. LSE algorithms can be used to attempt to find some simple types of patterns, e.g. areas with high mean.

APPS can be viewed as a generalization of *active area search* (AAS) (Ma, Garnett, et al. 2014), which is a considerably simpler version of active search for region-based labels. In AAS, the label of a region is only determined by whether its mean value exceeds some threshold. APPS allows for arbitrary classifiers rather than simple thresholds, and in some cases its expected reward can still be computed analytically. This extends the usefulness of this class of algorithms considerably.

5.2 Problem Formulation

There are three key components of the APPS framework: a function f which maps input covariates to data observations, a predetermined set of regions wherein instances of function patterns are expected, and a classifier that evaluates the salience of the pattern of function values in each region. We define $f\colon \mathbb{R}^m \to \mathbb{R}$ to be the function of interest,[2] which can be observed at any

[2]For clarity, in this and the next sections we will focus on scalar-valued functions f. The extension to vector-valued functions is straightforward; we consider such a case in the experiments.

location $x \in \mathbb{R}^m$ to reveal a noisy observation z. We assume the observation model $z = f(x) + \varepsilon$, where $\varepsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. We suppose that a set of regions where matching patterns might be found is predefined, and will denote these $\{g_1, \ldots, g_k\}$; $g_i \subset \mathbb{R}^m$. Finally, for each region g, we assume a classifier h_g which evaluates f on g and returns the probability that it matches the target pattern, which we call *salience*: $h_g(f) = h(f; g) \in [0, 1]$, where the mathematical interpretation of h_g is similar to a functional of f. Classifier forms are typically the same for all regions with different parameters.

Unfortunately, in general, we will have little knowledge about f other than the limited observations made at our selected set of points. Classifiers which take functional inputs (such as our assumed h_g) generally do not account for uncertainty in their inputs, which should be inversely related to the number of observed data points. We thus consider the probability that $h_g(f)$ is high enough, marginalized across the range of functions f that might match our observations. As is common in nonparametric Bayesian modeling, we model f with a Gaussian process (GP) prior; we assume that hyperparameters, including prior mean and covariance functions, are set by domain experts. Given a dataset $\mathcal{D} = (X, z)$, we define

$$f \sim \mathcal{GP}(\mu, \kappa); \qquad f \mid \mathcal{D} \sim \mathcal{GP}(\mu_{f|\mathcal{D}}, \kappa_{f|\mathcal{D}}),$$

to be a given GP prior and its posterior conditioned on $\mathcal{D}$, respectively. Thus, since f is a random variable, we can obtain the marginal probability that g is salient,

$$T_g(\mathcal{D}) = \mathbb{E}_f\left[h_g(f) \mid \mathcal{D}\right]. \tag{5.1}$$

We then define a matching region as one whose marginal probability passes a given threshold θ. Unit reward is assigned to each matching region g:

$$r_g(\mathcal{D}) \coloneqq \mathbb{1}\{T_g(\mathcal{D}) > \theta\}.$$

We make two assumptions regarding the interactive procedure. The first is that once a region is flagged as potentially matching (i.e., its marginal probability exceeds θ), it will be immediately flagged for further review and no longer considered during the run. The second is that the data resulting from this investigation will not be made immediately available during the course of the algorithm; rather the classifiers h_g will be trained offline. We consider both of these assumptions to be reasonable when the cost of investigation is relatively high and the investigation collects different types of data. For example, if the algorithm is being used to run autonomous sensors and scientists collect separate data to follow up on a matching region, these assumptions allow the autonomous sensors to continue in parallel with the human intervention, and avoid the substantial complexity of incorporating a completely different modality of data into the modeling process.

Garnett et al. (2012) attempt to maximize their reward at the end of a fixed number of queries. Directly optimizing that goal involves an exponential lookahead process. However, this can be approximated by a greedy search like the one we perform. Similarly, one could attempt to maximize the area under the recall curve through the search process. This also requires an intractable amount of computation which is often replaced with a greedy search.

We now write down the greedy criterion our algorithm seeks to optimize. Define $\mathcal{D}_t$ to be the already collected (noisy) observations of f before time step t and $\mathcal{G}_t = \{g : T_g(\mathcal{D}_\tau) \leq \theta, \forall \tau \leq t\}$

to be the set of remaining search subjects; we aim to greedily maximize the sum of rewards over all the regions in $\mathcal{G}_t$ in expectation,

$$\max_{x_*} \mathbb{E}\left[\sum_{g\in\mathcal{G}_t} r_g(\mathcal{D}_*) \,\middle|\, x_*, \mathcal{D}_t\right], \tag{5.2}$$

where $\mathcal{D}_*$ is the (random) dataset augmented with x_*.

This criterion satisfies a desirable property: when the regions are uncoupled and the classifier h_g is probit-linear, the point that maximizes (5.2) in each region also minimizes the variance of that region's label (Section 5.3.2).

5.3 Method

For the aim of maximizing the greedy expected reward of finding matching patterns (5.2), a more careful examination of the GP model can yield a straightforward sampling method. This method, in the following, turns out to be quite useful in APPS problems with rather complex classifiers. Section 5.3.1 introduces an analytical solution in an important special case.

At each step, given $\mathcal{D}_t = (X, z)$ as the set of any already collected (noisy) observations of f and x_* as any potential input location, we can assume the distribution of possible observations z_* as

$$z_* \mid x_*, \mathcal{D}_t \;\sim\; \mathcal{N}\left(\mu_{f|\mathcal{D}_t}(x_*),\; \kappa_{f|\mathcal{D}_t}(x_*, x_*) + \sigma^2\right). \tag{5.3}$$

Conditioned on an observation value z_*, we can update our GP model to include the new observation (x_*, z_*), which further affects the marginal distribution of region classifier outputs and thus the probability this region is matching. With $\mathcal{D}_* = \mathcal{D}_t \cup \{(x_*, z_*)\}$ as the updated dataset, we use $r_g(\mathcal{D}_*)$ to be the updated reward of region g. The utility of this proposed location x_* for region g is thus measured by the *expected* reward function, marginalizing out the unknown observation value z_*:

$$u_g(x_*, \mathcal{D}_t) := \mathbb{E}_{z_*}\left[r_g(\mathcal{D}_*) \mid x_*, \mathcal{D}_t\right] \tag{5.4}$$

$$= \Pr\{T_g(\mathcal{D}_*) > \theta \mid x_*, \mathcal{D}_t\}. \tag{5.5}$$

Finally, in active pointillistic pattern search, we select the next observation location x_* by considering its expected reward over the remaining regions:

$$x_* = \operatorname*{argmax}_x u(x, \mathcal{D}_t) = \operatorname*{argmax}_x \sum_{g\in\mathcal{G}_t} u_g(x, \mathcal{D}_t). \tag{5.6}$$

For the most general definition of the region classifier h_g, the basic algorithm is to compute (5.4) and thus (5.6) via sampling at two stages:

1. Sample the outer variable z_* in (5.4) according to (5.3).
2. For every draw of z_*, sample enough of $(f \mid \mathcal{D}_*)$ to compute the marginal reward $T_g(\mathcal{D}_*)$ in (5.1), in order to obtain one draw for the expectation in (5.4).

To speed up the process, we can evaluate (5.6) for a subset of possible x_* values, as long as a good action is likely to be contained in the set.

5.3.1 Analytic Expected Utility for Functional Probit Models

For a broad family of classifiers, we can compute both (5.1) and (5.5) analytically, allowing us to efficiently perform exact searches for potentially complex patterns. This family is formed by a probit link function of any affine functional of f.

Suppose we have observed data $\mathcal{D}$, yielding the posterior $p(f \mid \mathcal{D}) = \mathcal{GP}(f; \mu_{f|\mathcal{D}}, \kappa_{f|\mathcal{D}})$. Let L_g be a linear functional, $L_g \colon f \mapsto L_g f \in \mathbb{R}$, associated with region g. The family of classifiers is:

$$h_g(f) = \Phi(L_g f + b_g),$$

where Φ is the cumulative distribution function of the standard normal and $b \in \mathbb{R}$ is an offset. Two examples of such functionals are:

- $L_g f \colon f \mapsto \frac{c}{|g|} \int_g f(x) \mathrm{d}x$, where $|g|$ is the volume of region $g \subset \mathbb{R}^m$. Here $L_g f$ is the mean value of f on g, scaled by an arbitrary $c \in \mathbb{R}$. When $|c| \to \infty$ the model becomes quite similar to that of Ma, Garnett, et al. (2014).
- $L_g f \colon f \mapsto w^\mathsf{T} f(\Xi)$, where Ξ is a finite set of fixed points $\{\xi_i\}_{i=1}^{|\Xi|}$, and $w \in \mathbb{R}^{|\Xi|}$ is an arbitrary vector. This mapping applies a linear classifier to a fixed, discrete set of values from f.

Section 3.1 of Ma, Sutherland, et al. (2015) shows that the expected reward is:

$$u_g(x_*, \mathcal{D}) = \Phi\left(\frac{L_g \mu_{f|\mathcal{D}} + b - \sqrt{1 + L_g^2 \kappa_{f|\mathcal{D}_*}}\, \Phi^{-1}(\theta)}{\sqrt{V_{*|\mathcal{D}}^{-1} L_g \left[\kappa_{f|\mathcal{D}}(\cdot, x_*)\right]^2}} \right) \tag{5.7}$$

where L_g^2 is the bilinear form $L_g^2 \kappa : \kappa \mapsto L[L\kappa(x, \cdot)] = L[L\kappa(\cdot, x')]$.

5.3.2 Analysis for Independent Regions

The analytical solution to (5.5) by (5.7) enables us to further study the theory behind the exploration/exploitation tradeoff of APPS in one nontrivial case: when all regions are approximately independent. This assumption allows us to ignore the effect a data point has on regions other than its own. We will answer two questions in this case: which region will APPS explore next, and what location will be queried for that region.

Define

$$\rho_g(x_*) \coloneqq \frac{\sqrt{V_{*|\mathcal{D}}^{-1} L_g \left[\kappa_{f|\mathcal{D}}(\cdot, x_*)\right]^2}}{\sqrt{1 + L_g^2 \kappa_{f|\mathcal{D}}}} = \left| \mathrm{Corr}\left(z_*, L_g f + b + \varepsilon_1 \;\middle|\; x_*, \mathcal{D} \right) \right|,$$

where $\varepsilon_1 \sim \mathcal{N}(0, 1)$ is independent noise, denote how informative the point z_* is to the label of its region g. Also define how close g is to receiving a reward by

$$R_g \coloneqq \frac{\Phi^{-1}(T_g(\mathcal{D}))}{\Phi^{-1}(\theta)}.$$

Section 3.2 of Ma, Sutherland, et al. (2015) shows that for regions not currently carrying a reward:

1. For any region g, $u_g(x, \mathcal{D})$ is maximized by choosing the x that yields $\rho_g^* \coloneqq \max_x \rho_g(x)$.
2. If two regions g and g' can be equally explored ($\rho_g^* = \rho_{g'}^*$), then the region with more probability of matching R will be selected.
3. If two regions are equally likely to match the desired pattern ($R_g = R_{g'}$), the more explorable region (that with a larger ρ^*) will be selected.
4. In general, APPS will trade off the two factors by maximizing $\left(R_g - \sqrt{1 - (\rho_g^*)^2}\right) / \rho_g^*$.

5.4 Empirical Evaluation

Ma, Sutherland, et al. (2015) evaluates the framework in three different settings, with three different classifiers. We reproduce only one of these evaluations here. The others are based on real environmental monitoring data and electoral prediction data, using the analytical results of Section 5.3.1.

The problem we consider here requires more complex pattern classifiers. We study the task of identifying vortices in a vector field based on limited observations of flow vectors. Linear classifiers are insufficient for this problem,[3] so we will demonstrate the flexibility of our approach with a black-box classifier.

To illustrate this setting, we consider the results of a large-scale simulation of a turbulent fluid in three dimensions over time in the Johns Hopkins Turbulence Databases[4] (Perlman et al. 2007). Following Sutherland, Xiong, et al. (2012), we aim to recognize vortices in two-dimensional slices of the data at a single timestep, based on the same small training set of 11 vortices and 20 non-vortices, partially shown in Figure 5.1.

Recall that h_g assigns probability estimates to the entire function class $\mathcal{F}$ confined to region g. We can consider the average flow across sectors (angular slices from the center) of our region as building blocks in detecting vortices. We count how many sectors have clockwise/counter-clockwise flows to give a classification result, in three steps:

1. First, we divide a region into K sectors. In each sector, we take the integral of the inner product between the actual flow vectors and a template. The template is an "ideal" vortex, but with larger weights in the center than the periphery. This produces a K-dimensional summary statistic $L_g(f)$ for each region.
2. Next, we improve robustness against different flow speeds in the data by scaling $L_g(f)$ to have maximum entry 1, and flip its sign if its mean is negative. Call the result $\tilde{L}_g(f)$.
3. Finally, we feed the normalized $\tilde{L}_g(f)$ vector through a 2-layer neural network of the form

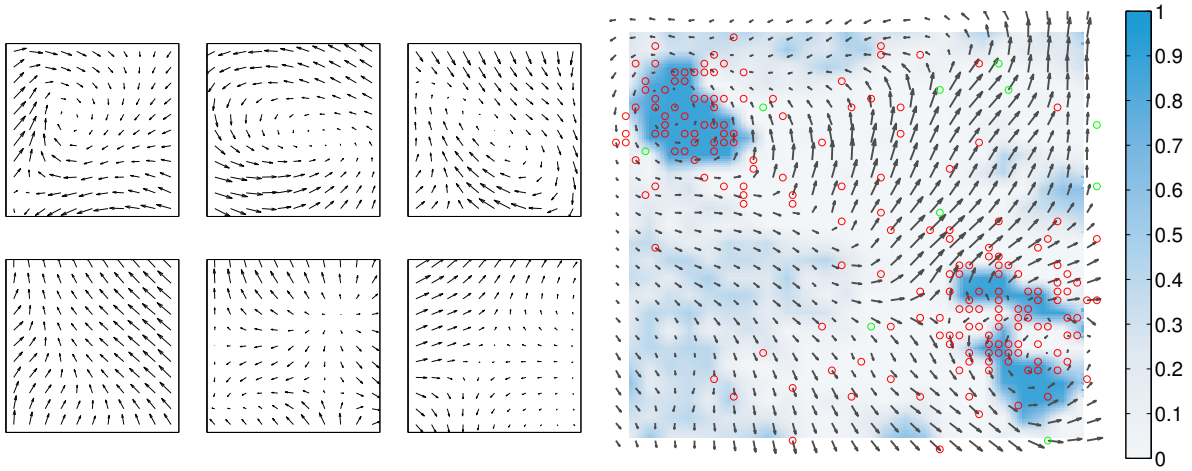
$$h_g(f) = \sigma\left(w_{\text{out}} \sum_{i=1}^{K} \sigma\left(w_{\text{in}} \tilde{L}_g(f)_i + b_{\text{in}}\right) + b_{\text{out}}\right),$$

   where σ is the logistic sigmoid function.

[3] The set of vortices is not convex: consider the midpoint between a clockwise vortex and its identical counter-clockwise case.

[4] http://turbulence.pha.jhu.edu

$L_g(f) \mid \mathcal{D}$ obeys a K-dimensional multivariate normal distribution, from which we can sample many possible $L_g(f)$, which we then normalize and pass through the neural network as described above. This gives samples of probabilities h_g, whose mean is a Monte Carlo estimate of (5.1).

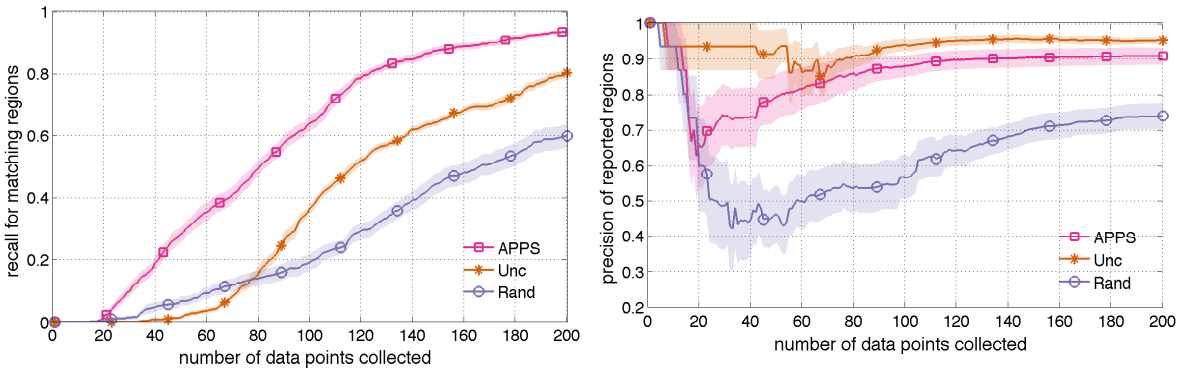


(a) (b)

Figure 5.1: (a): Positive (top) and negative (bottom) training examples for the vortex classifier. (b): The velocity field used; each arrow is the average of a 2×2 square of actual data points. Background color shows the probability obtained by each region classifier on the 200 circled points; red circles mark points selected by one run of APPS initialized at the green circles.

We used $K = 4$ sectors, and the weights in the template were fixed such that the length scale matches the distance from the center to an edge. The network was optimized for classification accuracy on the training set. We then identified a 50×50-pixel slice of the data that contains two vortices, some other "interesting" regions, and some "boring" regions, mostly overlapping with Figure 11 of Sutherland, Xiong, et al. (2012); the region, along with the output of the classifier when given all of the input points, is shown in Figure 5.1a. We then ran APPS, initialized with 10 uniformly random points, for 200 steps. We defined the regions to be squares of size 11×11 and spaced them every 2 points along the grid, for 400 total regions. We again thresholded at $\theta = 0.7$. We evaluate (5.1) via a Monte Carlo approximation: first we took 4 samples of z_*, and then 15 samples from the posterior of f over the window for each z_*. Furthermore, at each step we evaluate a random subset of 80 possible candidates x_*.

Figure 5.2a shows recall curves of APPS, uncertainty sampling (UNC), and random selection (RAND), where for the purpose of these curves we call the true label the output of the classifier when all data is known, and the proposed label is true if $T_g > \theta$ at that point of the search (evaluated using more Monte Carlo samples than in the search process, to gain assurance in our evaluation but without increasing the time required for the search). We can see that active pattern search substantially outperforms uncertainty sampling and random selection. It is interesting to observe that RAND was initially better than, but later crossed by UNC. In the beginning, since UNC is purely explorative, its reward uniformly remained low across multiple runs, whereas in some runs RAND queries can be lucky enough to concentrate around matching regions. At a later



(a) Recall curves.

(b) Precision curves.

Figure 5.2: Results for the vortex experiment. Color bands show standard errors over 15 runs.

phase, RAND faces the coupon collector's problem and may select redundant boring observations, whereas UNC keeps making progress at a constant rate.

Chapter 6

Proposed work

6.1 Integration with deep computer vision models

In Section 4.2, we considered using the features learned by a standard convolutional deep network as samples from an image-level distribution of local features, and classified images based on those sets of features. Here features are trained using fully-connected final layers as the learning model, but then used in a separate distributional kernel model.

We can instead make a coherent model which combines feature extraction with a learning model based on a distributional kernel, by treating the approximate distributional embedding as a layer in the network. With the mean map-based embedding, gradients propagate through this layer easily, and so standard stochastic gradient algorithms can be used either to fine-tune features trained on a different task or to learn features well-suited to distributional kernel models from the start.

In fact, we can also consider using distributional embeddings in intermediate layers of the net; gradients propagate in the same way. This could help in learning more complex filters inside the net with fewer training examples.

I propose to:

1. Implement the necessary components to add approximate distribution embedding layers to standard deep learning software such as Caffe or Torch.
2. Evaluate the use of fine-tuning features for scene recognition models using distributional classifiers in the last layers.
3. Experiment with image recognition using distributional filters in earlier layers.

6.2 Word embeddings as distributions

Until recently, much work in natural language processing treated words as unique symbols, e.g. with "one-hot" vectors, where the ith word from a vocabulary of size V is represented as a vector with ith component 1 and all other components 0. It has recently become widely accepted that applications can benefit from richer word embeddings which take into account the similarity between distinct words, and much work has been done on dense *word embeddings* so that distances

or inner products between word embeddings represent word similarity in some way (e.g. Collobert and Weston 2008; Turian et al. 2010; Mikolov et al. 2013). These embeddings can be learned in various ways, but often involve optimizing the representation's performance in some supervised learning task.

Document representations First, it is worth noting that although this breaks the traditional "bag of words" text model (where documents can be represented simply by the sum of the words' one-hot encodings), we can represent documents by viewing them as sample sets of word vectors.

Kusner et al. (2015) recently adopted this model, using kNN classifiers based on the EMD between documents, and obtained excellent empirical results. EMD, however, is expensive to compute even for each pair of documents when the vocabulary is large, and additionally must be computed pairwise between documents; an approximate embedding in the style of Chapter 3 is not known.

Yoshikawa et al. (2014), in their empirical results, considered this model with MMD-based kernels (but computing pairwise kernel values rather than approximate embeddings). Their main contribution, however, is to optimize the word embedding vectors for final classification performance; by doing so with random initializations, they saw mild performance improvements over MMD kernels using substantially less training data for the embeddings but at much higher computational cost. Yoshikawa et al. (2015) extend the approach to Gaussian process regression models, but do not compare to separately-learned word embeddings.

I propose to:

4. Empirically compare these embedding methods, particularly on larger datasets, to establish the best method for bag-of-words document representation.
5. Fine-tune word embeddings learned on a standard dataset simultaneously with learning the model for a particular application, as is common in deep learning models for computer vision, using techniques similar to those of Section 6.1.

Richer word representation Embedding words as a single vector does not allow for as rich a word representation as we might wish. Vilnis and McCallum (2015) embed words instead as Gaussian distributions, and use the KL divergence between word embeddings to measure asymmetric hypernym relationships: for example, their embedding for the word *Bach* is "included" in their embeddings for *famous* and *man*, and mostly included in *composer*. Gaussian distributions, of course, are still fairly limiting; for example, a multimodal embedding might be able to capture word sense ambiguity, whereas a Gaussian embedding would be forced to attempt to combine both senses in a single broad embedding.

We can thus consider richer, nonparametric classes of word embeddings: perhaps by representing a word as a (possibly weighted) set of latent vectors. Comparisons could then be performed either with an MMD-based kernel, when symmetry is desired, or with KL estimators (or similar) when not.

One approach would be to choose these vectors arbitrarily, optimizing them for the output of some learning problem: this would be implemantionally similar to the approach of Yoshikawa et al. (2014, 2015) for MMD distances, or somewhat like that of Vilnis and McCallum (2015) but with greater computational cost, and greater flexibility, for KL distances.

Another approach is inspired by the classic distributional hypothesis of Harris (1954), that the semantics of words are characterized by the contexts in which it appears. Many word embedding approaches can be viewed as matrix factorizations of a matrix M with rows corresponding to words, columns to some notion of context, and entries containing some measure of association between the two; the factorization $M = WC^\mathsf{T}$ then typically discards the matrix C and uses the rows of W as word vectors. This approach is sometimes taken explicitly; interestingly, the popular method of Mikolov et al. (2013) can be seen as approximating this form as well (Levy and Goldberg 2014). This view inspires a natural alternative: treat each word as the sample set of contexts in which it appears, representing each context via the learned context vectors. This is perhaps the most direct instantiation of the distributional hypothesis: compare words by comparing the distribution of contexts in which they appear.

I propose to:

6. Develop efficient methods for learning nonparametric distributional word embeddings for asymmetric divergences.
7. Empirically evaluate nonparametric distribution-based word embeddings, based both on arbitrary embeddings and context features, both on analogy tasks and downstream document classification tasks.

6.3 Kernel learning for distribution embeddings

Mean map methods rely on having a good base kernel in order to make good comparisons between distributions, whether for kernels in learning problems or for two sample tests. The most common technique is to choose a simple family of kernels, perhaps the Gaussian kernel with various bandwidths selected as multiples of the median inter-point distance, and then pick the kernel that performs best on a validation set — as with typical hyperparameter optimization. For two-sample tests, Sriperumbudur, Fukumizu, Gretton, Lanckriet, et al. (2009) propose instead using the kernel that yields the maximum distance between distributions over certain families of distributions. Gretton, Sriperumbudur, et al. (2012) find the optimal positive linear combination of kernels for (linear variants of) a two-sample test. Neither technique, however, seems to have been thoroughly evaluated outside the context of two-sample testing.

We can also consider more complex kernel learning frameworks than multiple kernel learning. Z. Yang et al. (2015) recently proposed a spectral kernel learning method capable of learning richer kernels than simple linear combinations or bandwidth selection; integrating that method into the mean map kernel, for two-sample testing as well as for machine learning models, could prove fruitful in increasing the power and reducing the amount of human intervention needed in using these models.

For distances being used within a deep network, as in Section 6.1, allowing the network to adapt the embedding may also provide an effective technique for learning the base kernel.

I propose to:

8. Adapt the spectral kernel method of Z. Yang et al. (2015) to mean map methods, in both classification or regression settings and in the two-sample test setting (as in Gretton, Sriperumbudur, et al. 2012).

9. Evaluate kernel learning for MMD kernel embeddings in deep networks.

6.4 Embeddings for other kernels

Despite the advantages of mean-map approaches, it may be that in some settings, other distances such as EMD may induce more useful kernels. If this holds true experimentally, it may make sense to attempt to develop approximate embeddings for other kernels.

Given the close relation between MMD and other integral probability metrics (Müller 1997), as discussed in Chapter 2, the question of finding base kernels such that MMD approximates related distances is also quite interesting.

In fact, Sriperumbudur, Gretton, et al. (2010, Theorem 21) show the following: Let k be a measurable kernel with $\sup_{x \in \mathcal{X}} k(x,x) \leq C < \infty$, and define $\rho(x,y) = \|k(\cdot,x) - k(\cdot,y)\|_{\mathcal{H}}$. Then $\mathrm{MMD}_k(P,Q) \leq \sqrt{C}\,\mathrm{TV}(P,Q)$, and if $(\mathcal{X},\rho)$ is separable then $\mathrm{MMD}_k(P,Q) \leq \mathrm{EMD}_\rho(P,Q) \leq \sqrt{\mathrm{MMD}_k(P,Q)^2 + 4C}$.

The first statement implies that choosing k to maximize MMD_k (from a family with fixed C), similar to Sriperumbudur, Fukumizu, Gretton, Lanckriet, et al. (2009), makes MMD approximate TV more closely. But how close is this approximation? For certain classes of distributions, can we find a kernel that closely approximates TV?

When k is the linear kernel (so $\rho(x,y) = \|x-y\|_2$) and $\mathcal{X} \subseteq \{x \in \mathbb{R}^d : \|x\| \leq R\}$ (so $C = R^2$), MMD approximates the Euclidean EMD, but the bounds are quite loose — note that the linear-kernel MMD is simply the distance between the means of the distributions. Can we choose a kernel k such that MMD_k more closely approximates the Euclidean EMD? (Since MMD is Hilbertian and EMD is not, some distortion is unavoidable.)

I propose to:

10. Look into whether, for certain classes of distributions, kernels can be obtained such that MMD approximates EMD or TV — particularly kernels admitting a good approximate embedding.
11. Investigate emperically if areas where EMD or TV are in common usage cannot be handled more simply by more efficient MMD kernels, particularly with the kernel learning methods of Section 6.3.

6.5 Active learning on distributions

Suppose we have a collection of distributions, but initially we have very few samples from each distribution. We can choose to take additional iid observations, but doing so is relatively expensive; perhaps it requires real-world expenditure of time or resources to collect samples, or perhaps these distributions are available only through computationally intensive numerical simulations. We may wish to learn a classification or regression function mapping from these distributions to some label (similar to traditional active learning settings), to locate distributions which follow some prespecified pattern (similar to the setting of Chapter 5 with independent regions), or to find the distribution which is "best" in some sense (as in pure-exploration bandit problems, Bubeck et al. 2010). In any of these cases, we need to choose some selection criterion

that will appropriately consider the utility of selecting points from distributions, a problem that is related to but certainly distinct from typical fully-observed active learning models.

In the dark matter prediction experiments of Section 4.3, we assumed that each observed galaxy has a well-known line-of-sight velocity estimated via redshift. In practice, good velocity estimates are available only through relatively-expensive spectroscopic imaging; cheaper few-color imaging techniques give extremely uncertain velocity estimates. We could simply ignore the imaging estimates and apply the previous model, selecting a random galaxy from each halo to perform spectroscopy upon. It would probably be more effective, however, to consider active learning methods that begin with visual imaging, and then identify which objects will be useful for spectroscopy in order to best identify the masses of their dark matter halos. One modeling option would be to take a probability distribution over the sample set, and then identify the resulting distribution of the mean map embedding and therefore its predicted label under a learned predictor; we would then identify objects to observe that most reduce uncertainty in the predicted label. This could be conducted either for a single halo, where the objective is to best learn its mass, or across multiple halos, where the objective is either to find the most massive halos (active search) or to reduce some form of overall uncertainty in all of the halo mass predictions (active learning).

I propose to:

12. Develop and evaluate efficient active learning selection criteria for each of the problem settings discussed here.

6.6 Timeline

My rough plan for the proposed work is as follows:

September – December 2015: Integration with deep networks; experiments in computer vision and document representation. Conference submission. (Partially complete.)

October – December 2015: Investigate embeddings for other kernels. If successful, experiments in various domains and possible conference submission.

December 2015 – February 2016: Kernel learning for distribution embeddings. Conference submission.

March – April 2016: Word embeddings. Possible conference submission.

April – July 2016: Active learning on distributions, with astronomical application. Submission to a machine learning conference and/or an astrophysics venue.

August 2016: Thesis writing.

September 2016: Thesis defense.

Bibliography

Akiake, Hirotugu (1973). "Information theory and an extension of the maximum likelihood principle". In: *2nd International Symposium on Information Theory* (page 25).

Amari, Shun-ichi (1985). *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics 28. Springer (page 6).

Andoni, Alexandr and Ilya Razenshteyn (2015). "Optimal Data-Dependent Hashing for Approximate Near Neighbors". In: *ACM Symposium on Theory of Computing*. arXiv: 1501.01062 (page 10).

Auer, Peter, Nicoló Cesa-Bianchi, and Paul Fischer (2002). "Finite-time Analysis of the Multiarmed Bandit Problem". In: *Machine Learning* 47, pages 235–256 (page 32).

Bengio, Yoshua, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet (2004). "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering". In: *Advances in Neural Information Processing Systems*. NIPS (page 13).

Berlinet, Alain and Christine Thomas-Agnan (2004). *Reproducing kernel Hilbert spaces in Probability and Statistics*. Kluwer Academic Publishers (page 9).

Betancourt, Michael (2015). "Adiabatic Monte Carlo". Version 5. In: arXiv: 1405.3489v5 (page 23).

Beygelzimer, Alina, Sham Kakade, and John Langford (2006). "Cover trees for nearest neighbor". In: *International Conference on Machine Learning*, pages 97–104 (page 10).

Bochner, Salomon (1959). *Lectures on Fourier integrals*. Princeton University Press (page 15).

Boiman, Oren, Eli Shechtman, and Michal Irani (2008). "In defense of nearest-neighbor based image classification". In: *Computer Vision and Pattern Recognition* (page 5).

Bosch, A., A. Zisserman, and X. Muñoz (2006). "Scene classification via pLSA". In: *ECCV* (page 28).

Bosch, Anna, Andrew Zisserman, and Xavier Munoz (2008). "Scene Classification Using a Hybrid Generative/Discriminative Approach". In: *IEEE Trans. PAMI* 30.4 (page 27).

Bretagnolle, Jean, Didier Dacunha Castelle, and Jean-Louis Krivine (1966). "Lois stables et espaces L^P". French. In: *Annales de l'institut Henri Poincaré (B) Probabilités et Statistiques* 2 (3), pages 231–259 (page 12).

Brochu, Eric, Vlad M Cora, and Nando de Freitas (2010). *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*. arXiv: 1012.2599 (page 32).

Brown, Jason L, Alison Cameron, Anne D Yoder, and Miguel Vences (2014). "A necessarily complex model to explain the biogeography of the amphibians and reptiles of Madagascar." In: *Nature communications* 5, page 5046 (page 31).

Bubeck, Sébastien, Rémi Munos, and Gilles Stoltz (2010). "Pure exploration in multi-armed bandits problems". In: *Algorithmic Learning Theory*, pages 23–37. arXiv: 0802.2655 (page 42).

Cha, Sung Hyuk and Sargur N. Srihari (2002). "On measuring the distance between histograms". In: *Pattern Recognition* 35.6, pages 1355–1370 (page 8).

Chen, Yihua, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti (2009). "Similarity-based classification: Concepts and algorithms". In: *Journal of Machine Learning Research* 10, pages 747–776 (pages 12, 13).

Chwialkowski, Kacper, Aaditya Ramdas, Dino Sejdinovic, and Arthur Gretton (2015). "Fast Two-Sample Testing with Analytic Representations of Probability Measures". In: arXiv: 1506.04725 (page 19).

Collobert, Ronan and Jason Weston (2008). "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning". In: *ICML* (page 40).

Csiszár, I. (1963). "Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten". German. In: *Magyar. Tud. Akad. Mat. Kutato Int. Kozl* 8, pages 85–108 (page 6).

Eisenstein, Daniel J., David H. Weinberg, Eric Agol, et al. (2011). "SDSS-III: Massive Spectroscopic Surveys of the Distant Universe, the Milky Way, and Extra-Solar Planetary Systems". In: *The Astronomical Journal* 142, 72, page 72. arXiv: 1101.1529 (page 31).

Flaxman, Seth R., Yu-xiang Wang, and Alexander J. Smola (2015). "Who Supported Obama in 2012? Ecological Inference through Distribution Regression". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*. ACM Press, pages 289–298 (pages 1, 19).

Fuglede, Bent (2005). "Spirals in Hilbert space: With an application in information theory". In: *Expositiones Mathematicae* 23.1, pages 23–45 (pages 12, 22).

Gardner, Andrew, Christian a. Duncan, Jinko Kanno, and Rastko R. Selmic (2015). "Earth Mover's Distance Yields Positive Definite Kernels For Certain Ground Distances". In: arXiv: 1510.02833. URL: http://arxiv.org/abs/1510.02833 (page 12).

Garnett, Roman, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard P Mann (2012). "Bayesian Optimal Active Search and Surveying". In: *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)* (pages 32, 33).

Gotovos, Alkis, Nathalie Casati, Gregory Hitz, and Andreas Krause (2013). "Active Learning for Level Set Estimation". In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)* (page 32).

Grauman, Kristen and Trevor Darrell (2007). "The Pyramid Match Kernel: Efficient Learning with Sets of Features". In: *JMLR* 8, pages 725–760 (pages 10, 28).

Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alex J Smola (2012). "A Kernel Two-Sample Test". In: *The Journal of Machine Learning Research* 13 (pages 9, 11).

Gretton, Arthur, Bharath K. Sriperumbudur, Dino Sejdinovic, Heiko Strathmann, and Massimiliano Pontil (2012). "Optimal kernel choice for large-scale two-sample tests". In: *Advances in Neural Information Processing Systems*. Volume 25, pages 1214–1222 (page 41).

Haasdonk, Bernard and Claus Bahlmann (2004). "Learning with Distance Substitution Kernels". In: *Pattern Recognition: 26th DAGM Symposium*, pages 220–227 (page 11).

Harris, Z. (1954). "Distributional structure". In: *Word* 10.23, pages 146–162 (page 41).

Jebara, T., R. Kondor, A. Howard, K. Bennett, and N. Cesa-bianchi (2004). "Probability product kernels". In: *JMLR* 5, pages 819–844 (pages 9, 10).

Jitkrittum, Wittawat, Arthur Gretton, Nicolas Heess, S M Ali Eslami, Balaji Lakshminarayanan, Dino Sejdinovic, and Zoltán Szabó (2015). "Kernel-Based Just-In-Time Learning for Passing Expectation Propagation Messages". In: *Uncertainty in Artificial Intelligence* (pages 1, 19, 25).

Khosravifard, Mohammadali, Dariush Fooladivanda, and T. Aaron Gulliver (2007). "Confliction of the Convexity and Metric Properties in f-Divergences". In: *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences* E90-A.9, pages 1848–1853 (page 6).

Klypin, Anatoly, Gustavo Yepes, Stefan Gottlober, Francisco Prada, and Steffen Hess (2014). "MultiDark simulations: the story of dark matter halo concentrations and density profiles". In: arXiv: 1411.4001 (page 29).

Krishnamurthy, Akshay, Kirthevasan Kandasamy, Barnabás Póczos, and Larry Wasserman (2014). "Nonparametric Estimation of Rényi Divergence and Friends". In: *International Conference on Machine Learning*. arXiv: 1402.2966 (page 10).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances In Neural Information Processing Systems*. arXiv: 1102.0183 (page 28).

Kroemer, O. B., R. Detry, J. Piater, and J. Peters (2010). "Combining active learning and reactive control for robot grasping". In: *Robotics and Autonomous Systems* 58.9, pages 1105–1116 (page 32).

Kusner, Matt, Yu Sun, Nicholas Kolkin, and Kilian Weinberger (2015). "From Word Embeddings To Document Distances". In: *Proceedings of The 32nd International Conference on Machine Learning*, pages 957–966 (pages 5, 40).

Lafferty, John, Han Liu, and Larry Wasserman (2012). "Sparse Nonparametric Graphical Models". In: *Statistical Science* 27.4, pages 519–537. arXiv: 1201.0794 (page 23).

Lan, Shiwei, Jeffrey Streets, and Babak Shahbaba (2014). "Wormhole Hamiltonian Monte Carlo". In: *AAAI*. arXiv: 1306.0063 (page 23).

Lázaro-Gredilla, Miguel, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal (2010). "Sparse Spectrum Gaussian Process Regression". In: *Journal of Machine Learning Research* 11, pages 1865–1881 (page 21).

Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce (2006). "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *CVPR* (page 28).

Leung, Thomas and Jitendra Malik (2001). "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons". In: *IJCV* 43, pages 29–44 (page 10).

Levy, Omer and Yoav Goldberg (2014). "Neural Word Embedding as Implicit Matrix Factorization". In: *Advances in Neural Information Processing Systems*, pages 2177–2185 (page 41).

Li, Shukai and Ivor W Tsang (2011). "Learning to Locate Relative Outliers". In: *Asian Conference on Machine Learning*. Volume 20. JMLR: Workshop and Conference Proceedings, pages 47–62 (page 19).

Liese, Friedrich and Igor Vajda (2006). "On divergences and informations in statistics and information theory". In: *IEEE Transactions on Information Theory* 52.10, pages 4394–4412 (page 6).

Lopez-Paz, David, Krikamol Muandet, Bernhard Schölkopf, and Ilya Tolstikhin (2015). "Towards a Learning Theory of Cause-Effect Inference". In: *ICML*. arXiv: 1502.02398 (pages 1, 19).

Low, Kian Hsiang, Jie Chen, John M. Dolan, Steve Chien, and David R. Thompson (2012). "Decentralized Active Robotic Exploration and Mapping for Probabilistic Field Classification in Environmental Sensing". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*. AAMAS '12. Valencia, Spain, pages 105–112 (page 32).

Lowe, David G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2, pages 91–110 (page 27).

Ma, Yifei, Roman Garnett, and Jeff Schneider (2014). "Active Area Search via Bayesian Quadrature". In: *Seventeenth International Conference on Artificial Intelligence and Statistics*. AISTATS (pages 2, 32, 35).

Ma, Yifei, Dougal J. Sutherland, Roman Garnett, and Jeff Schneider (2015). "Active Pointillistic Pattern Search". In: *Eighteenth International Conference on Artificial Intelligence and Statistics*. AISTATS (pages 3, 31, 35, 36).

Martins, André F. T., Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo (2009). "Nonextensive Information Theoretic Kernels on Measures". In: *The Journal of Machine Learning Research* 10 (pages 7, 8).

Mehta, Nishant A. and Alexander G. Gray (2010). "Generative and Latent Mean Map Kernels". In: arXiv: 1005.0188 (page 19).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems* (pages 40, 41).

Moon, Kevin R. and Alfred O. Hero (2014a). "Ensemble estimation of multivariate f-divergence". In: *2014 IEEE International Symposium on Information Theory*. IEEE, pages 356–360. arXiv: 1404.6230 (page 11).

— (2014b). "Multivariate f-divergence Estimation With Confidence". In: *Advances in Neural Information Processing Systems*, pages 2420–2428 (page 11).

Moreno, Pedro J., Purdy P. Ho, and Nuno Vasconcelos (2004). "A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications". In: *NIPS* (page 10).

Muandet, Krikamol, Kenji Fukumizu, Bharath K Sriperumbudur, Arthur Gretton, and Bernhard Schölkopf (2014). "Kernel Mean Estimation and Stein's Effect". In: *International Conference on Machine Learning*. arXiv: arXiv:1306.0842v2 (page 11).

Muandet, Krikamol, Bernhard Schölkopf, Kenji Fukumizu, and Francesco Dinuzzo (2012). "Learning from Distributions via Support Measure Machines". In: *Advances in Neural Information Processing Systems*. arXiv: arXiv:1202.6504v2 (page 9).

Muja, Marius and David G. Lowe (2009). "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration". In: *International Conference on Computer Vision Theory and Applications (VISAPP '09)* (page 10).

Müller, Alfred (1997). "Integral Probability Metrics and their Generating Classes of Functions". In: *Advances in Applied Probability* 29.2, pages 429–443 (pages 6, 42).

Naor, Assaf and Gideon Schechtman (2007). "Planar Earthmover is not in L_1". In: *SIAM Journal on Computing* 37.3, pages 804–826 (page 12).

Nguyen, Xuanlong, Martin J. Wainwright, and Michael I. Jordan (2010). "Estimating divergence functionals and the likelihood ratio by convex risk minimization". In: *IEEE Transactions on Information Theory* 56.11, pages 5847–5861. arXiv: 0809.0853 (page 11).

Nielsen, Frank and Richard Nock (2011). "On Rényi and Tsallis entropies and divergences for exponential families". In: arXiv: 1105.3259 (page 9).

Niranjan, Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger (2010). "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". In: *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)* (page 32).

Ntampaka, Michelle, Hy Trac, Dougal J. Sutherland, Nicholas Battaglia, Barnabás Póczos, and Jeff Schneider (2014). "A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters". In: *The Astrophysical Journal* 803.2, page 50. arXiv: 1410.0686 (pages 3, 29).

Ntampaka, Michelle, Hy Trac, Dougal J. Sutherland, Sebastian Fromenteau, Barnabás Póczos, and Jeff Schneider (2015). "Dynamical Mass Measurements of Contaminated Galaxy Clusters Using Machine Learning". In: arXiv: 1509.05409 (page 30).

Oliva, Aude and Antonio Torralba (2001). "Modeling the shape of the scene: a holistic representation of the spatial envelope". In: *International Journal of Computer Vision* 42.3 (page 27).

Oliva, Junier B., Willie Neiswanger, Barnabás Póczos, Jeff Schneider, and Eric Xing (2014). "Fast Distribution To Real Regression". In: *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*. arXiv: 1311.2236 (pages 2, 20, 25).

Oliva, Junier B., Barnabás Póczos, and Jeff Schneider (2013). "Distribution to distribution regression". In: *Proceedings of The 30th International Conference on Machine Learning* (page 5).

Osborne, Michael A., Roman Garnett, and Stephen J. Roberts (2009). "Gaussian Processes for Global Optimization". In: *Proceedings of the 3rd Learning and Intelligent Optimization Conference (LION 3)* (page 32).

Pedregosa, F., G. Varoquaux, A. Gramfort, et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pages 2825–2830 (page 3).

Perlman, Eric, Randal Burns, Yi Li, and Charles Meneveau (2007). "Data Exploration of Turbulence Simulations using a Database Cluster". In: *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing* (page 36).

Póczos, Barnabás, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman (2013). "Distribution-Free Distribution Regression". In: *Artificial Intelligence and Statistics*. AISTATS. arXiv: 1302.0082 (page 5).

Póczos, Barnabás and Jeff Schneider (2011). "On the Estimation of α-Divergences". In: *International Conference on Artificial Intelligence and Statistics* (page 10).

Póczos, Barnabás, Liang Xiong, and Jeff Schneider (2011). "Nonparametric Divergence Estimation with Applications to Machine Learning on Distributions". In: *Uncertainty in Artificial Intelligence* (page 5).

Póczos, Barnabás, Liang Xiong, Dougal J. Sutherland, and Jeff Schneider (2012). "Nonparametric kernel estimators for image classification". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2989–2996 (pages 3, 6, 10, 27, 28).

Qin, Jianzhao and Nelson H.C. Yung (2010). "SIFT and color feature fusion using localized maximum-margin learning for scene classfication". In: *International Conference on Machine Vision* (page 28).

Rahimi, Ali and Benjamin Recht (2007). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. MIT Press (pages 2, 15, 16, 18).

Ramdas, Aaditya and Leila Wehbe (2014). "Nonparametric Independence Testing for Small Sample Sizes". In: arXiv: 1406.1922 (page 11).

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press (page 17).

Saatçi, Yunus (2011). "Scalable Inference for Structured Gaussian Process Models". PhD thesis. University of Cambridge (page 21).

Schoenberg, I. J. (1938). "Metric spaces and positive definite functions". In: *Transactions of the American Mathematical Society* 44.3, pages 522–536 (page 11).

Schwarz, Gideon (1978). "Estimating the Dimension of a Model". In: *Ann. Statist.* 6.2, pages 461–464 (page 25).

Settles, Burr (2012). *Active Learning*. Morgan & Claypool (page 32).

Shirdhonkar, Sameer and David W. Jacobs (2008). "Approximate earth mover's distance in linear time". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8 (page 10).

Simonyan, Karen and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *ICLR*. arXiv: 1409.1556 (page 29).

Singh, Shashank and Barnabás Póczos (2014). "Exponential Concentration of a Density Functional Estimator". In: *Advances in Neural Information Processing Systems*, pages 3032–3040 (page 10).

Sriperumbudur, Bharath K., Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf (2009). "Kernel choice and classifiability for RKHS embeddings of probability distribu-

tions". In: *Advances in Neural Information Processing Systems*. Volume 22. MIT Press, pages 1750–1758 (pages 41, 42).

Sriperumbudur, Bharath K., Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet (2009). "On integral probability metrics, ϕ-divergences and binary classification". In: arXiv: 0901.2698 (page 6).

— (2012). "On the empirical estimation of integral probability metrics". In: *Electronic Journal of Statistics* 6, pages 1550–1599 (page 11).

Sriperumbudur, Bharath K., Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet (2010). "Hilbert space embeddings and metrics on probability measures". In: *Journal of Machine Learning Research* 11, pages 1517–1561. arXiv: 0907.5309 (pages 9, 42).

Sriperumbudur, Bharath K. and Zoltán Szabó (2015). "Optimal Rates for Random Fourier Features". In: arXiv: 1506.02155 (pages 18, 19).

Stein, Charles (1956). "Inadmissibility of the Usual Estimator for the Mean of a Multi-Variate Normal Distribution". In: *Proc. Third Berkeley Symp. Math. Statist. Prob* 1.4, pages 197–206 (page 11).

Sutherland, Dougal J., Junier B. Oliva, Barnabás Póczos, and Jeff Schneider (2015). *Linear-Time Learning on Distributions with Approximate Kernel Embeddings*. arXiv: 1509.07553 (pages 2, 3, 19, 22, 23, 25, 28).

Sutherland, Dougal J. and Jeff Schneider (2015). "On the Error of Random Fourier Features". In: *Uncertainty in Artificial Intelligence*. arXiv: 1506.02785 (pages 2, 16, 18, 19).

Sutherland, Dougal J., Liang Xiong, Barnabás Póczos, and Jeff Schneider (2012). "Kernels on Sample Sets via Nonparametric Divergence Estimates". In: arXiv: 1202.0302 (pages 27, 36, 37).

Szabó, Zoltán, Bharath Sriperumbudur, Barnabás Póczos, and Arthur Gretton (2014). "Learning Theory for Distribution Regression". In: *Artificial Intelligence and Statistics*. AISTATS. arXiv: 1411.2066 (page 9).

Tesch, Matthew, Jeff Schneider, and Howie Choset (2013). "Expensive function optimization with stochastic binary outcomes". In: *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)* (page 32).

Turian, Joseph, Lev Ratinov, and Yoshua Bengio (2010). "Word representations: A simple and general method for semi-supervised learning". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (page 40).

Valada, Abhinav, Christopher Tomaszewski, Balajee Kannan, Prasanna Velagapudi, George Kantor, and Paul Scerri (2012). "An Intelligent Approach to Hysteresis Compensation while Sampling Using a Fleet of Autonomous Watercraft". In: *Intelligent Robotics and Applications*. Volume 7507. Lecture Notes in Computer Science (page 31).

Vedaldi, Andrea and Brian Fulkerson (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/ (pages 26, 27).

Vedaldi, Andrea and Andrew Zisserman (2012). "Efficient additive kernels via explicit feature maps". In: *IEEE transactions on pattern analysis and machine intelligence* 34.3, pages 480–92 (pages 22, 25).

Vilnis, Luke and Andrew McCallum (2015). "Word Representations via Gaussian Embedding". In: *International Conference on Learning Representations*. arXiv: 1412.6623 (page 40).

Wang, Fei, Tanveer Syeda-Mahmood, Baba C. Vemuri, David Beymer, and Anand Rangarajan (2009). "Closed-Form Jensen-Renyi Divergence for Mixture of Gaussians and Applications to Group-Wise Shape Registration". In: *Med Image Comput Comput Assist Interv.* 12.1, pages 648–655 (page 9).

Wang, Qing, Sanjeev R Kulkarni, and Sergio Verdú (2009). "Divergence Estimation for Multidimensional Densities Via k-Nearest-Neighbor Distances". In: *IEEE Transactions on Information Theory* 55.5, pages 2392–2405 (pages 10, 25, 29).

Wu, Jianxin, Bin-Bin Gao, and Guoqing Liu (2015). “Visual Recognition Using Directional Distribution Distance”. In: arXiv: 1504.04792 (pages 28, 29).

Xiong, Liang (2013). “On Learning from Collective Data”. PhD thesis. Carnegie Mellon University (page 13).

Yang, Kun, Hao Su, and Wing Hung Wong (2014). “co-BPM: a Bayesian Model for Estimating Divergence and Distance of Distributions”. In: arXiv: 1410.0726 (page 11).

Yang, Zichao, Alexander J Smola, and Andrew Gordon Wilson (2015). “A la Carte — Learning Fast Kernels”. In: *AISTATS* (page 41).

Yoshikawa, Yuya, Tomoharu Iwata, and Hiroshi Sawada (2014). “Latent Support Measure Machines for Bag-of-Words Data Classification”. In: *Advances in Neural Information Processing Systems*, pages 1961–1969 (page 40).

— (2015). “Non-Linear Regression for Bag-of-Words Data via Gaussian Process Latent Variable Set Model”. In: *AAAI*, pages 3129–3135 (page 40).

Zhang, J, M Marszałek, S Lazebnik, and C Schmid (2006). “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study”. In: *International Journal of Computer Vision* 73.2, pages 213–238 (page 12).

Zhao, Ji and Deyu Meng (2014). “FastMMD: Ensemble of Circular Discrepancy for Efficient Two-Sample Test”. In: arXiv: 1405.2664 (page 19).

Zhou, Bolei, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva (2014). “Learning Deep Features for Scene Recognition using Places Database”. In: *NIPS* (page 29).

Zwicky, Fritz (1933). “Die Rotverschiebung von extragalaktischen Nebeln”. German. In: *Helvetica Physica Acta* 6, pages 110–127 (page 29).