

# New Directions for Learning with Kernels and Gaussian Processes

Edited by

Arthur Gretton<sup>1</sup>, Philipp Hennig<sup>2</sup>, Carl Edward Rasmussen<sup>3</sup>, and Bernhard Schölkopf<sup>4</sup>

1 University College London, GB, [arthur.gretton@gmail.com](mailto:arthur.gretton@gmail.com)

2 MPI für Intelligente Systeme – Tübingen, DE, [ph@tue.mpg.de](mailto:ph@tue.mpg.de)

3 University of Cambridge, GB, [cer54@cam.ac.uk](mailto:cer54@cam.ac.uk)

4 MPI für Intelligente Systeme – Tübingen, DE, [bs@tue.mpg.de](mailto:bs@tue.mpg.de)

---

## Abstract

The Dagstuhl Seminar on 16481 “New Directions for Learning with Kernels and Gaussian Processes” brought together two principal theoretical camps of the machine learning community at a crucial time for the field. Kernel methods and Gaussian process models together form a significant part of the discipline’s foundations, but their prominence is waning while more elaborate but poorly understood hierarchical models are ascendant. In a lively, amiable seminar, the participants re-discovered common conceptual ground (and some continued points of disagreement) and productively discussed how theoretical rigour can stay relevant during a hectic phase for the subject.

**Seminar** November 27 to December 2, 2016 – <http://www.dagstuhl.de/16481>

**1998 ACM Subject Classification** G.1.1 Interpolation, G.1.2 Approximation, G.3 Probability and Statistics, I.2.6 Learning

**Keywords and phrases** gaussian processes, kernel methods, machine learning, probabilistic numerics, probabilistic programming

**Digital Object Identifier** 10.4230/DagRep.6.11.142

**Edited in cooperation with** Michael Schober


## 1 Summary

*Arthur Gretton*

*Philipp Hennig*

*Carl Edward Rasmussen*

*Bernhard Schölkopf*

**License**  Creative Commons BY 3.0 Unported license

© Arthur Gretton, Philipp Hennig, Carl Edward Rasmussen, and Bernhard Schölkopf

Machine learning is a young field that currently enjoys rapid, almost dizzying advancement both on the theoretical and the practical side. On account of either, the until quite recently obscure discipline is increasingly turning into a central area of computer science. Dagstuhl seminar 16481 on “*New Directions for Learning with Kernels and Gaussian Processes*” attempted to allow a key community within machine learning to gather its bearings at this crucial moment in time.

Positive definite kernels are a concept that dominated machine learning research in the first decade of the millennium. They provide infinite-dimensional hypothesis classes



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

New Directions for Learning with Kernels and Gaussian Processes, *Dagstuhl Reports*, Vol. 6, Issue 11, pp. 142–167

Editors: Arthur Gretton, Philipp Hennig, Carl Edward Rasmussen, and Bernhard Schölkopf



DAGSTUHL  
REPORTS

Dagstuhl Reports  
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that deliver expressive power in an elegant analytical framework. In their probabilistic interpretation as Gaussian process models, they are also a fundamental concept of Bayesian inference:

A *positive definite kernel*  $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  on some input domain  $\mathbb{X}$  is a function with the property that, for all finite sets  $\{x_1, \dots, x_N\} \subset \mathbb{X}$ , the matrix  $K \in \mathbb{R}^{N \times N}$ , with elements  $k_{ij} = k(x_i, x_j)$ , is positive semidefinite. According to a theorem by Mercer, given certain regularity assumptions, such kernels can be expressed as a potentially *infinite* expansion

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i^*(x'), \quad \text{with} \quad \sum_{i=1}^{\infty} \lambda_i < \infty, \quad (1)$$

where  $*$  is the conjugate transpose,  $\lambda_i \in \mathbb{R}_+$  is a non-negative *eigenvalue* and  $\phi_i$  is an *eigenfunction* with respect to some measure  $\nu(x)$ : a function satisfying

$$\int k(x, x') \phi_i(x) d\nu(x) = \lambda_i \phi_i(x'). \quad (2)$$

Random functions  $f(x)$  drawn by independently sampling Gaussian weights for each eigenfunction,

$$f(x) = \sum_{j=1}^{\infty} f_j \phi_j(x) \quad \text{where} \quad f_j \sim \mathcal{N}(0, \lambda_j), \quad (3)$$

are draws from the centered *Gaussian process* (GP)  $p(f) = \mathcal{GP}(f; 0, k)$  with *covariance function*  $k$ . The logarithm of this Gaussian process measure is, up to constants and some technicalities, the square of the norm  $\|f\|_k^2$  associated with the *reproducing kernel Hilbert space* (RKHS) of functions reproduced by  $k$ .

Supervised machine learning methods that *infer* an unknown function  $f$  from a data set of input-output pairs  $(X, Y) := \{(x_i, y_i)\}_{i=1, \dots, N}$  can be constructed by minimizing an empirical risk  $\ell(f(X); Y)$  regularized by  $\|\cdot\|_k^2$ . Or, algorithmically equivalent but with different philosophical interpretation, by computing the *posterior* Gaussian process measure arising from conditioning  $\mathcal{GP}(f; 0, k)$  on the observed data points under a likelihood proportional to the exponential of the empirical risk.

The prominence of kernel/GP models was founded on this conceptually and algorithmically compact yet statistically powerful description of inference and learning of nonlinear functions. In the past years, however, hierarchical ('deep') parametric models have bounced back and delivered a series of impressive empirical successes. In areas like speech recognition and image classification, deep networks now far surpass the predictive performance previously achieved with nonparametric models. One central goal of the seminar was to discuss how the superior adaptability of deep models can be transferred to the kernel framework while retaining at least some analytical clarity. Among the central lessons from the 'deep resurgence' identified by the seminar participants is that the kernel community has been too reliant on theoretical notions of universality. Instead, representations must be learned on a more general level than previously accepted. This process is often associated with an 'engineering' approach to machine learning, in contrast to the supposedly more 'scientific' air surrounding kernel methods. But its importance must not be dismissed. At the same time, participants also pointed out that deep learning is often misrepresented, in particular in popular expositions, as an almost magic kind of process; when in reality the concept is closely related to kernel methods, and can be understood to some degree through this connection: Deep models provide a hierarchical parametrization of the feature functions  $\phi_i(x)$  in terms of a finite-dimensional family. The continued relevance of the established theory for kernel/GP models

hinges on how much of the power of deep models can be understood from within the RKHS view, and how much new concepts are required to understand the expressivity of a deep learning machine.

There is also unconditionally good news: In a separate but related development, kernels have had their own renaissance lately, in the young areas of probabilistic programming (‘computing of probability measures’) and probabilistic numerics (‘probabilistic descriptions of computing’). In both areas, kernels and Gaussian processes have been used as a descriptive language. And, similar to the situation in general machine learning, only a handful of comparably simple kernels have so far been used. The central question here, too, is thus how kernels can be designed for challenging, in particular high-dimensional regression problems. In contrast to the wider situation in ML, though, kernel design here should take place at compile-time, and be a structured algebraic process mapping source code describing a graphical model into a kernel. This gives rise to new fundamental questions for the theoretical computer science of machine learning.

A third thread running through the seminar concerned the internal conceptual schism between the probabilistic (Gaussian process) view and the statistical learning theoretical (RKHS) view on the model class. Although the algorithms and algebraic ideas used on both sides overlap *almost* to the point of equivalence, their philosophical interpretations, and thus also the required theoretical properties, differ strongly. Participants for the seminar were deliberately invited from both “denominations” in roughly equal number. Several informal discussions in the evenings, and in particular a lively break-out discussion on Thursday helped clear up the mathematical connections (while also airing key conceptual points of contention from either side). Thursday’s group is planning to write a publication based on the results of the discussion; this would be a highly valuable concrete contribution arising from the seminar, that may help drawing this community closer together.

Despite the challenges to some of the long-standing paradigms of this community, the seminar was infused with an air of excitement. The participants seemed to share the sensation that machine learning is still only just beginning to show its full potential. The mathematical concepts and insights that have emerged from the study of kernel/GP models may have to evolve and be adapted to recent developments, but their fundamental nature means they are quite likely to stay relevant for the understanding of current and future model classes. Far from going out of fashion, mathematical analysis of the statistical and numerical properties of machine learning model classes seems slated for a revival in coming years. And much of it will be leveraging the notions discussed at the seminar.

## 2 Table of Contents

### Summary

*Arthur Gretton, Philipp Hennig, Carl Edward Rasmussen, and Bernhard Schölkopf* 142

### Overview of Talks

Random Fourier Features for Operator-Valued Kernels <i>Florence d’Alché-Buc</i> . . . . .	147
Practical Challenges of Gaussian Process Applications <i>Marc Deisenroth</i> . . . . .	147
Deep kernels and deep Gaussian processes <i>David Duvenaud</i> . . . . .	148
Finding Galaxies in the Shadows of Quasars with Gaussian Processes <i>Roman Garnett</i> . . . . .	148
Comparing samples from two distributions <i>Arthur Gretton</i> . . . . .	149
GPs and Kernels for Computation – new opportunities in probabilistic numerics <i>Philipp Hennig</i> . . . . .	150
GPy and GPFlow <i>James Hensman</i> . . . . .	150
Approximate EP for Deep Gaussian Processes <i>Joseé Miguel Hernández-Lobato</i> . . . . .	151
Modelling Challenges in AutoML <i>Frank Hutter</i> . . . . .	151
Convergence guarantees for kernel-based quadrature <i>Motonobu Kanagawa</i> . . . . .	152
Don’t Panic: Deep Learning Methods are Mostly Harmless <i>Neil D. Lawrence</i> . . . . .	152
Horses <i>David Lopez-Paz</i> . . . . .	153
Kernel Learning with Convolutional Kernel Networks <i>Julien Mairal</i> . . . . .	153
Shrinkage Estimators <i>Krikamol Muandet</i> . . . . .	154
MMD/VAE/f-GAN: Methods for Estimating Probabilistic Models <i>Sebastian Nowozin</i> . . . . .	154
Score matching and kernel based estimators for the drift of stochastic differential equations <i>Manfred Opper</i> . . . . .	155
Gaussian Processes – Past and Future? <i>Carl Edward Rasmussen</i> . . . . .	155
How to fit a simple model <i>Carl Edward Rasmussen</i> . . . . .	156

String Gaussian Processes & Generalized Spectral Kernels <i>Stephen Roberts</i> . . . . .	156
Kernels – Past and Future? <i>Bernhard Schölkopf</i> . . . . .	156
Kernel Embeddings and Bayesian Quadrature <i>Dino Sejdinovic</i> . . . . .	157
Kernel Mean Embeddings <i>Carl-Johann Simon-Gabriel</i> . . . . .	157
Random Fourier Features and Beyond <i>Bharath Sriperumbudur</i> . . . . .	158
Learning with Hierarchical Kernels <i>Ingo Steinwart</i> . . . . .	158
Distribution Regression <i>Zoltán Szabó</i> . . . . .	159
Stochastic (partial) differential equations and Gaussian processes <i>Simo Särkkä</i> . . . . .	159
Consistent Kernel Mean Estimation for Functions of Random Variables <i>Ilya Tolstikhin</i> . . . . .	160
Uncertain inputs in Gaussian Processes <i>Mark van der Wilk</i> . . . . .	160
Frequentist properties of GP learning methods <i>Harry van Zanten</i> . . . . .	160
The ML Invasion of ABC <i>Richard Wilkinson</i> . . . . .	161
<b>Working groups</b>	
Generative Models <i>David Duvenaud</i> . . . . .	162
Limitations of GPs / non-Gaussian-Processes <i>Stefan Harmeling</i> . . . . .	163
The separation between Kernels/GPs and Deep Learning <i>Sebastian Nowozin</i> . . . . .	164
Connections and Differences between Kernels and GPs <i>Dino Sejdinovic</i> . . . . .	166
<b>Participants</b> . . . . .	167

## 3 Overview of Talks

### 3.1 Random Fourier Features for Operator-Valued Kernels

*Florence d'Alché-Buc (Telecom ParisTech, FR)*

**License** © Creative Commons BY 3.0 Unported license  
© Florence d'Alché-Buc

**Joint work of** Romain Brault and Florence d'Alché-Buc

**Main reference** R. Brault, M. Heinonen, F. d'Alché Buc, “Random Fourier Features For Operator-Valued Kernels”, in Proc. of the 8th Asian Conf. on Machine Learning, JMLR W&CP, Vol. 63, 2016.

**URL** <http://www.jmlr.org/proceedings/papers/v63/Brault39.pdf>

Devoted to multi-task learning and structured output learning, operator-valued kernels provide a flexible tool to build vector-valued functions in the context of Reproducing Kernel Hilbert Spaces. To scale up operator-valued kernel-based regression devoted to multi-task and structured output learning, we extend the celebrated Random Fourier Feature methodology to get an approximation of operator-valued kernels. We propose a general principle for Operator-valued Random Fourier Feature construction relying on a generalization of Bochner’s theorem for shift-invariant operator-valued Mercer kernels. We prove the uniform convergence of the kernel approximation for bounded and unbounded operator random Fourier features using appropriate Bernstein matrix concentration inequality. Numerical experiments show the quality of the approximation and the efficiency of the corresponding linear models on multiclass and regression problems.

### 3.2 Practical Challenges of Gaussian Process Applications

*Marc Deisenroth (Imperial College London, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Marc Deisenroth

**Main reference** G. Bertone, M. P. Deisenroth, J. S. Kim, S. Liem, R. Ruiz de Austri, M. Welling, “Accelerating the BSM interpretation of LHC data with machine learning”, arXiv:1611.02704 [hep-ph], 2016.

**URL** <https://arxiv.org/abs/1611.02704v1>

In many applications, we face practical challenges with Gaussian processes and kernel methods. For example, in robotics and personalized healthcare, data-efficient learning (i.e., learning from small data sets) is critical. We can achieve this in multiple ways, e.g., by carefully modeling uncertainty in the model and the inference, transfer learning or the incorporation of structural priors. Focusing on uncertainty representation, it is critical to propagate uncertainty through a (Gaussian process) system, which is computationally expensive (training may not be the computational bottleneck). Other applications include the optimization (or learning) of simulators of very expensive experiments (e.g., LHC, bioprocesses or neotissue engineering). Challenge we face are high-dimensional optimization problems and scalability in the number of data points. Generally, scalability seems to be a general problem, and we should think about scale-free model architectures, inference and the software that allows us to perform distributed computing.

### 3.3 Deep kernels and deep Gaussian processes

*David Duvenaud (Toronto, CA)*

**License** © Creative Commons BY 3.0 Unported license  
© David Duvenaud

**Joint work of** David Duvenaud, Oren Rippel, Ryan P. Adams, Zoubin Ghahramani

**Main reference** D. Duvenaud, O. Rippel, R. P. Adams, Z. Ghahramani, “Avoiding pathologies in very deep networks”, arXiv:1402.5836 [stat.ML], 2014.

**URL** <https://arxiv.org/abs/1402.5836v3>

To suggest better neural network architectures, we analyze the properties different priors on compositions of functions.

We showed how we can construct deep kernels by composing their implicit features, and examine the properties of such kernels as we increase their depth.

We then showed how such models are different from deep Gaussian processes, and by visualizing draws from deep GP priors examined their properties as a function of depth.

Finally, we show that you get additive covariance if you do dropout on Gaussian processes.

### 3.4 Finding Galaxies in the Shadows of Quasars with Gaussian Processes

*Roman Garnett (Washington University – St. Louis, US)*

**License** © Creative Commons BY 3.0 Unported license  
© Roman Garnett

**Joint work of** Roman Garnett, Shirley Ho, Jeff Schneider

**Main reference** R. Garnett, S. Ho, J. Schneider, “Detecting Damped Lyman- $\alpha$  Absorbers with Gaussian Processes”, arXiv:1605.04460 [astro-ph.CO], 2016.

**URL** <https://arxiv.org/abs/1605.04460v1>

I discussed recent application of Gaussian processes to a problem from astrophysics: detecting damped Lyman- $\alpha$  absorbers in lines of sight to quasars. DLAs represent proto-galaxies in the ancient universe and their distribution is of interest to cosmology. The state of the art for detecting DLAs is visual inspection; however we show we can construct an automated method via Bayesian model selection, with GPs as our models of spectroscopic data. We use a dataset of  $\sim 50,000$  quasar observations from SDSS-III to derive a custom “quasar kernel”. The learned kernel has structure markedly different from off-the-shelf kernels. Performance on the detection task relied critically on this structure. Finally, I pointed out I had to rely on quasi-Monte Carlo to estimate model evidence for the DLA model because the integrand had dynamic range on the order of  $\sim 6000$  nats. No off-the-shelf model can handle such data.

### 3.5 Comparing samples from two distributions

*Arthur Gretton (University College London, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Arthur Gretton

**Joint work of** Kacper Chwialkowski, Arthur Gretton, Wittawat Jitkrittum, Dino Sejdinovic, Bharath Sriperumbudur, Heiko Strathmann, Dougal Sutherland, Zoltán Szabo

**Main reference** W. Jitkrittum, Z. Szabo, K. Chwialkowski, A. Gretton, “Interpretable Distribution Features with Maximum Testing Power”, arXiv:1605.06796v2 [stat.ML], 2016.

**URL** <https://arxiv.org/abs/1605.06796v2>

We provide an overview of kernel approaches to comparing distributions. The focus is on choosing the function class, and adapting the test statistic, so as to maximize the power of the associated tests.

We begin with an introduction to embeddings of probabilities to a reproducing kernel Hilbert space (RKHS), where an embedding is simply the expectation of the kernel function that defines the RKHS. We demonstrate that the difference in these embeddings can be interpreted as an integral probability metric, called the Maximum Mean Discrepancy (MMD). This statistic can be used in a test of homogeneity, where two samples are observed, and the null hypothesis is that both samples are drawn from the same distribution.

The power of a statistical test based on the MMD will depend on the particular RKHS used. We show that the asymptotic distribution of the statistic is Gaussian under the alternative, and an infinite sum of weighted chi squared variables under the null. Since the null distribution has faster shrinking variance, it is shown that the kernel maximizing the test power is the one which gives the largest ratio of the MMD to its variance (the optimization is performed on a held-out validation set). We demonstrate that this optimized kernel can distinguish between samples from a generative adversarial network, and samples drawn from a reference test set.

An alternative approach to homogeneity testing is to look for maximum of the witness function associated with the MMD, which is a smooth function with largest amplitude where the probability mass of  $P$  and  $Q$  is most different. We can therefore use the values of the witness function at a particular set of points to construct a test statistic. Our statistic involves normalizing these witness function values by their joint covariance. We may optimize a lower bound on the test power by maximizing the test statistic over the witness point locations on a held-out validation set. We use this test to distinguish positive and negative emotions on a facial expression database, showing that a distinguishing feature reveals the facial areas most relevant to emotion.

Finally, we address the problem of comparing a model to a sample, for instance in the context of statistical model criticism. In this case, the MMD witness function can be modified by a Stein operator, to have zero expectation under the model distribution. The resulting statistic is denoted the Maximum Stein Discrepancy (MSD). This Stein operator can be computed even when the distribution cannot be normalized. We use the MSD to demonstrate the inadequacy of fit of a simple regression model to data with heteroscedastic noise.



### 3.6 GPs and Kernels for Computation – new opportunities in probabilistic numerics

*Philipp Hennig (MPI für Intelligente Systeme – Tübingen, DE)*

**License** © Creative Commons BY 3.0 Unported license

© Philipp Hennig

**Main reference** P. Hennig, M. Osborne, M. Girolami, “Probabilistic numerics and uncertainty in computations,” Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 471(2179), 2015.

**URL** <http://dx.doi.org/10.1098/rspa.2015.0142>

**URL** <http://probabilistic-numerics.org>

The computational complexity of machine learning models (not just kernel/GP models) is dominated by numerical tasks: optimization, integration, linear algebra, and the solution of differential algebra. The algorithms we use for these tasks have mostly arrived in our community from other disciplines, such as computational physics, and simulation. Interestingly, these methods can actually be interpreted as active learning algorithms themselves, since they estimate latent/incomputable quantities (e.g. the value of an integral) from observable/computable quantities (e.g. values of the integrand at various, actively chosen nodes). Over recent years, this observation has given rise to a class of numerical methods known as probabilistic numerical algorithms: Methods that take in and return probability measures, rather than point estimates. A string of papers have revealed that many popular and foundational numerical methods can be written as least-squares regression, and thus interpreted as MAP estimators arising from Gaussian probabilistic models. Careful analysis shows that the associated posterior variances can be calibrated at low computational cost, meaning that they provide a meaningful notion of uncertainty in computation. Now, this new framework can be used to build new functionality sorely needed in machine learning: Increased performance through custom prior assumptions; stability of computations under stochastic computations, and new notions of algorithmic safety through statistical hypothesis testing.

### 3.7 GPy and GPFlow

*James Hensman (Lancaster University, GB)*

**License** © Creative Commons BY 3.0 Unported license

© James Hensman

**Joint work of** James Hensman, Alex Matthews, Mark van der Wilk, Neil Lawrence, Max Ziwiessle and others  
**Main reference** A. G. de G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrà, Z. Ghahramani, J. Hensman, “GPflow: A Gaussian process library using TensorFlow”, arXiv:1610.08733 [stat.ML], 2016.

**URL** <https://arxiv.org/abs/1610.08733>

**URL** <https://www.github.com/SheffieldML/GPy>

In this talk, I present a live demo of working with the Python-based frameworks GPy and GPFlow.

Some discussion has arisen surrounding the reasons for the success of Deep Learning, and one of the contributing factors is widely agreed to be the availability of Deep Learning software. In this talk I argue that Deep Learning software can easily be adapted to suit kernel methods.

I describe how reverse mode differentiation of the Cholesky algorithm has been added to TensorFlow by Alex Matthews and myself. I then describe GPy and GPflow, two frameworks for Gaussian process computation.

I also present a live demo designed to introduce the audience to the concepts needed to understand TensorFlow, and how to adapt it to their own needs and projects.

### 3.8 Approximate EP for Deep Gaussian Processes

*José Miguel Hernández-Lobato (Harvard University – Cambridge, US)*

**License** © Creative Commons BY 3.0 Unported license  
© José Miguel Hernández-Lobato

**Main reference** T. D. Bui, D. Hernández-Lobato, Y. Li, J. M. Hernández-Lobato, and R. E. Turner, “Deep Gaussian Processes for Regression using Approximate Expectation Propagation”, in Proc. of the 33rd Int’l Conf. on Machine Learning (JMLR 2016), Vol. 48, pp. 1472–1481, 2016.

**URL** <http://jmlr.org/proceedings/papers/v48/bui16.pdf>

Deep Gaussian processes (DGPs) are multi-layer hierarchical generalisations of Gaussian processes (GPs) and are formally equivalent to neural networks with multiple, infinitely wide hidden layers. DGPs are nonparametric probabilistic models and as such are arguably more flexible, have a greater capacity to generalise, and provide better calibrated uncertainty estimates than alternative deep models. We develop a new approximate Bayesian learning scheme that enables DGPs to be applied to a range of medium to large scale regression problems for the first time. The new method uses an approximate Expectation Propagation procedure and a novel and efficient extension of the probabilistic backpropagation algorithm for learning. We evaluate the new method for non-linear regression on eleven real-world datasets, showing that it always outperforms GP regression and is almost always better than state-of-the-art deterministic and sampling-based approximate inference methods for Bayesian neural networks.

### 3.9 Modelling Challenges in AutoML

*Frank Hutter (Universität Freiburg, DE)*

**License** © Creative Commons BY 3.0 Unported license  
© Frank Hutter

**Main reference** M. Feurer, A. Klein, K. Eggenberger, J. Springenberg, M. Blum, F. Hutter, “Efficient and Robust Automated Machine Learning”, in Proc. of Advances in Neural Information Processing Systems 28 (NIPS 2015), pp. 2962–2970, Neural Information Processing Systems Foundation Inc., 2015.

**URL** <https://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning>

**URL** <http://www.ml4aad.org>

In this talk, I briefly overviewed recent developments in the field of automated machine learning, which gives rise to very popular applications of Gaussian processes in Bayesian optimization. I then discussed some of the modelling challenges that occur in this field (such as high dimensionality, conditional spaces, large number of data points, heteroscedasticity, large noise, modelling across data sets, and modelling of learning curves) and initial solutions; some of these solutions were based on Gaussian processes, and some were based on random forests and Bayesian neural networks. We then discussed the challenges of treating all of these problems using Gaussian processes.

### 3.10 Convergence guarantees for kernel-based quadrature

*Motonobu Kanagawa (Institute of Statistical Mathematics – Tokyo, JP)*

**License** © Creative Commons BY 3.0 Unported license  
© Motonobu Kanagawa

**Joint work of** Motonobu Kanagawa, Bharath Sriperumbudur, Kenji Fukumizu  
**Main reference** M. Kanagawa, B. K. Sriperumbudur, K. Fukumizu, “Convergence guarantees for kernel-based quadrature rules in misspecified settings”, in Proc. of Advances in Neural Information Processing Systems 29 (NIPS 2016), pp. 3288–3296, Neural Information Processing Systems Foundation Inc., 2016.

**URL** <http://papers.nips.cc/paper/6174-convergence-guarantees-for-kernel-based-quadrature-rules-in-misspecified-settings>

In this talk, I present recent results on kernel-based quadrature. Kernel-based quadrature rules are becoming important in machine learning and statistics, as they achieve super- $\sqrt{n}$  convergence rates in numerical integration, and thus provide alternatives to Monte Carlo integration in challenging settings where integrands are expensive to evaluate or where integrands are high dimensional. These rules are based on the assumption that the integrand has a certain degree of smoothness, which is expressed as that the integrand belongs to a certain reproducing kernel Hilbert space (RKHS). However, this assumption can be violated in practice (e.g., when the integrand is a black box function), and no general theory has been established for the convergence of kernel quadratures in such misspecified settings. In this talk, I explain that it is actually possible to prove that kernel quadratures can be consistent even when the integrand does not belong to the assumed RKHS, i.e., when the integrand is less smooth than assumed. Specifically, I show that one can derive convergence rates that depend on the (unknown) lesser smoothness of the integrand, where the degree of smoothness is expressed via powers of RKHSs or via Sobolev spaces.

### 3.11 Don’t Panic: Deep Learning Methods are Mostly Harmless

*Neil D. Lawrence (University of Sheffield, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Neil D. Lawrence

**URL** <http://inverseprobability.com/2016/11/29/new-directions-in-kernels-and-gaussian-processes.html>

With the success of deep learning software and a wide variety of successful applications deep learning methods seem to be making the transition to a domain of engineering. A challenge is that the potential pitfalls of the deployment of these ideas has not been characterised. Currently empirical results are leading our theoretical understanding. To be a robust engineering discipline deep learning pipelines need to be placed on stronger theoretical foundations. This presents an opportunity for better characterized methods to augment deep learning ideas and prevent us from succumbing to the pitfalls. The greater interpretability of kernel and GP methods as well as their more elegant mathematical characterization present opportunities in

1. Meta learning and characterisation of the deep deep learning pipeline.
2. Privacy, fairness and transparency.
3. Integration of physical systems with data driven models.
4. Quantifying the value of data.

### 3.12 Horses

*David Lopez-Paz (Facebook – AI Research, US)*

**License** © Creative Commons BY 3.0 Unported license  
© David Lopez-Paz

I talked about “horses”, which are “systems that do not address the problem that they seem to be addressing”. In particular, our current machine learning solutions are horses, since they are vulnerable to slight mismatches between training and testing data (domain adaptation, adversarial perturbation, etc.).

Taming horses is the biggest challenge for machine learning. More specifically, machine learning has a predictive focus (minimize the loss  $L(y, y')$  between the true targets  $y = f(x)$  and our estimates  $y' = f'(x)$ ). This contrasts the scientific method, which explains “why” things happen in terms of mechanisms (minimize the loss  $L(f, f')$  between the true mechanism  $f$  and our estimate  $f'$ ).

Since correlation is to prediction what causation is to explanation, I propose to tame horses by developing machine learning algorithms that leverage only causal dependencies, and ignore confounding dependencies.

### 3.13 Kernel Learning with Convolutional Kernel Networks

*Julien Mairal (INRIA – Grenoble, FR)*

**License** © Creative Commons BY 3.0 Unported license  
© Julien Mairal

**Main reference** J. Mairal, “End-to-End Kernel Learning with Supervised Convolutional Kernel Networks”, in Proc. of Advances in Neural Information Processing Systems 29 (NIPS 2016), pp. 1399–1407, Neural Information Processing Systems Foundation Inc., 2016.

**URL** <http://papers.nips.cc/paper/6184-end-to-end-kernel-learning-with-supervised-convolutional-kernel-networks>

In this talk, we present a new image representation based on a multilayer kernel machine that performs end-to-end learning. Unlike traditional kernel methods, where the kernel is handcrafted or adapted to data in an unsupervised manner, we learn how to shape the kernel for a supervised prediction problem. We proceed by generalizing convolutional kernel networks, which originally provide unsupervised image representations, and we derive backpropagation rules to optimize model parameters. As a result, we obtain a new type of convolutional neural network with the following properties: (i) at each layer, learning filters is equivalent to optimizing a linear subspace in a reproducing kernel Hilbert space (RKHS), where we project data; (ii) the network may be learned with supervision or without; (iii) the model comes with a natural regularization function (the norm in the RKHS). We show that the method achieves reasonably competitive performance on some standard “deep learning” image classification datasets such as CIFAR-10 and SVHN, and also state-of-the-art results for image super-resolution, demonstrating the applicability of the approach to a large variety of image-related tasks.

### 3.14 Shrinkage Estimators

*Krikamol Muandet (Mahidol University, TH)*

**License** © Creative Commons BY 3.0 Unported license  
© Krikamol Muandet

**Joint work of** Krikamol Muandet, Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf  
**Main reference** K. Muandet, B. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, “Kernel mean shrinkage estimators”, *Journal of Machine Learning Research*, 17(48):1–41, 2016.

**URL** <http://jmlr.org/papers/v17/14-195.html>

A mean function in a reproducing kernel Hilbert space (RKHS), or a kernel mean, is central to kernel methods in that it is used by many classical algorithms such as kernel principle component analysis, and it also forms the core inference step of modern kernel methods that rely on embedding probability distributions in RKHSs. Given a finite sample, an empirical average has been used commonly as a standard estimator of the true kernel mean. Despite a widespread use of this estimator, we show that it can be improved thanks to the well-known Stein phenomenon. We propose a new family of estimators called kernel mean shrinkage estimators (KMSEs), which benefit from both theoretical justifications and good empirical performance. The results demonstrate that the proposed estimators outperform the standard one, especially in a “large  $d$ , small  $n$ ” paradigm.

### 3.15 MMD/VAE/f-GAN: Methods for Estimating Probabilistic Models

*Sebastian Nowozin (Microsoft Research UK – Cambridge, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Sebastian Nowozin

**Joint work of** Sebastian Nowozin, Ryota Tomioka, Botond Cseke, Diane Bouchacourt, Pawan Kumar  
**Main reference** S. Nowozin, B. Cseke, R. Tomioka, “f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization”, in *Proc. of Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pp. 271–279, Neural Information Processing Systems Foundation Inc., 2016.

**URL** <http://papers.nips.cc/paper/6066-f-gan-training-generative-neural-samplers-using-variational-divergence-minimization>

**Main reference** D. Bouchacourt, P. K. Mudigonda, S. Nowozin, “DISCO Nets : DISsimilarity COefficients Networks”, in *Proc. of Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pp. 352–360, Neural Information Processing Systems Foundation Inc., 2016.

**URL** <http://papers.nips.cc/paper/6066-f-gan-training-generative-neural-samplers-using-variational-divergence-minimization>

Estimating generative or discriminative probabilistic models is important in practical applications. To formalize estimation we can think of measures of discrepancy between two distributions: the model distribution and the unknown true distribution. The classes of discrepancy measures are: 1. integral probability metrics, 2. proper scoring rules, and 3. f-divergences. Integral probability metrics take the supremum of a difference of two expectations over a class of functions. Depending on the choice of function class this realizes metrics such as the total variation, maximum mean discrepancy, or the Wasserstein metric. If the function class is taken to be a RKHS the resulting metric is the kernel MMD. Proper scoring rules require a more detailed access to the model distribution through its density function or properties thereof. Typical examples are the likelihood, the Brier score, or Bernardo’s quadratic scoring rule. In some cases, taking the integral of a scoring rule yields an f-divergence. f-divergences require access to the density function of both the model distribution and the true distribution, which is not available. Recently a variational lower bound on f-divergences allows to circumvent this requirement by introducing an additional variational function. Training a generative model with this variational approach yields a

saddle-point problem to solve, an approach known as the generative-adversarial network (GAN) approach. These new approaches to estimating models in the likelihood-free setting have allowed new levels of performance in fitting complicated distributions such as learning distributions of natural images.

### 3.16 Score matching and kernel based estimators for the drift of stochastic differential equations

*Manfred Opper (TU Berlin, DE)*

**License** © Creative Commons BY 3.0 Unported license  
© Manfred Opper

**Joint work of** Philipp Batz, Andreas Ruttor, Manfred Opper

**Main reference** P. Batz, A. Ruttor, M. Opper, “Variational estimation of the drift for stochastic differential equations from the empirical density”, *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2016, Number 8, 2016.

**URL** <http://dx.doi.org/10.1088/1742-5468/2016/08/083404>

Score matching is a method for estimating the logarithms of a probability density (up to a constant) which is not based on a likelihood. Using a kernel method this approach has recently been generalised to nonparametric density estimation.

I show that this method relates to a drift estimation problem for certain classes of stochastic differential equations and can be generalised to treat interesting types of Langevin equations.

I also show that the kernel method can be understood as a proper Bayesian approach in the limit, where observations of the stochastic process are densely sampled in time.

### 3.17 Gaussian Processes – Past and Future?

*Carl Edward Rasmussen (University of Cambridge, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Carl Edward Rasmussen

**Main reference** C. E. Rasmussen, C. K. I. Williams, “Gaussian Processes for Machine Learning”, MIT Press, ISBN-10 0-262-18253-X, 2006.

**URL** <http://www.gaussianprocess.org/gpml>

I attempt to give an overview of challenges of GPs in the past and what the situation might look like in the future. I discussed 5 central theorems: 1) Uses of GPs; although GPs are often used simply to model functions, their central advantage are situations where the predictive error bars are central: probabilistic numerics, decision making, RL and active learning among others. 2) Practical considerations: providing good code/toolboxes and automation, covariance functions and inducing points. These questions haven’t really been addressed satisfactorily. 3) Computational considerations: these questions have largely been solved, especially inducing point methods are very good. 4) Covariance structures: we still don’t have a clear idea how to implement more sophisticated covariance functions, or how practically to do inference when the number of hyperparameters (statistically) prohibit ML type 2 treatment. 5) Towards the future: can we construct little Lego brick GPs which can take probabilistic inputs and can be assembled as stacked into useful structures?

### 3.18 How to fit a simple model

*Carl Edward Rasmussen (University of Cambridge, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Carl Edward Rasmussen

In this talk, I present a paradox of suboptimal learning when fitting parameters of a simple model class. We show how it can be beneficial to learn a complex model which is then projected onto the simple model class rather than directly map from data to parameters.

### 3.19 String Gaussian Processes & Generalized Spectral Kernels

*Stephen Roberts (University of Oxford, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Stephen Roberts

**Joint work of** Yves-Laurent Kom Samo, Stephen Roberts  
**Main reference** Y. L. Kom Samo, S. Roberts, “String and Membrane Gaussian Processes”, *Journal of Machine Learning Research*, 17(131):1–87, 2016.  
**URL** <http://jmlr.org/papers/v17/15-382.html>

In this talk we introduce ways of invoking highly non-stationary kernels. String GPs allow for a domain to be broken into a series of conditionally independent Gaussian Processes, which merely ensure continuity in  $f$  and  $f'$  at the boundaries. We show how this allows for not just non-stationary modelling in the extreme, but also a competitive scaling to large data sets. Using Lebesgue’s decomposition theorem, it is showed that the two major methodologies in spectral kernel learning represent the continuous and singular components of the measure and how this can be extended to more general cases using a bi-measure.

### 3.20 Kernels – Past and Future?

*Bernhard Schölkopf (MPI für Intelligente Systeme – Tübingen, DE)*

**License** © Creative Commons BY 3.0 Unported license  
© Bernhard Schölkopf

**Main reference** A. Scibior, C.-J. Simon-Gabriel, I. O. Tolstikhin, B. Schölkopf, “Consistent Kernel Mean Estimation for Functions of Random Variables”, in *Proc. of Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pp. 1732–1740, Neural Information Processing Systems Foundation Inc., 2016.  
**URL** <http://papers.nips.cc/paper/6545-consistent-kernel-mean-estimation-for-functions-of-random-variables>

The talk summarized the development of kernel methods for machine learning, focusing on the main ideas and some future possibilities: introduction of p.d. kernels within the theory of integral equations, their use in potential functions methods, in SVM, the general “kernel trick”, the observation that kernels can be defined on arbitrary sets of objects, the link to GPs, and finally the idea to represent distributions by kernel means, underlying kernel tests such as MMD and kernel independence tests.

Kernel mean representations lend themselves well to the development of kernel methods for probabilistic programming, i.e., methods for lifting functional operations defined for data types to the same functional operations for distributions over these data types.

### 3.21 Kernel Embeddings and Bayesian Quadrature

*Dino Sejdinovic (University of Oxford, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Dino Sejdinovic

**Joint work of** Francois-Xavier Briol, Chris Oates, Mark Girolami, Michael Osborne, Dino Sejdinovic

**Main reference** F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic, “Probabilistic Integration: A Role for Statisticians in Numerical Analysis?”, arXiv:1512.00933v5 [stat.ML], 2016.

**URL** <https://arxiv.org/abs/1512.00933v5>

The talk overviewed kernel embeddings as implicit representations of probability measures, leading to the framework allowing nonparametric hypothesis testing and learning on distributions as inputs. In addition, the theory of kernel embeddings allows an alternative interpretation of Bayesian Quadrature (BQ) which does not require invoking the Gaussian process model which puts prior measures on known integrands. This interpretation leads to a recipe for the method applicable where kernel embeddings are not analytically available, while still matching the convergence rates of BQ.

### 3.22 Kernel Mean Embeddings

*Carl-Johann Simon-Gabriel (MPI für Intelligente Systeme – Tübingen, DE)*

**License** © Creative Commons BY 3.0 Unported license  
© Carl-Johann Simon-Gabriel

**Joint work of** Carl-Johann Simon-Gabriel, Bernhard Schölkopf

**Main reference** C.-J. Simon-Gabriel and B. Schölkopf, “Kernel Distribution Embeddings: Universal Kernels, Characteristic Kernels and Kernel Metrics on Distributions”, arXiv:1604.05251 [stat.ML], 2016.

**URL** <https://arxiv.org/abs/1604.05251>

We started with a brief introduction to KMEs, that motivated the embedding function:

$$m : \mathcal{P} \rightarrow \mathcal{H}$$

$$P \mapsto \int k(\cdot, x) dP(x)$$

This embedding defines a distance  $d_k$  between probability measures, which metrizes the usual weak convergence if and only if  $k$  is continuous and  $m$  is injective. We then showed how to systematically link the following three frequently used concepts: universal, characteristic and strictly positive definite kernels. From these links, we concluded that KMEs can be extended so as to embed not only probability measures, but also generalised measurers, aka. Schwartz-distributions. Furthermore, these extensions can remain injective, if the original embedding is injective. The sets of Schwartz-distributions can be seen as sets of measures and of their (distributional) derivatives. Interestingly, the embedding of the derivative  $P'$  of  $P$  can be easily deduced from the embedding of  $P$ . We hope that these extended embeddings will find applications in numerical methods for differential equation solving.



### 3.23 Random Fourier Features and Beyond

*Bharath Sriperumbudur (Pennsylvania State University – University Park, US)*

**License** © Creative Commons BY 3.0 Unported license  
© Bharath Sriperumbudur

**Joint work of** Zoltán Szabó, Bharath Sriperumbudur

**Main reference** B. K. Sriperumbudur, Z. Szabó, “Optimal rates for random Fourier features”, in Proc. of Advances in Neural Information Processing Systems 28 (NIPS 2015), pp. 1144–1152, Neural Information Processing Systems Foundation Inc., 2015.

**URL** <http://papers.nips.cc/paper/5740-optimal-rates-for-random-fourier-features>

In this talk, I will recall the primal and dual formulations of linear ridge regression and kernel ridge regression as a motivating example to introduce feature approximations in the primal setting. Kernel methods have traditionally focused on the dual setting as it does not require the knowledge of the feature maps and also scales only with the sample size. To improve the scalability of kernel methods, various approximations to the dual problem has been studied in terms of incomplete Cholesky factorization, Nyström methods etc. Recently, a Fourier feature based finite dimensional approximation has been introduced which enables to work with the primal setting. In this talk, I will discuss the quality of approximation of Fourier features and present results on the optimality of approximation rates. Then, I will discuss various generalizations and directions of random feature approximations, some of which include rates of approximation for derivatives of kernels, optimal approximation rates for operator-valued kernels and possibility of other approximations to improve the scalability of kernel methods in the primal setting.

### 3.24 Learning with Hierarchical Kernels

*Ingo Steinwart (Universität Stuttgart, DE)*

**License** © Creative Commons BY 3.0 Unported license  
© Ingo Steinwart

**Main reference** I. Steinwart, P. Thomann, N. Schmid, “Learning with Hierarchical Gaussian Kernels”, arXiv:1612.00824 [stat.ML], 2016.

**URL** <http://arxiv.org/abs/1612.00824>

We investigate iterated compositions of weighted sums of Gaussian kernels and provide an interpretation of the construction that shows some similarities with the architectures of deep neural networks. On the theoretical side, we show that these kernels are universal and that SVMs using these kernels are universally consistent. We further describe a parameter optimization method for the kernel parameters and empirically compare this method to SVMs, random forests, a multiple kernel learning approach, and to some deep neural networks.

### 3.25 Distribution Regression

Zoltán Szabó (*Ecole Polytechnique – Palaiseau, FR*)

**License** © Creative Commons BY 3.0 Unported license  
© Zoltán Szabó

**Joint work of** Zoltán Szabó, Bharath Sriperumbudur, Barnabás Póczos, Arthur Gretton

**Main reference** Z. Szabó, B. K. Sriperumbudur, B. Póczos, A. Gretton, “Learning Theory for Distribution Regression”, *Journal of Machine Learning Research* 17(152):1–40, 2016.

**URL** <http://jmlr.org/papers/v17/14-510.html>

We focus on the distribution regression problem (DRP): we regress from probability measures to Hilbert-space valued outputs, where the input distributions are only available through samples (this is the ‘two-stage sampled’ setting). Several important statistical and machine learning problems can be phrased within this framework including point estimation tasks without analytical solution (such as entropy estimation), or multi-instance learning. However, due to the two-stage sampled nature of the problem, the theoretical analysis becomes quite challenging: to the best of our knowledge the only existing method with performance guarantees to solve the DRP task requires density estimation (which often performs poorly in practise) and the distributions to be defined on a compact Euclidean domain. We present a simple, analytically tractable alternative to solve the DRP task: we embed the distributions to a reproducing kernel Hilbert space and perform ridge regression from the embedded distributions to the outputs. We prove that this scheme is consistent under mild conditions, and construct explicit finite sample bounds on its excess risk as a function of the sample numbers and the problem difficulty, which hold with high probability. Specifically, we establish the consistency of set kernels in regression, which was a 17-year- old-open question, and also present new kernels on embedded distributions. The practical efficiency of the studied technique is illustrated in aerosol prediction using multispectral satellite images.

### 3.26 Stochastic (partial) differential equations and Gaussian processes

Simo Särkkä (*Aalto University, FI*)

**License** © Creative Commons BY 3.0 Unported license  
© Simo Särkkä

Stochastic partial differential equations and stochastic differential equations can be seen as alternatives to kernels in representation of Gaussian processes. Linear operator equations give spatial kernels, temporal kernels are equivalent to linear Itô stochastic differential equations. The differential equation representations allow for the use of differential equation numerical methods on Gaussian processes. For example, finite-differences, finite elements, basis function methods, and Galerkin methods can be used. In temporal and spatio-temporal case we can use linear-time Kalman filter and smoother approaches.

### 3.27 Consistent Kernel Mean Estimation for Functions of Random Variables

*Ilya Tolstikhin (MPI für Intelligente Systeme – Tübingen, DE)*

**License** © Creative Commons BY 3.0 Unported license  
© Ilya Tolstikhin

**Joint work of** Adam Scibior, Carl-Johann Simon-Gabriel, Ilya Tolstikhin, Bernhard Schölkopf  
**Main reference** A. Scibior, C.-J. Simon-Gabriel, I. O. Tolstikhin, B. Schölkopf, “Consistent Kernel Mean Estimation for Functions of Random Variables”, in Proc. of Advances in Neural Information Processing Systems 29 (NIPS 2016), pp. 1732–1740, Neural Information Processing Systems Foundation Inc., 2016.  
**URL** <http://papers.nips.cc/paper/6545-consistent-kernel-mean-estimation-for-functions-of-random-variables>

Given a random variable  $X$  and a function defined over the same space, we consider a problem of constructing a flexible representation for a distribution of  $f(X)$ . Following the approach of [1], we propose to do so by using mean embeddings of probability distributions into corresponding Reproducing Kernel Hilbert Spaces. Our new results show that a consistent estimation of the mean embedding of  $X$  leads to a consistent estimation of the mean embedding of  $f(X)$ . In particular, this result shows the consistency of a new estimator proposed by [1]. Apart from asymptotic results we also provide finite sample guarantees for Matern kernels and discuss possible applications, including probabilistic programming.

#### References

- 1 B. Schölkopf, K. Muandet, K. Fukumizu, S. Harmeling, and J. Peters: “Computing functions of random variables via reproducing kernel Hilbert space representations.” In: Statistics and Computing 25(4), 755–766, 2015.

### 3.28 Uncertain inputs in Gaussian Processes

*Mark van der Wilk (University of Cambridge, GB)*

**License** © Creative Commons BY 3.0 Unported license  
© Mark van der Wilk

We introduced Sparse GP inference using Variational Bayesian inference. In this framework, we aim to minimise the KL divergence of some easily computable approximate posterior to the true posterior. The structure of the approximations lends itself really well to handling uncertain inputs as well. The ability to do so is essential for the idea of making GPs “building blocks” of Machine Learning, like neural network layers, if uncertainty is to be taken into account. Finally we contrast the goal of uncertain input GPs to distribution regression from the kernel literature. Could there be any way to combine the methods?

### 3.29 Frequentist properties of GP learning methods

*Harry van Zanten (University of Amsterdam, NL)*

**License** © Creative Commons BY 3.0 Unported license  
© Harry van Zanten

GPs have been routinely used as priors in a variety of nonparametric statistical problems, including regression and classification. In this talk I gave a short overview of theoretical

results that have been obtained during the last 10 years or so about the frequentist performance of such methods.

The first wave of convergence rate results for nonparametric Bayes with fixed GP priors showed that to achieve optimal rates, the regularity of the GP has to exactly match the regularity of the function that is being estimated. These results apply to all the popular kernels, including the Matern and the squared exponential, for instance. Since the true regularity of the function of interest is typically not known, it is therefore necessary to have adaptive procedures that automatically set tuning, or hyper parameters in an optimal way. A second wave of results showed that both hierarchical and empirical Bayes methods can do this properly, provided they are carefully constructed. The results show that matters are actually a bit delicate. It is important for instance which priors are placed on hyper parameters, or which hyper parameters are held fixed and which are tuned. The third class of results that I discussed deal with the frequentist interpretation of nonparametric credible sets. Ideally, we would like a 95% credible set to be a frequentist 95% confidence set as well and at the same time have minimal, optimal size. It turns out that in the adaptive setting in which you don't know the regularity of the truth and use for instance hierarchical Bayes or empirical Bayes, this is fundamentally impossible. Whatever priors you use, there are always ground truths for which the credible sets are completely misleading. This means that confidence statements in nonparametric settings are fundamentally conditional: in nonparametric problems you can only believe credible sets, or error bars, if you *really* believe that your prior reflects the fine properties of the truth.

Several fundamental issues concerning GP methods are not fully understood yet. One of the most interesting ones is perhaps the issue of the fundamental limitations and possibilities of distributed GP methods. Under which conditions can such methods achieve the same optimal, adaptive performance as a centralised methods?

## References

- 1 A. W. van der Vaart, and J. H. van Zanten: “Rates of contraction of posterior distributions based on Gaussian process priors.” In: *The Annals of Statistics*, 1435–1463, 2008.
- 2 A. W. van der Vaart, and J. H. van Zanten: “Adaptive Bayesian estimation using a Gaussian random field with inverse Gamma bandwidth.” In: *The Annals of Statistics*, 2655–2675, 2009.
- 3 A. W. van der Vaart, and J. H. van Zanten: “Information rates of nonparametric Gaussian process methods.” In: *Journal of Machine Learning Research* 12, 2095–2119, 2011.
- 4 B. Szabó, A. W. van der Vaart, and J. H. van Zanten: “Frequentist coverage of adaptive nonparametric Bayesian credible sets.” In: *The Annals of Statistics* 43(4), 1391–1428, 2015.

## 3.30 The ML Invasion of ABC

*Richard Wilkinson (University of Sheffield, GB)*

License © Creative Commons BY 3.0 Unported license  
© Richard Wilkinson

There are classes of models for which we can easily sample realizations from the model, but where we cannot compute the likelihood function  $\pi(x|\theta)$ . These typically are in scientific problems where the model represents our physical knowledge about the system. ABC methods are a class of Monte Carlo algorithms for doing all inference using simulation from the model.


In particular, they target some form of posterior distribution  $\tilde{\pi}(\theta|D) \propto \tilde{\pi}(D|\theta)\pi(\theta)$  where the likelihood function used,  $\tilde{\pi}$ , may be very different from the true likelihood  $\pi$ .

In the past few years, some of the most interesting ideas in ABC have arisen in machine learning groups. These include approaches for bypassing the need to choose a set of summary statistics by instead using a kernel embedding & using the MMD metric to determine whether the simulator output is comparable to the data, & the use of generative adversarial networks as alternatives to pseudo-likelihood based approaches. There has also been significant work using surrogate models for the likelihood function, for example, by using Gaussian processes to approximate the likelihood function & then using the GP in a MCMC inference scheme to find the posterior.

## 4 Working groups

### 4.1 Generative Models

*David Duvenaud (Toronto, CA)*

License  Creative Commons BY 3.0 Unported license  
© David Duvenaud

Sebastian Nowozin introduced variational autoencoders, amortized inference, and the reparameterization trick. He said that kernel MMD is a good way to introduce non-probabilistic practitioners to the probabilistic way of doing things, because all you need to do is simply add noise to the input of whatever generative procedure and a diversity term on pairs of outputs to the loss. He says it's a lot easier to implement and get to work than GAN training.

Arthur, Ilya and Bernhard had a long discussion about the relationships between kernel MMD, VAEs and GANs. Ilya suggests that there is an opportunity for theoreticians to clean up and organize all the tricks that are required for training GANs.

Sebastian makes the point that modeling natural images is mainly interesting of a proxy task for modeling high-dimensional densities, and it's a task where humans can evaluate the quality of samples.

We talked about how the blurry images of VAEs can be address by using more sophisticated likelihoods, as in <https://arxiv.org/pdf/1611.05013v1.pdf>.

Krikamol asks if we can incorporate MCMC into VAEs or GANs, and I mention that Max Welling has a paper that does this: <https://arxiv.org/abs/1410.6460>.

We also talked about a paper by Roger Grosse that attempts to properly evaluate the predictive probability of VAEs and GANs: <https://openreview.net/pdf?id=B1M8JF9xx>.

I also suggest there might be room for kernel people to help with the new “Operator Variational Inference” method: <https://arxiv.org/pdf/1610.09033.pdf>.

Then we broke for coffee.

## 4.2 Limitations of GPs / non-Gaussian-Processes

Stefan Harmeling (Universität Düsseldorf, DE)

License  Creative Commons BY 3.0 Unported license  
© Stefan Harmeling

**Participants:** Carl Rasmussen, Hannes Nickisch, Manfred Opper, Marc Deisenroth, Maren Mahsereci, Mark van der Wilk, Michael Osborne, Philipp Hennig, Roman Garnett, Simo Särkkä, Stefan Harmeling, Stephen Roberts.

1. *Question:* Are stacked or deep GPs useful extensions of GPs?  
*Answer:* Yes, stacked or deep GPs are working and they are useful, since they bring new possibilities a GP has not. Their deep structure might also lead to fewer parameters. However, learning the parameters of stacked or deep GPs might require lots of data.
2. *Question:* How can we formulate non-GPs on functions spaces?  
*Answer:* (i) The concatenation of two function-valued random variables is usually not a GP distribution. Or even simpler, (ii) a function-valued random variable can be concatenated with a nonlinear function to obtain a function-valued random variable that follows no longer a GP distribution. With such tricks we can impose constraints on function distributions such as monotonicity, convexity, non-negativity, etc. However, for lots of data the posterior GP might fulfill the constraints with high probability. We can also apply nonlinear functionals to the GPs (as Hilbert space elements), e.g., we can use GPs as inputs to nonlinear ordinary or partial differential equations to create non-Gaussian process outputs. The scaled mixtures of probability measures is one way which leads to e.g. Student-t and related processes. Using stochastic processes as inputs to other processes (as in deep GPs) also leads to non-Gaussian processes.
3. *Question:* In practice a large number of hyper-parameters make inference harder. Can we tie parameters to the rescue?  
*Answer:* It might help as is suggested by deep learning, where even randomly tying parameters can improve performance. Another difficulty of inference might be a situation, in which the input data lies on a low-dimensional manifold. This could lead to uncertainty off the manifold which might be sometimes problematic. Deep learning suggests the intuition that increasing the number of parameters can actually make optimisation of those parameters *easier*. While this might hold for Gaussian process models with lots of hyperparameters, we don't have much assurance that this wouldn't result in overfitting.
4. *Question:* What is a simple example of a family of functions that a GP can not model properly?  
*Answer:* The set of step functions (i.e. with exactly one step at an unknown location) can not be the support of a GP distribution in function space. So using covariance functions (kernels) to specify distributions in function space is limited. It is either that a GP spreads its mass too thinly, or spreads it too wide.


One other point: Perhaps certain *prior* distributions are hard to model using Gaussian processes. However, as data comes in, the posterior of a non-Gaussian process might be easy to model using a GP. This has the flavour of variational inference and has more desirable properties than relying on the data to constrain a GP prior.

### Summary

1. Stacked and deep GPs are useful!
2. Create non-GPs e.g. by concatenation.
3. Lots of parameters make inference harder.
4. There is no GP for step functions.

## 4.3 The separation between Kernels/GPs and Deep Learning

*Sebastian Nowozin (Microsoft Research UK – Cambridge, GB)*

License  Creative Commons BY 3.0 Unported license  
© Sebastian Nowozin

Motivation: what are the known success cases and failure modes of kernels and deep learning methods? Which applications are they best suited for?

**Practical Observations.** Bayesian probabilistic models provide a clean framework for thinking about applications such as one-shot learning and transfer learning; increasingly deep learning methods attempt to achieve the same applications where uncertainty is important, often successfully.

Gaussian processes are well established for regression problems, in particular in the small data setting. For example, in sample-efficient reinforcement learning. Similarly, in spatio-temporal modeling they work well.

Building kernels for high-dimensional input data ( $d > 10$  or  $d > 100$ ) or heterogeneous data is difficult. For Gaussian processes, performing hyperparameter optimization is challenging when there are a large number of hyperparameters. Perhaps stochastic optimization methods or variational inference for the hyperparameters can improve this.

Deep learning offer a flexible framework of overparameterized functions. They place representation learning first in terms of an explicit feature map, which allows representation to be useful for different applications.

Kernel methods still dominate in testing, such as with maximum mean discrepancy (MMD), because guarantees are important in hypothesis testing. Kernel methods are also popular in structured input settings, where we handle strings or graph structures, for example in bioinformatics.

In Bayesian optimization Gaussian processes are successful but other probabilistic models are possible. Bayesian neural networks may be an alternative but there are computational issues as well.

The deep learning community also spends a good amount of engineering and efficient implementation; does the GP community spend the same amount of effort?

Kernel methods provide mathematical tools to potentially prove guarantees within control applications.

Kernel methods are also *easy* to automate and predictable both in runtime and in the influence of parameters.

**Computational Issues.** Deep learning methods are perhaps also really successful because they are scalable; therefore, computational issues may be an important aspect of practical success.

Fast GPs scale with a flexible parameters, the number of inducing points. The variational approach is one approach but there are perhaps more efficient approaches. Yet, they remain much less scalable than deep learning methods.

Kernel methods (e.g. SVM) have also achieved better scalability and now can be used for up to 10M data points with controlled and guaranteed approximation guarantees. Popular packages such as scikit-learn also scales to 500k points without practical difficulties.

Representation power of functions: deep learning can represent complicated functions, sparse GPs with inducing points cannot. Perhaps there are interesting extensions of the inducing point approach to enhance the representational power of GPs.

Prediction: neural network prediction is fast, but for GPs prediction is expensive.

Dataset size: small data is good for kernel methods, big data is good for deep learning.

**Theoretical Limitations.** Standard Gaussian processes have some known limitations.

What guarantees can be provided for the error bars of a GP? They correspond to posterior credible intervals, but therefore are conditioned on the assumed model class. Non-parametric uncertainty quantification is not possible in general (results of Richard Nickl). Do we care about the prediction accuracy or about the calibration properties of the error bars? Are error bars sufficient in practice, even if we know the model assumptions to be wrong? From the Bayesian viewpoint error bars depend only on the modeling assumptions, so we need to question modeling assumptions if we are not satisfied with the quality of the error bars.

Why does deep learning work in very high dimensions? Are there fundamental assumptions about real world densities that we have not understood yet?

Classification is still difficult with Gaussian process models.

There are two additional observations regarding kernel methods versus deep learning:

1. Degree-of-freedom bottleneck: a GP has effective  $N$  parameters to determine a function, where  $N$  is the number of samples. A deep network has a potentially larger number.
2. Kernel-information-bottleneck: a kernel consists of  $N \times N$  scalars. If every instance contains a large amount of information (e.g. a megapixel image), more information flows to a deep neural network system than to a kernel method.

**Deep Kernel Methods versus Deep Learning.** Kernel methods typically use a handcrafted kernel, whereas deep learning methods learn the representation by data.

Practical advantages of non-parametric methods are most likely not existing; the advantage is in theory, being able to prove how to increase function class as the data grows in order to guarantee the right function is recovered.

Deep learning methods can also represent uncertainty either by directly fitting a variance or by using Bayesian neural networks.


Main difficulties in deep kernel methods is in scalability. Main disadvantage compared to deep neural networks is the difficulty of running them on GPU, for example for computing covariance matrices as part of performing a deep kernel computation, or for computing the posterior variance for a given data point. Do deep GPs scale to many layers? This is unclear.

Optimization problems arising in deep neural networks and in deep learning are the same, for example initialization and optimization.



## 4.4 Connections and Differences between Kernels and GPs

Dino Sejdinovic (University of Oxford, GB)

License  Creative Commons BY 3.0 Unported license  
© Dino Sejdinovic

The discussion recognized that there are shared mathematical foundations of the frequentist kernel methods and Gaussian Processes (GP). These foundations are based on the theory of Gaussian Hilbert spaces and the fact that the notions of *orthogonality and independence coincide* on the  $L_2$ -spaces of gaussian random variables. There is a need for a dictionary translating different concepts within these two communities using this shared mathematical framework – there is potentially lots to be gained from these different interpretations. As an example, we discussed the frequentist interpretation of the standard GP posterior covariance. It turns out this can be viewed as an inner product between the component of features orthogonal to the data subspace, i.e. *similarity not explained by the data*. There is also a “worst-case error” over the reproducing kernel Hilbert space (a specified class of functions with the encoded regularity) interpretation, which can be used to quantify uncertainty. Similar connection exists between maximum mean discrepancy (MMD) and the GP posterior variance in Bayesian quadrature.

Main similarities arise in standard supervised learning settings (e.g. kernel ridge regression and GP regression are closely related) but the connections in unsupervised settings are less well understood. For example, are there Bayesian counterparts to kernel PCA or density estimation using infinite dimensional exponential families? How are they related?

Another point of discussion revolved around the result that *the samples from GP almost surely do not lie in the RKHS* with the corresponding covariance kernel, even though the posterior mean does. This has important implications on model specification within the two frameworks. What are useful ways to think about this?

It was also reiterated that the two frameworks have different philosophies, with the frequentist focus on risk and the GP framework focusing on describing posterior measures and being oblivious to the task that follows it. Thus, there are important differences in the decision making process and since the two frameworks generally do different things with the same mathematical objects and interpret them differently – a caution should be exercised when translating these mathematical objects and this process may in some cases be misleading.

## Participants

- Florence d'Alché-Buc  
Telecom ParisTech, FR
- Marc Deisenroth  
Imperial College London, GB
- David Duvenaud  
Toronto, CA
- Roman Garnett  
Washington University –  
St. Louis, US
- Arthur Gretton  
University College London, GB
- Stefan Harmeling  
Universität Düsseldorf, DE
- Philipp Hennig  
MPI für Intelligente Systeme –  
Tübingen, DE
- James Hensman  
Lancaster University, GB
- José Miguel  
Hernández-Lobato  
Harvard University –  
Cambridge, US
- Frank Hutter  
Universität Freiburg, DE
- Motonobu Kanagawa  
Institute of Statistical  
Mathematics – Tokyo, JP
- Andreas Krause  
ETH Zürich, CH
- Neil D. Lawrence  
University of Sheffield, GB
- David Lopez-Paz  
Facebook – AI Research, US
- Maren Mahsereci  
MPI für Intelligente Systeme –  
Tübingen, DE
- Julien Mairal  
INRIA – Grenoble, FR
- Krikamol Muandet  
Mahidol University, TH
- Hannes Nickisch  
Philips – Hamburg, DE
- Sebastian Nowozin  
Microsoft Research UK –  
Cambridge, GB
- Cheng Soon Ong  
Data61 – Canberra, AU
- Manfred Opper  
TU Berlin, DE
- Peter Orbanz  
Columbia University –  
New York, US
- Michael A. Osborne  
University of Oxford, GB
- Carl Edward Rasmussen  
University of Cambridge, GB
- Stephen Roberts  
University of Oxford, GB
- Volker Roth  
Universität Basel, CH
- Simo Särkkä  
Aalto University, FI
- Bernt Schiele  
MPI für Informatik –  
Saarbrücken, DE
- Michael Schober  
MPI für Intelligente Systeme –  
Tübingen, DE
- Bernhard Schölkopf  
MPI für Intelligente Systeme –  
Tübingen, DE
- Dino Sejdinovic  
University of Oxford, GB
- Carl-Johann Simon-Gabriel  
MPI für Intelligente Systeme –  
Tübingen, DE
- Bharath Sriperumbudur  
Pennsylvania State University –  
University Park, US
- Ingo Steinwart  
Universität Stuttgart, DE
- Zoltán Szabó  
Ecole Polytechnique –  
Palaiseau, FR
- Ilya Tolstikhin  
MPI für Intelligente Systeme –  
Tübingen, DE
- Raquel Urtasun  
University of Toronto, CA
- Mark van der Wilk  
University of Cambridge, GB
- Harry van Zanten  
University of Amsterdam, NL
- Richard Wilkinson  
University of Sheffield, GB

