Chapter 1

Conclusions and Outlook

This chapter summarizes the contributions of this thesis, articulates some of the questions raised by this work, and relates the kernel-based model-building procedure of sections 1.4 to 1.4 to the larger project of automating statistics and machine learning.

1.1 Summary of contributions

The main contribution of this thesis was to develop a way to automate the construction of structured, interpretable nonparametric regression models using Gaussian processes. This was done in several parts: First, section 1.4 presented a systematic overview of kernel construction techniques, and examined the resulting GP priors. Next, section 1.4 showed the viability of a breadth-first search over an open-ended space of kernels, and showed that the corresponding GP models could be automatically decomposed into diverse parts illustrating the structure found in the data. Section 1.4 showed that sometimes parts of kernels can be described modularly, allowing automatically written text to be included in detailed reports describing GP models. An example report is included in appendix ??. Together, these chapters demonstrate a proof-of-concept of what could be called an "automatic statistician" capable of the performing some of the model construction and analysis currently performed by experts.

The second half of this thesis examined several extensions of Gaussian processes, all of which enabled the automatic determination of model choices that were previously set by trial and error or cross-validation. Section 1.4 characterized and visualized deep Gaussian processes, related them to existing deep neural networks, and derived novel deep kernels. Section 1.4 investigated additive GPs, and showed that they have the same covariance as a GP using dropout. Section 1.4 extended the GP latent variable model into a Bayesian clustering model which automatically infers the nonparametric shape of each cluster, as well as the number of clusters.

1.2 Structured versus unstructured models

One question left unanswered by this thesis is: when to prefer the structured, kernel-based models described in sections 1.4 to 1.4 to the relatively unstructured deep GP models described in section 1.4? This section considers some advantages and disadvantages of the two approaches.

- **Difficulty of optimization.** The discrete nature of the space of composite kernel structures can be seen as both a blessing and a curse. Certainly, a mixed discrete-and-continuous search space requires more complex optimization procedures than the continuous-only optimization possible in deep GPs.

  However, the discrete nature of the space of composite kernels offers the possibility of learning heuristics to suggest which types of structure to add, based on features of the dataset, or previous model fits. For example, finding periodic structure or growing variance in the residuals suggests adding periodic or linear components to the kernel, respectively. It is not clear whether such heuristics could easily be constructed for optimizing the variational parameters of a deep GP.

- **Long-range extrapolation.** Another open question is whether the inductive bias of deep GPs can be made to allow the sorts of long-range extrapolation shown in sections 1.4 and 1.4. As an example, consider the problem of extrapolating a periodic function. A deep GP could learn a latent representation similar to that of the periodic kernel, projecting into a basis equivalent to $[\sin(x), \cos(x)]$ in the first hidden layer. However, to extrapolate a periodic function, the sin and cos functions would have to repeat beyond the input range of the training data, which would not happen if each layer assumed only local smoothness.

  One obvious possibility is to marry the two approaches, building deep GPs with structured kernels. However, we may lose some of the advantages of interpretability by this approach, and inference would become more difficult.

  Another point to consider is that, in high dimensions, the distinction between interpolation and extrapolation becomes less meaningful. If the training and test data both live on a low-dimensional manifold, then learning a suitable representation of that manifold may be sufficient for obtaining high predictive accuracy.

- **Ease of interpretation.** Historically, the statistics community has put more emphasis on the interpretabilty and meaning of models than the machine learning community, which has focused more on predictive performance. To begin to automate the practice of statistics, developing model-description procedures for powerful open-ended model classes seems to be a necessary step.

  At first glance, automatic model description may seem to require a decomposition of the model being described into discrete components, as in the additive decomposition demonstrated in sections 1.4 to 1.4.

  On the other hand, Damianou and Lawrence (2013) showed that deep GPs allow summarization of high-dimensional structure through sampling from the posterior, examining the dimension of each latent layer, visualizing latent coordinates, and examining how the predictive distribution changes as one moves in the latent space. Perhaps more sophisticated procedures could also allow intelligible text-based descriptions of such models.

The warped mixture model of section 1.4 represents a compromise between these two approaches, combining a discrete clustering model with an unstructured warping function. However, the explicit clusterting model may by unnecessary: the results of Damianou and Lawrence (2013) suggest that clustering can be automatically (but implicitly) accomplished by a sufficiently deep, unstructured GP.

1.3 Approaches to automating model construction

This thesis is a small part of a larger push to automate the practice of model building and inference. Broadly speaking, this problem is being attacked from two directions.

From the top-down, the probabilistic programming community is developing automatic inference engines for extremely broad classes of models (Goodman et al., 2008; Koller et al., 1997; Mansinghka et al., 2014; Milch et al., 2007; Stan Development Team, 2014; Wood et al., 2014) such as the class of all computable distributions (Li and Vitányi, 1997; Solomonoff, 1964). As discussed in **??**, model construction procedures can usually be seen as a search through an open-ended model class. Exhaustive search strategies have been constructed for the space of computable distributions (Hutter, 2002; Levin, 1973; Schmidhuber, 2002), but they remain impractically slow.

An alternative, bottom-up, approach is to design procedures which extend and combine existing model classes for which relatively efficient inference algorithms are already known. The language of models proposed in section 1.4 is an example of this bottom-up approach. Another example is Grosse (2014), who built an open-ended language of matrix decomposition models and a corresponding compositional language of relatively efficient approximate inference algorithms. Similarily, Steinruecken (2014) showed how to compose inference algorithms for sequence models. These approaches have the advantage that inference is usually feasible for any model in the language, but extending the language may require developing new inference algorithms.

If sufficiently powerful building-blocks are composed, the line between the top-down and bottom-up approaches becomes blurred. For example, deep generative models (Adams et al., 2010; Bengio et al., 2013; Damianou and Lawrence, 2013; Rippel and Adams, 2013; Salakhutdinov and Hinton, 2009) could be considered an example of the bottom-up approach, since they compose individual model "layers" to produce more powerful models. However, large neural nets can capture enough different types of structure that they could also be seen as an example of the universalist, top-down approach.

1.4 End note

It seems clear that one way or another, large parts of the existing practice of model-building will eventually be automated. However, it remains to be seen which of the above model-building approaches will be most useful. I hope that this thesis will contribute to our understanding of the strengths and weaknesses of these different approaches, and towards the use of more powerful model classes by practitioners in other fields.

References

Ryan P. Adams, Hanna M. Wallach, and Zoubin Ghahramani. Learning the structure of deep sparse graphical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010. (page 4)

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013. (page 4)

Andreas Damianou and Neil D. Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013. (pages 3 and 4)

Noah D. Goodman, Vikash K. Mansinghka, Daniel M. Roy, K. Bonawitz, and Joshua B. Tenenbaum. Church: A language for generative models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 220–229, 2008. (page 3)

Roger B. Grosse. *Model Selection in Compositional Spaces*. PhD thesis, Massachusetts Institute of Technology, 2014. (page 4)

Marcus Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(03):431–443, 2002. (page 3)

Daphne Koller, David McAllester, and Avi Pfeffer. Effective Bayesian inference for stochastic programs. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 1997. (page 3)

Leonid A. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973. (page 3)

Ming Li and Paul M.B. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 1997. (page 3)

Percy Liang, Michael I. Jordan, and Dan Klein. Learning programs: A hierarchical Bayesian approach. In *Proceedings of the 27th International Conference on Machine Learning*, pages 639–646, 2010.

Vikash Mansinghka, Daniel Selsam, and Yura Perov. Venture: a higher-order probabilistic programming platform with programmable inference. *arXiv preprint arXiv:1404.0099*, 2014. (page 3)

Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: Probabilistic models with unknown objects. *Statistical relational learning*, page 373, 2007. (page 3)

Oren Rippel and Ryan P. Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013. (page 4)

Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009. (page 4)

Jürgen Schmidhuber. The speed prior: a new simplicity measure yielding near-optimal computable predictions. In *Computational Learning Theory*, pages 216–228. Springer, 2002. (page 3)

Ray J. Solomonoff. A formal theory of inductive inference. Part I. *Information and control*, 7(1):1–22, 1964. (page 3)

Stan Development Team. Stan: A c++ library for probability and sampling, version 2.2, 2014. URL `http://mc-stan.org/`. (page 3)

Christian Steinruecken. *Lossless Data Compression*. PhD thesis, Cavendish Laboratory, University of Cambridge, 2014. (page 4)

Frank Wood, Jan Willem van de Meent, and Vikash Mansinghka. A new approach to probabilistic programming inference. In *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, 2014. (page 3)