Probabilistic Machine Learning
Lecture 10
Gaussian Process Regression: An Extensive Example

Philipp Hennig

25 May 2023

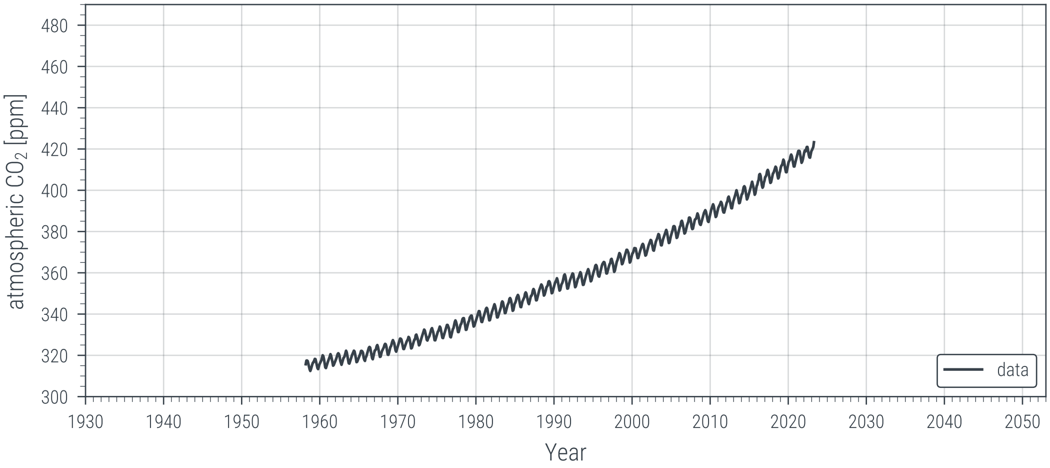
Eberhard Karls Universität Tübingen

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

A Dataset

NOAA Mauna Loa CO_2 data

adapted from Rasmussen & Williams, 2007



Additive Kernels and Multi-Output GPs

a sum in the kernel implies the data is emerges from a sum of functions

- Our data represents an (approximately linear) combination of multiple functions:

$$
\begin{aligned}
p(f_1) &= \mathcal{GP}(f_1; 0, k_1) \\
p(f_2) &= \mathcal{GP}(f_2; 0, k_2) \\
p(f_1, f_2) &= \mathcal{GP}\left(\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}\right) \\
p(f) &= \mathcal{GP}\left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}; \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) \\
&= \mathcal{GP}(f; 0, k_1 + k_2)
\end{aligned}
$$

Can we **learn** the kernel?

Hierarchical Bayesian Inference

Bayesian model adaptation

$$p(f \mid \mathbf{y}, x, \boldsymbol{\theta}) = \frac{p(\mathbf{y} \mid f, x, \boldsymbol{\theta})p(f \mid, \boldsymbol{\theta})}{\int p(\mathbf{y} \mid f, x, \boldsymbol{\theta})p(f \mid, \boldsymbol{\theta})\, df} = \frac{p(\mathbf{y} \mid f, x, \boldsymbol{\theta})p(f \mid, \boldsymbol{\theta})}{p(\mathbf{y} \mid x, \boldsymbol{\theta})}$$

- **Model parameters** like θ are also known as **hyper-parameters**.
- This is largely a computational, practical distinction:

data	are observed	→ condition
variables	are the things we care about	→ full probabilistic treatment
parameters	are the things we have to deal with to get the model right	→ integrate out
hyper-parameters	are the top-level, too expensive to properly infer	→ fit

The **model evidence** in Bayes' Theorem is the (marginal) **likelihood** for the model. So we would like

$$p(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y} \mid \boldsymbol{\theta}')p(\boldsymbol{\theta}')\, d\boldsymbol{\theta}'}$$

Can We **Learn** the Kernel?

Bayesian Hierarchical Inference

$$p(f \mid \theta) = \mathcal{GP}(f; m_\theta, k_\theta) \qquad \text{e.g.} \quad m_\theta(\bullet) = \phi(\bullet)^\intercal \boldsymbol{\theta}, \text{ or } k_\theta(\bullet, \circ) = \theta_1 \exp\left(-\frac{(\bullet - \circ)^2}{2\theta_2^2}\right).$$

- The **evidence** in Bayes' theorem is the **marginal likelihood for the model**

$$p(f \mid \mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{y} \mid f, \mathbf{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{\int p(\mathbf{y} \mid f, \mathbf{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})\, df} = \frac{p(\mathbf{y} \mid f, \mathbf{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})}$$

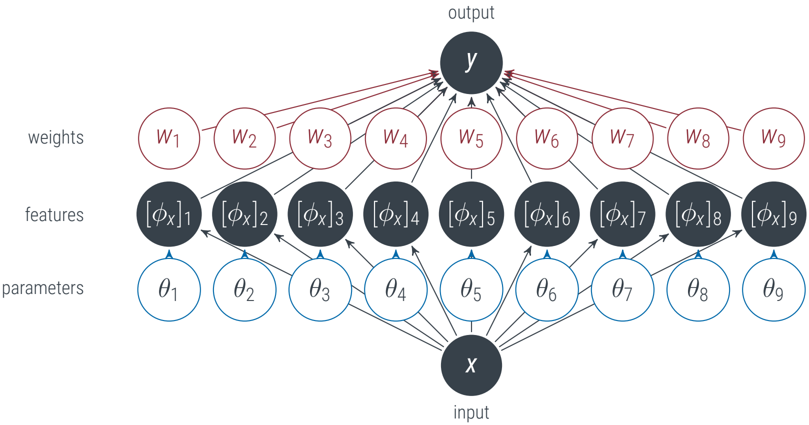
- For Gaussians *and Gaussian processes*, die evidence has **analytic form**:

$$\underbrace{\mathcal{N}(\mathbf{y}; \phi_X^{\boldsymbol{\theta}\,\intercal} w + \mathbf{b}, \Lambda)}_{p(\mathbf{y}|f,\mathbf{x},\boldsymbol{\theta})} \cdot \underbrace{\mathcal{N}(w, \mu, \Sigma)}_{p(f)} = \underbrace{\mathcal{N}(w; m_{\text{post}}^{\boldsymbol{\theta}}, V_{\text{post}}^{\boldsymbol{\theta}})}_{p(f|\mathbf{y},\mathbf{x},\boldsymbol{\theta})} \cdot \underbrace{\mathcal{N}(\mathbf{y}; \phi_X^{\boldsymbol{\theta}\,\intercal} \mu + b, \phi_X^{\boldsymbol{\theta}\,\intercal} \Sigma \phi_X^{\boldsymbol{\theta}} + \Lambda)}_{p(\mathbf{y}|\boldsymbol{\theta},\mathbf{x})}$$

$$\mathcal{N}(\mathbf{y}; f^{\boldsymbol{\theta}}(X), \Lambda^{\boldsymbol{\theta}}) \cdot \mathcal{GP}(f, \mu^{\boldsymbol{\theta}}, k^{\boldsymbol{\theta}}) = \mathcal{GP}(f; m_{\text{post}}^{\boldsymbol{\theta}}, V_{\text{post}}^{\boldsymbol{\theta}}) \cdot \mathcal{N}(\mathbf{y}; \mu^{\boldsymbol{\theta}}(X), \Lambda^{\boldsymbol{\theta}} + k^{\boldsymbol{\theta}}(X, X))$$

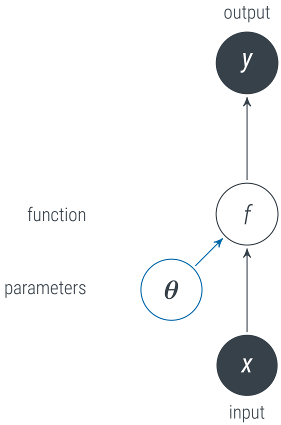
Kernel Learning is Fitting an Entire Population of Features

The Graphical Model View



Kernel Learning is Fitting an Entire Population of Features

The Graphical Model View



ML / MAP in Practice

Finding the "best fit" θ in Gaussian models

[e.g. DJC MacKay, *The evidence framework applied to classification networks*, 1992]

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \int \mathcal{N}(\boldsymbol{y}; f(X), \Lambda_{\boldsymbol{\theta}}) \cdot \mathcal{N}(f_X, \mu_X, k_{XX})\, df_X \\
&= \arg\max_{\boldsymbol{\theta}} \mathcal{N}(\boldsymbol{y}; \mu_X; k_{XX} + \Lambda_{\boldsymbol{\theta}}) \int \mathcal{N}(f_X; \mu_{y,X}, v_{y,XX})\, df_X \\
&= \arg\max_{\boldsymbol{\theta}} \mathcal{N}(\boldsymbol{y}; \quad \mu_X^{\theta}, \quad k_{XX}^{\theta} + \Lambda^{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log \mathcal{N}(\boldsymbol{y}; \quad \mu_X^{\theta}, \quad k_{XX}^{\theta} + \Lambda^{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} - \log \mathcal{N}(\boldsymbol{y}; \quad \mu_X^{\theta}, \quad k_{XX}^{\theta} + \Lambda^{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \left(\underbrace{(\boldsymbol{y} - \mu_X^{\theta})^\intercal \left(k_{XX}^{\theta} + \Lambda^{\theta}\right)^{-1} (\boldsymbol{y} - \phi_X^{\boldsymbol{\theta}\intercal} \mu)}_{\text{square error}} + \underbrace{\log \left| k_{XX}^{\theta} + \Lambda^{\theta} \right|}_{\text{model complexity / \textbf{Occam factor}}} \right) + \frac{N}{2} \log 2\pi
\end{aligned}
$$

Numquam ponenda est pluralitas sine necessitate.
Plurality must never be posited without necessity.

William of Occam
(1285 (Occam, Surrey)–1349 (Munich, Bavaria))
stained-glass window by Lawrence Lee

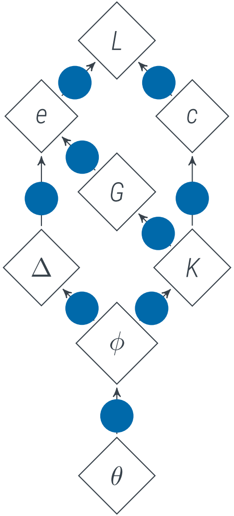
What does the Optimizer need from us?

A bit of algorithmic wizardry

$$L(\theta) = \frac{1}{2}\Bigg(\underbrace{(y - \phi_X^{\theta\top}\mu)^\top \underbrace{\Big(\underbrace{\phi_X^{\theta\top}\Sigma\phi_X^{\theta}}_{=:K} + \Lambda\Big)^{-1}}_{=:G} \overbrace{(y - \phi_X^{\theta\top}\mu)}^{=:\Delta}}_{=:e} + \log \underbrace{\left|\phi_X^{\theta\top}\Sigma\phi_X^{\theta} + \Lambda\right|}_{=:c} \Bigg)$$

What does the Optimizer need from us?

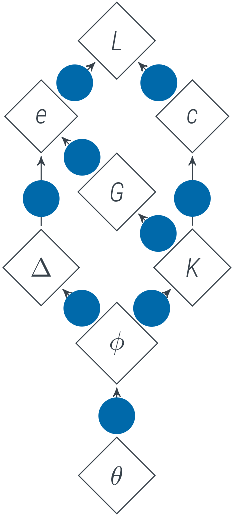
Automatic Differentiation



$$L(\theta) = \frac{1}{2}\Bigg(\underbrace{(y - {\phi_X^{\theta}}^\intercal \mu)^\intercal \underbrace{\Bigg(\underbrace{{\phi_X^{\theta}}^\intercal \Sigma \phi_X^{\theta}}_{=:K} + \Lambda\Bigg)^{-1}}_{=:G} \overbrace{(y - {\phi_X^{\theta}}^\intercal \mu)}^{=:\Delta}}_{=:e} + \log \underbrace{\left| {\phi_X^{\theta}}^\intercal \Sigma \phi_X^{\theta} + \Lambda \right|}_{=:c} \Bigg)$$

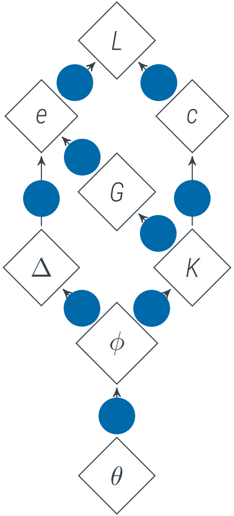
$$= m_9 + m_8 = ({m_6}^\intercal m_5 m_6) + \log |m_7 + \Lambda|$$

$$= \dots$$



$$L(\boldsymbol{\theta}) = \frac{1}{2}\Bigg(\underbrace{(y - \phi_X^{\boldsymbol{\theta}\intercal}\mu)^\intercal \underbrace{\Big(\underbrace{\phi_X^{\boldsymbol{\theta}\intercal}\Sigma\phi_X^{\boldsymbol{\theta}}}_{=:K} + \Lambda\Big)^{-1}}_{=:G} \overbrace{(y - \phi_X^{\boldsymbol{\theta}\intercal}\mu)}^{=:\Delta}}_{=:e} + \underbrace{\log\left|\phi_X^{\boldsymbol{\theta}\intercal}\Sigma\phi_X^{\boldsymbol{\theta}} + \Lambda\right|}_{=:c}\Bigg)$$

$$\begin{aligned}
\frac{\partial L}{\partial \theta} &= \frac{\partial L}{\partial e}\frac{\partial e}{\partial \theta} + \frac{\partial L}{\partial c}\frac{\partial c}{\partial \theta} = \dot{m}_9\frac{\partial e}{\partial \theta} + \dot{m}_8\frac{\partial c}{\partial \theta} = \dot{m}_9\left(\frac{\partial e}{\partial \Delta}\frac{\partial \Delta}{\partial \theta} + \frac{\partial e}{\partial G}\frac{\partial G}{\partial \theta}\right) + \dot{m}_8\frac{\partial c}{\partial K}\frac{\partial K}{\partial \theta} \\
&= \dot{m}_9\left(\dot{m}_6\frac{\partial \Delta}{\partial \theta} + \dot{m}_5\frac{\partial G}{\partial \theta}\right) + \dot{m}_8\dot{m}_7\frac{\partial K}{\partial \theta} = \dot{m}_9\left(\dot{m}_6\frac{\partial \Delta}{\partial \phi}\frac{\partial \phi}{\partial \theta} + \dot{m}_5\frac{\partial G}{\partial K}\frac{\partial K}{\partial \theta}\right) + \dot{m}_8\dot{m}_7\frac{\partial K}{\partial \theta} \\
&= \dot{m}_9\dot{m}_6\dot{m}_2\frac{\partial \phi}{\partial \theta} + (\dot{m}_9\dot{m}_5\dot{m}_4 + \dot{m}_8\dot{m}_7)\frac{\partial K}{\partial \theta} = \left(\dot{m}_9\dot{m}_6\dot{m}_2 + (\dot{m}_9\dot{m}_5\dot{m}_4 + \dot{m}_8\dot{m}_7)\frac{\partial K}{\partial \phi}\right)\frac{\partial \phi}{\partial \theta} \\
&= (\dot{m}_9\dot{m}_6\dot{m}_2 + (\dot{m}_9\dot{m}_5\dot{m}_4 + \dot{m}_8\dot{m}_7)\,\dot{m}_3)\,\frac{\partial \phi}{\partial \theta}\frac{\partial \theta}{\partial \theta} \\
&= (\dot{m}_9\dot{m}_6\dot{m}_2 + (\dot{m}_9\dot{m}_5\dot{m}_4 + \dot{m}_8\dot{m}_7)\,\dot{m}_3)\,\dot{m}_1 1
\end{aligned}$$



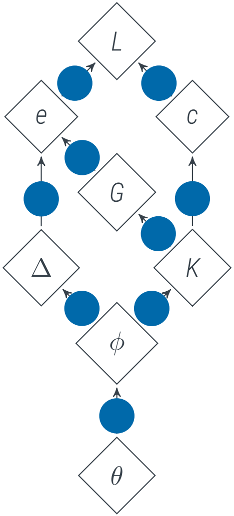
$$L(\theta) = \frac{1}{2}\Bigg(\underbrace{(y-\phi_X^{\boldsymbol{\theta}\top}\mu)^\top \underbrace{\Big(\underbrace{\phi_X^{\boldsymbol{\theta}\top}\Sigma\phi_X^{\boldsymbol{\theta}}}_{=:K}+\Lambda\Big)^{-1}}_{=:G}\overbrace{(y-\phi_X^{\boldsymbol{\theta}\top}\mu)}^{=:\Delta}}_{=:e} + \underbrace{\log\left|\phi_X^{\boldsymbol{\theta}\top}\Sigma\phi_X^{\boldsymbol{\theta}}+\Lambda\right|}_{=:c}\Bigg)$$

$$\dot{m}_9 = \frac{\partial L}{\partial e} = {}^1\!/_2 \qquad \dot{m}_8 = \frac{\partial L}{\partial c} = {}^1\!/_2 \qquad [\dot{m}_7]_{ij} = \frac{\partial c}{\partial K_{ij}} = K_{ij}^{-1}$$

$$[\dot{m}_6]_i = \frac{\partial e}{\partial \Delta_i} = 2[G\Delta]_i \qquad [\dot{m}_5]_{ij} = \frac{\partial e}{\partial G_{ij}} = \Delta_i\Delta_j \qquad [\dot{m}_4]_{ij,k\ell} = \frac{\partial G_{ij}}{\partial K_{k\ell}} = -G_{ik}G_{j\ell}$$

$$[\dot{m}_3]_{ij,ab} = \frac{\partial K_{ij}}{\partial \phi_{ab}} = \delta_{ia}[\Sigma\phi]_{bj} + \delta_{jb}[\Sigma\phi]_{kj}$$

$$[\dot{m}_2]_{i,ab} = \frac{\partial \Delta_i}{\partial \phi_{ab}} = -\delta_{ia}\mu_b \qquad [\dot{m}_1]_{ab,\ell} = \frac{\partial \phi_{ab}}{\partial \theta_\ell} = \text{your choice!}$$



$$L(\theta) = \frac{1}{2}\left(\underbrace{(y - \phi_X^{\boldsymbol{\theta}\top}\mu)^\top \underbrace{\left(\underbrace{\phi_X^{\boldsymbol{\theta}\top}\Sigma\phi_X^{\boldsymbol{\theta}}}_{=:K} + \Lambda\right)^{-1}}_{=:G} \overbrace{(y - \phi_X^{\boldsymbol{\theta}\top}\mu)}^{=:\Delta}}_{=:e} + \log\underbrace{\left|\phi_X^{\boldsymbol{\theta}\top}\Sigma\phi_X^{\boldsymbol{\theta}} + \Lambda\right|}_{=:c} \right)$$

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \phi}\frac{\partial \phi}{\partial \theta} =: \bar{m}_1 = \left(\frac{\partial L}{\partial \Delta}\frac{\partial \Delta}{\partial \phi} + \frac{\partial L}{\partial K}\frac{\partial K}{\partial \phi}\right)\frac{\partial \phi}{\partial \theta} =: (\bar{m}_2 + \bar{m}_3)\frac{\partial \phi}{\partial \theta}$$

$$\bar{m}_2 = \frac{\partial L}{\partial e}\frac{\partial e}{\partial \Delta}\frac{\partial \Delta}{\partial \phi} =: \bar{m}_6\frac{\partial \Delta}{\partial \phi} \qquad \bar{m}_3 = \left(\frac{\partial L}{\partial G}\frac{\partial G}{\partial K} + \frac{\partial L}{\partial c}\frac{\partial c}{\partial K}\right)\frac{\partial K}{\partial \phi} =: (\bar{m}_4 + \bar{m}_7)\frac{\partial K}{\partial \phi}$$

$$\bar{m}_4 = \frac{\partial L}{\partial e}\frac{\partial e}{\partial G}\frac{\partial G}{\partial K} =: \bar{m}_5\frac{\partial G}{\partial K} \qquad \bar{m}_5 = \frac{\partial L}{\partial e}\frac{\partial e}{\partial G} =: \bar{m}_9\frac{\partial e}{\partial G} \qquad \bar{m}_6 = \frac{\partial L}{\partial e}\frac{\partial e}{\partial \Delta} =: \bar{m}_9\frac{\partial e}{\partial \Delta}$$

$$\bar{m}_7 = \frac{\partial L}{\partial c}\frac{\partial c}{\partial K} =: \bar{m}_8\frac{\partial c}{\partial K} \qquad \bar{m}_8 = \bar{m}_9 = {}^1/_2$$

$\bar{w}_i = \frac{\partial L}{\partial \text{subgraph}_i}$ are known as *adjoints*. Traverse graph backward to collect the derivative. This is faster than forward-mode for single-output-many-input functions, but requires storing the above structure (known as a *Wengert list*). (cf. “Backpropagation”)

From Lecture 6:

$$\text{If } p(x) = \mathcal{N}(x; \mu, \Sigma) \qquad \text{and} \qquad p(y \mid x) = \mathcal{N}(y; A^\intercal x + b, \Lambda), \text{ then}$$

$$p(B^\intercal x + c \mid y) = \mathcal{N}[B^\intercal x + c; B^\intercal \mu + c + B^\intercal \Sigma A (A^\intercal \Sigma A + \Lambda)^{-1}(y - A^\intercal \mu - b),$$
$$B^\intercal \Sigma B - B^\intercal \Sigma A (A^\intercal \Sigma A + \Lambda)^{-1} A^\intercal \Sigma B]$$

In our case $y = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ and $p(f^1, f^2) = \mathcal{GP}\left(\begin{bmatrix} f^1 \\ f^2 \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k^1 & 0 \\ 0 & k^2 \end{bmatrix}\right)$ Thus we can *extract* the individual signals f^1 and f^2 from the data y by setting $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and get, in particular

$$p(f^1_\bullet \mid y) = \mathcal{GP}(f^1; k^1_{\bullet X}(k^1_{XX} + k^2_{XX} + \Lambda)^{-1} y, \qquad k^1_{\bullet\circ} - k^1_{\bullet, X}(k^1_{XX} + k^2_{XX} + \Lambda)^{-1} k^2_{X,\circ})$$

Summary:

- An unstructured kernel regression model can only do so much. **Extrapolation** and extracting **structural knowledge** require prior knowledge about the **causal structure**.
- Linear models with elaborate features can be quite expressive, while remaining interpretable
- Complicated processes require complicated (and questionable!) prior assumptions

Please cite this course, as

```
@techreport{Tuebingen_ProbML23,
    title =
    {Probabilistic Machine Learning},
    author = {Hennig, Philipp},
    series = {Lecture Notes
        in Machine Learning},
    year = {2023},
    institution = {Tübingen AI Center}}
```

The ability to build structured predictive models is a **key skill**. Everyone can run a TensorFlow script! Masters of structured probabilistic inference are highly sought after.