• **Parametric models** assume some finite set of parameters $\theta$. Given the parameters, future predictions, $x$, are independent of the observed data, $\mathcal{D}$:

\[ P(x|\theta, \mathcal{D}) = P(x|\theta) \]

therefore $\theta$ capture everything there is to know about the data.

• So the complexity of the model is bounded even if the amount of data is unbounded. This makes them not very flexible.

• **Non-parametric models** assume that the data distribution cannot be defined in terms of such a finite set of parameters. But they can often be defined by assuming an *infinite dimensional* $\theta$. Usually we think of $\theta$ as a *function*.

• The amount of information that $\theta$ can capture about the data $\mathcal{D}$ can grow as the amount of data grows. This makes them more flexible.
Bayesian nonparametrics

A simple framework for modelling complex data.

Nonparametric models can be viewed as having infinitely many parameters

Examples of non-parametric models:

<table>
<thead>
<tr>
<th>Parametric</th>
<th>Non-parametric</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>polynomial regression</td>
<td>Gaussian processes</td>
<td>function approx.</td>
</tr>
<tr>
<td>logistic regression</td>
<td>Gaussian process classifiers</td>
<td>classification</td>
</tr>
<tr>
<td>mixture models, k-means</td>
<td>Dirichlet process mixtures</td>
<td>clustering</td>
</tr>
<tr>
<td>hidden Markov models</td>
<td>infinite HMMs</td>
<td>time series</td>
</tr>
<tr>
<td>factor analysis / pPCA / PMF</td>
<td>infinite latent factor models</td>
<td>feature discovery</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Nonlinear regression and Gaussian processes

Consider the problem of nonlinear regression:
You want to learn a function $f$ with error bars from data $D = \{X, y\}$

A Gaussian process defines a distribution over functions $p(f)$ which can be used for Bayesian regression:

$$p(f|D) = \frac{p(f)p(D|f)}{p(D)}$$

Let $f = (f(x_1), f(x_2), \ldots, f(x_n))$ be an $n$-dimensional vector of function values evaluated at $n$ points $x_i \in \mathcal{X}$. Note, $f$ is a random variable.

**Definition:** $p(f)$ is a Gaussian process if for any finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the marginal distribution over that subset $p(f)$ is multivariate Gaussian.
A picture
Neural networks and Gaussian processes

Bayesian neural network

Data: $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N} = (X, y)$

Parameters $\theta$ are the weights of the neural net

- Parameter prior: $p(\theta | \alpha)$
- Parameter posterior: $p(\theta | \alpha, \mathcal{D}) \propto p(y | X, \theta)p(\theta | \alpha)$
- Prediction: $p(y' | \mathcal{D}, x', \alpha) = \int p(y' | x', \theta)p(\theta | \mathcal{D}, \alpha) \, d\theta$

A Gaussian process models functions $y = f(x)$

A multilayer perceptron (neural network) with infinitely many hidden units and Gaussian priors on the weights $\rightarrow$ a GP (Neal, 1996)

See also recent work on Deep Gaussian Processes (Damianou and Lawrence, 2013)