

Bayesian regression and non-parametric hypothesis testing

(Some random things you can do with kernels)

heiko.strathmann@gmail.com

Shogun Toolbox Workshop

July 12, 2013

Two Subjects?

- ▶ **Gaussian Processes:** Building block of many state-of-the-art regression and classification methods.
 - ▶ (Logistic) Regression as Bayesian inference
 - ▶ Model selection
 - ▶ GSoC 2012 (Jacob Walker), GSoC 2013 (Roman Votyakov)

Two Subjects?

- ▶ **Gaussian Processes:** Building block of many state-of-the-art regression and classification methods.
 - ▶ (Logistic) Regression as Bayesian inference
 - ▶ Model selection
 - ▶ GSoC 2012 (Jacob Walker), GSoC 2013 (Roman Votyakov)
- ▶ **Embedding distributions into kernel spaces:**
 - ▶ The very basic idea
 - ▶ Two-sample testing (MMD)
 - ▶ independence testing (HSIC)
 - ▶ GSoC 2012 and afterwards (Me)

Two Subjects?

- ▶ **Gaussian Processes:** Building block of many state-of-the-art regression and classification methods.
 - ▶ (Logistic) Regression as Bayesian inference
 - ▶ Model selection
 - ▶ GSoC 2012 (Jacob Walker), GSoC 2013 (Roman Votyakov)
- ▶ **Embedding distributions into kernel spaces:**
 - ▶ The very basic idea
 - ▶ Two-sample testing (MMD)
 - ▶ independence testing (HSIC)
 - ▶ GSoC 2012 and afterwards (Me)
- ▶ **Some Shogun demos**

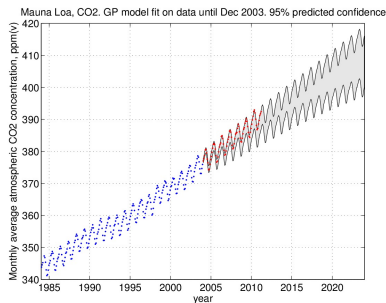
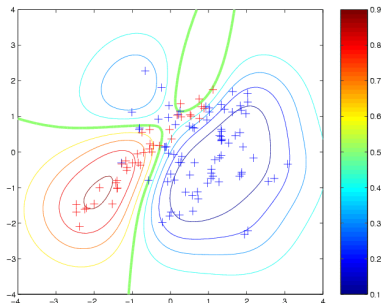
Are GPs and MMD related? Not really, but we wanted to talk about both :-)

Table of Contents

Gaussian Processes

Kernel-based hypothesis testing

These are quite useful



For example for the usual classification regression problems.

Gaussian Processes are Bayesian

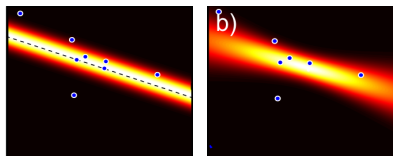
We get proper probability distributions for predictions.

- ▶ Calibrated confidence level in $[0, 1]$ for each prediction (without hacks)
- ▶ Aware of uncertainty
- ▶ Framework to learn model parameters from data (point-estimates or even integrate them out)

Gaussian Processes are Bayesian

We get proper probability distributions for predictions.

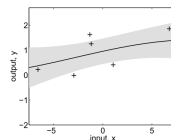
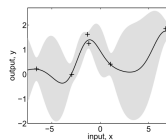
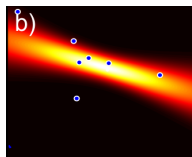
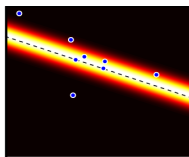
- ▶ Calibrated confidence level in $[0, 1]$ for each prediction (without hacks)
- ▶ Aware of uncertainty
- ▶ Framework to learn model parameters from data (point-estimates or even integrate them out)



Gaussian Processes are Bayesian

We get proper probability distributions for predictions.

- ▶ Calibrated confidence level in $[0, 1]$ for each prediction (without hacks)
- ▶ Aware of uncertainty
- ▶ Framework to learn model parameters from data (point-estimates or even integrate them out)

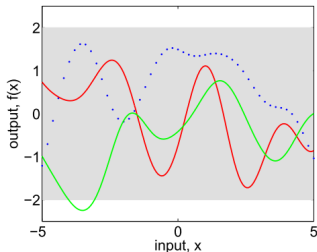


Gaussian Processes are non-parametric

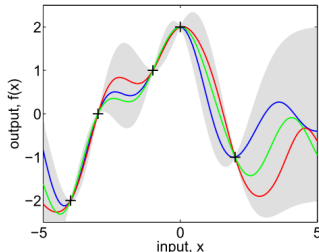
- ▶ No strong assumption to the data generating model (like in linear/logistic regression)
- ▶ In fact, there are many connections to kernel methods (feature embeddings)
- ▶ GPs involve distributions over model functions that can have arbitrary (smooth) shapes

Gaussian Processes are non-parametric

- ▶ No strong assumption to the data generating model (like in linear/logistic regression)
- ▶ In fact, there are many connections to kernel methods (feature embeddings)
- ▶ GPs involve distributions over model functions that can have arbitrary (smooth) shapes



(a), prior



(b), posterior

Gaussian Processes are simple: Just three rules

1. Sum Rule: $p(x) = \sum_y p(x, y) = \sum_y p(x|y)p(y)$

Gaussian Processes are simple: Just three rules

1. Sum Rule: $p(x) = \sum_y p(x, y) = \sum_y p(x|y)p(y)$
2. Bayes Rule: $p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}$

Gaussian Processes are simple: Just three rules

1. Sum Rule: $p(x) = \sum_y p(x, y) = \sum_y p(x|y)p(y)$
2. Bayes Rule: $p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}$
3. Manipulate Gaussian distributions: Given

$$p(x, y) = \mathcal{N} \left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \right),$$

then all

- ▶ $p(x|y)$
- ▶ $p(x) = \int p(x, y)dy = \int p(x|y)p(y)dy$
- ▶ $p(x|y)p(y)$

are Gaussian.

The basic idea, formally

A **Gaussian process** is a collection of random variables, any finite number of which have a joint Gaussian distribution.

The basic idea, formally

A **Gaussian process** is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- ▶ We write a GP as a distribution over functions

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')),$$

- ▶ $m(x)$ is the **mean** function
- ▶ $k(x, x')$ is the **covariance** function (or **kernel**).

How these functions look like?

- ▶ We can sample $f(x)$ for data $X := \{x_i\}_{i=1}^n$ and $X^* := \{x_j^*\}_{j=1}^m$
- ▶ Let $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{f}^* \in \mathbb{R}^m$ with $f_i = f(x_i)$ and $f_j^* = f(x_j^*)$
- ▶ Let $K(X, X^*) \in \mathbb{R}^{n \times m}$ be the covariance between X and X^*

How these functions look like?

- ▶ We can sample $f(x)$ for data $X := \{x_i\}_{i=1}^n$ and $X^* := \{x_j^*\}_{j=1}^m$
- ▶ Let $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{f}^* \in \mathbb{R}^m$ with $f_i = f(x_i)$ and $f_j^* = f(x_j^*)$
- ▶ Let $K(X, X^*) \in \mathbb{R}^{n \times m}$ be the covariance between X and X^*

\mathbf{f} and \mathbf{f}^* are jointly Gaussian distributed:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}\right)$$

How these functions look like?

- ▶ We can sample $f(x)$ for data $X := \{x_i\}_{i=1}^n$ and $X^* := \{x_j^*\}_{j=1}^m$
- ▶ Let $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{f}^* \in \mathbb{R}^m$ with $f_i = f(x_i)$ and $f_j^* = f(x_j^*)$
- ▶ Let $K(X, X^*) \in \mathbb{R}^{n \times m}$ be the covariance between X and X^*

\mathbf{f} and \mathbf{f}^* are jointly Gaussian distributed:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}\right)$$

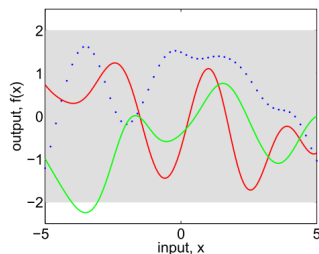
How the GP behave for test data X^* if it has seen training data X ?

$$\mathbf{f}^* | \mathbf{f} \sim \mathcal{N}(\mu, \Sigma), \text{ where}$$

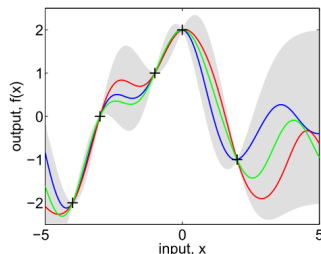
$$\mu = K(X^*, X)K(X, X)^{-1}\mathbf{f}$$

$$\Sigma = K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*)$$

Prior and Posterior, once again



(a), prior



(b), posterior

Intuition: Restrict distribution over functions to explain seen data

How to model data with a Gaussian Process

Model the joint distribution over data \mathbf{y} (i.e. labels) with corresponding covariates \mathbf{X} (i.e. features) as

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}),$$

where $p(\mathbf{f})$ is the GP prior (as before), and $p(\mathbf{y}|\mathbf{f})$ is the likelihood of the data given the latent variable \mathbf{f}

How to model data with a Gaussian Process

Model the joint distribution over data \mathbf{y} (i.e. labels) with corresponding covariates \mathbf{X} (i.e. features) as

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}),$$

where $p(\mathbf{f})$ is the GP prior (as before), and $p(\mathbf{y}|\mathbf{f})$ is the likelihood of the data given the latent variable \mathbf{f} , for example

- ▶ $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_{\text{noise}}^2 I)$ – regression with noisy observations

How to model data with a Gaussian Process

Model the joint distribution over data \mathbf{y} (i.e. labels) with corresponding covariates \mathbf{X} (i.e. features) as

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}),$$

where $p(\mathbf{f})$ is the GP prior (as before), and $p(\mathbf{y}|\mathbf{f})$ is the likelihood of the data given the latent variable \mathbf{f} , for example

- ▶ $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_{\text{noise}}^2 I)$ – regression with noisy observations
- ▶ $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n \sigma(y_i, f_i)$ – binary classification with activation function $\sigma : \{-1, +1\} \times \mathbb{R} \rightarrow [0, 1]$

How to model data with a Gaussian Process

Model the joint distribution over data \mathbf{y} (i.e. labels) with corresponding covariates \mathbf{X} (i.e. features) as

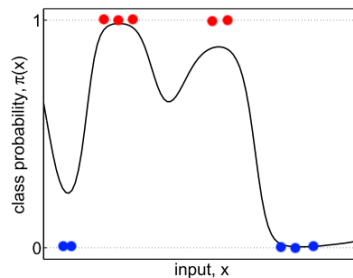
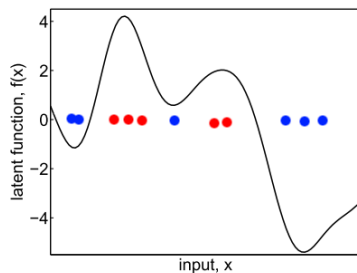
$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}),$$

where $p(\mathbf{f})$ is the GP prior (as before), and $p(\mathbf{y}|\mathbf{f})$ is the likelihood of the data given the latent variable \mathbf{f} , for example

- ▶ $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_{\text{noise}}^2 I)$ – regression with noisy observations
- ▶ $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n \sigma(y_i, f_i)$ – binary classification with activation function $\sigma : \{-1, +1\} \times \mathbb{R} \rightarrow [0, 1]$
- ▶ many more, e.g. multi-class logit \Rightarrow No need for competitions (OvO) as for SVMs, while still obtaining calibrated probabilities

Example likelihood: Logit-based binary classification

$$\sigma(y, f) = \frac{1}{1 + \exp(-yf)}$$



Predictions: Averaging over all possible \mathbf{f}

Recall

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}).$$

Given data \mathbf{y} and covariates X , we are interested in label predictions for unseen covariates X^* , i.e.,

Predictions: Averaging over all possible \mathbf{f}

Recall

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}).$$

Given data \mathbf{y} and covariates X , we are interested in label predictions for unseen covariates X^* , i.e.,

$$\begin{aligned} p(\mathbf{y}^*|\mathbf{y}) &= \int p(\mathbf{y}^*, \mathbf{f}^*|\mathbf{y}) d\mathbf{f}^* \\ &= \int p(\mathbf{y}^*|\mathbf{f}^*)p(\mathbf{f}^*|\mathbf{y}) d\mathbf{f}^* \\ &= \text{average } p(\mathbf{y}^*|\mathbf{f}^*) \text{ over all possibilities of } p(\mathbf{f}^*|\mathbf{y}) \end{aligned}$$

Predictions: Averaging over all possible \mathbf{f}

Recall

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}).$$

Given data \mathbf{y} and covariates X , we are interested in label predictions for unseen covariates X^* , i.e.,

$$\begin{aligned} p(\mathbf{y}^*|\mathbf{y}) &= \int p(\mathbf{y}^*, \mathbf{f}^*|\mathbf{y}) d\mathbf{f}^* \\ &= \int p(\mathbf{y}^*|\mathbf{f}^*)p(\mathbf{f}^*|\mathbf{y}) d\mathbf{f}^* \\ &= \text{average } p(\mathbf{y}^*|\mathbf{f}^*) \text{ over all possibilities of } p(\mathbf{f}^*|\mathbf{y}) \end{aligned}$$

Problem: Need to know $p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$.

Problem: Need to know $p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$

We know $p(\mathbf{f}^*|\mathbf{f})$, apply Bayes Rule to get second term

$$\text{posterior} = p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f}} = \frac{\text{likelihood} \times \text{GP-prior}}{\text{marginal likelihood}}.$$

Problem: Need to know $p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$

We know $p(\mathbf{f}^*|\mathbf{f})$, apply Bayes Rule to get second term

$$\text{posterior} = p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f}} = \frac{\text{likelihood} \times \text{GP-prior}}{\text{marginal likelihood}}.$$

► **Regression:** $p(\mathbf{y}|\mathbf{f})$ is Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is Gaussian \Rightarrow :-)

Problem: Need to know $p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$

We know $p(\mathbf{f}^*|\mathbf{f})$, apply Bayes Rule to get second term

$$\text{posterior} = p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f}} = \frac{\text{likelihood} \times \text{GP-prior}}{\text{marginal likelihood}}.$$

- ▶ **Regression:** $p(\mathbf{y}|\mathbf{f})$ is Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is Gaussian \Rightarrow :-)
- ▶ **Binary Classification with Laplace Approximation:**
 - ▶ $p(\mathbf{y}|\mathbf{f})$ is non-Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is not Gaussian \Rightarrow :-(
 \Rightarrow Approximate $p(\mathbf{f}|\mathbf{y})$ with a Gaussian \Rightarrow :-)

Problem: Need to know $p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$

We know $p(\mathbf{f}^*|\mathbf{f})$, apply Bayes Rule to get second term

$$\text{posterior} = p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f}} = \frac{\text{likelihood} \times \text{GP-prior}}{\text{marginal likelihood}}.$$

- ▶ **Regression:** $p(\mathbf{y}|\mathbf{f})$ is Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is Gaussian $\Rightarrow :-)$
- ▶ **Binary Classification with Laplace Approximation:**
 - ▶ $p(\mathbf{y}|\mathbf{f})$ is non-Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is not Gaussian $\Rightarrow :-)$
 \Rightarrow Approximate $p(\mathbf{f}|\mathbf{y})$ with a Gaussian $\Rightarrow :-)$
 - ▶ Find the (unique) maximum of $p(\mathbf{y}|\mathbf{f}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f})$,
 - ▶ then do a second order Taylor expansion around the mode.
 - ▶ Solved in practice, similarities to SVM

Problem: Need to know $p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$

We know $p(\mathbf{f}^*|\mathbf{f})$, apply Bayes Rule to get second term

$$\text{posterior} = p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f}} = \frac{\text{likelihood} \times \text{GP-prior}}{\text{marginal likelihood}}.$$

- ▶ **Regression:** $p(\mathbf{y}|\mathbf{f})$ is Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is Gaussian $\Rightarrow :-)$
- ▶ **Binary Classification with Laplace Approximation:**
 - ▶ $p(\mathbf{y}|\mathbf{f})$ is non-Gaussian $\Rightarrow p(\mathbf{f}|\mathbf{y})$ is not Gaussian $\Rightarrow :-)$
 \Rightarrow Approximate $p(\mathbf{f}|\mathbf{y})$ with a Gaussian $\Rightarrow :-)$
 - ▶ Find the (unique) maximum of $p(\mathbf{y}|\mathbf{f}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f})$,
 - ▶ then do a second order Taylor expansion around the mode.
 - ▶ Solved in practice, similarities to SVM
- ▶ Many more: variational methods, sparsity methods, ...

Model Selection: Hyperparameters

Covariance function have parameters, e.g., the Gaussian kernel

$$k(x, x') = \gamma^2 \exp \left(-\frac{\|x - x'\|_2^2}{2\sigma^2} \right)$$

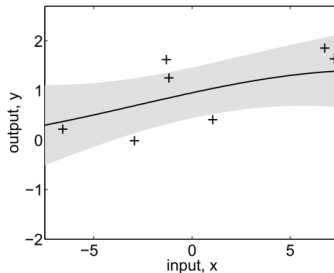
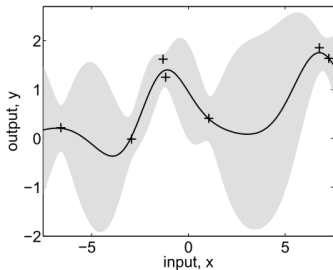
has parameter σ , which controls the model complexity. Which one to take?

Model Selection: Hyperparameters

Covariance function have parameters, e.g., the Gaussian kernel

$$k(x, x') = \gamma^2 \exp \left(-\frac{\|x - x'\|_2^2}{2\sigma^2} \right)$$

has parameter σ , which controls the model complexity. Which one to take?



Model selection: ask the machine learning guy

- ▶ “Just select the **best** parameter...” Is that always reasonable?

Model selection: ask the machine learning guy

- ▶ “Just select the **best** parameter...” Is that always reasonable?
- ▶ “Just do a **grid-search**”. That’s ugly, in fact, we can do nicer.

Model selection: ask the machine learning guy

- ▶ “Just select the **best** parameter...” Is that always reasonable?
- ▶ “Just do a **grid-search**”. That’s ugly, in fact, we can do nicer.
- ▶ Recall the marginal likelihood of a GP, which is the **averaged** likelihood over latent functions f ,

$$p(y|\theta) = \int p(y|f)p(f|\theta)df,$$

where now the **hyper-parameters** θ , which influence the GP prior $p(f|\theta)$, are included (e.g. $\theta = \{\sigma, \gamma\}$). We can maximise this, for example with gradient descent.

Model selection: ask the machine learning guy

- ▶ “Just select the **best** parameter...” Is that always reasonable?
- ▶ “Just do a **grid-search**”. That’s ugly, in fact, we can do nicer.
- ▶ Recall the marginal likelihood of a GP, which is the **averaged** likelihood over latent functions f ,

$$p(y|\theta) = \int p(y|f)p(f|\theta)df,$$

where now the **hyper-parameters** θ , which influence the GP prior $p(f|\theta)$, are included (e.g. $\theta = \{\sigma, \gamma\}$). We can maximise this, for example with gradient descent.

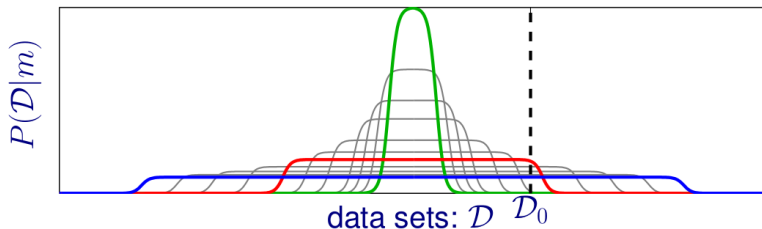
- ▶ Over-fitting? No! Ehm wait ... why?

Occam's Razor for $p(y|\theta) = \int p(y|f)p(f|\theta)df$

We average over all possible latent models f . If $p(f|\theta)$ is very rich, each element will only contribute little – even if $p(y|f)$ is large.

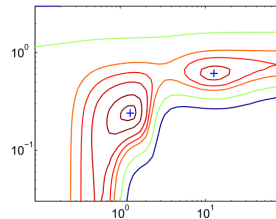
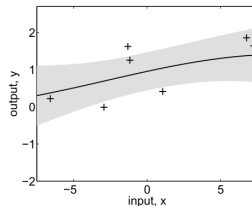
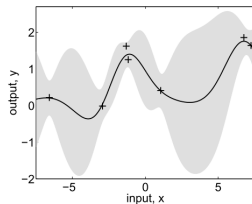
Occam's Razor for $p(y|\theta) = \int p(y|f)p(f|\theta)df$

We average over all possible latent models f . If $p(f|\theta)$ is very rich, each element will only contribute little – even if $p(y|f)$ is large.



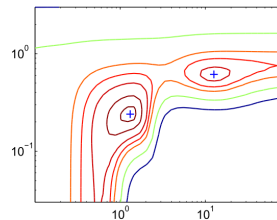
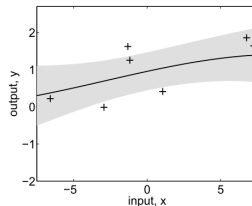
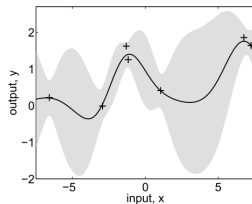
- ▶ **Too simple:** Cannot explain many datasets, $p(y|f)$ small
- ▶ **Too complex:** Can explain data, but $p(f|\theta)$ small
- ▶ **Just complex enough** to explain data

Model selection: ask the Bayesian guy



- When in doubt (small datasets), why not use **all possible θ** ?

Model selection: ask the Bayesian guy



- ▶ When in doubt (small datasets), why not use **all possible θ** ?
- ▶ Possible within GP framework, compute posterior

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta},$$

and average predictions. Usually done via MCMC. Tricky.

Demo Time!

Table of Contents

Gaussian Processes

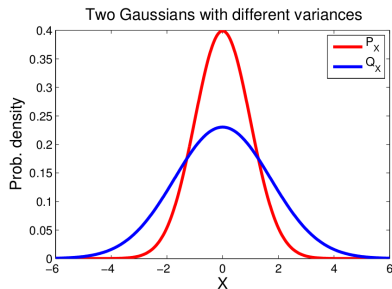
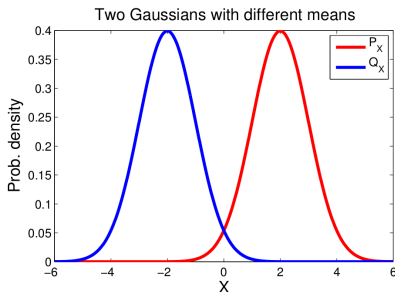
Kernel-based hypothesis testing

How to detect differences? $p(x) = q(y)$?

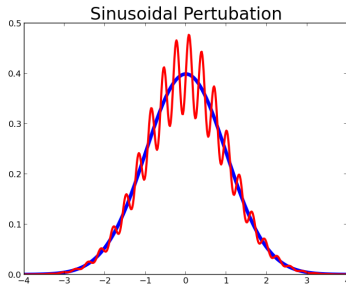
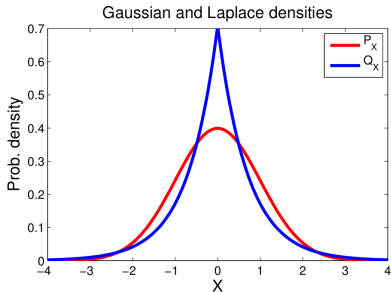
Given two probability distributions p, q on a domain \mathcal{X} , and two finite sets of iid samples drawn from them

$$X = \{x_i\}_{i=1}^n \sim p \quad Y = \{y_j\}_{j=1}^m \sim q,$$

can we decide whether $p \neq q$ with high confidence?



How to detect differences? $p(x) = q(y)$?



How to detect dependence? $p(x, y) = p(x)q(y)$?

... no doubt there is great pressure on governments and municipal governments in relation to the issue of child care, but the services de garde, reality is that there have been no cuts to child care funding from the federal to the provinces. In fact, we have increased federal investments for early childhood development. ...

... il est évident que les ordres de gouvernements provinciaux et municipaux subissent de fortes pressions en ce qui concerne les services de garde, mais le gouvernement n'a pas réduit le financement qu'il verse aux provinces pour les services de garde. Au contraire, nous avons augmenté le financement fédéral pour le développement des jeunes enfants. ...

Challenging two-sample testing

Two-sample testing in high dimensional distributions with complex structure:

- ▶ Classical methods are often not feasible.

Challenging two-sample testing

Two-sample testing in high dimensional distributions with complex structure:

- ▶ Classical methods are often not feasible.
- ▶ Strings/text-data (Websites, primary structure), Graphs (Protein networks, social media), etc
- ▶ Mapping to vector space needed.

Challenging two-sample testing

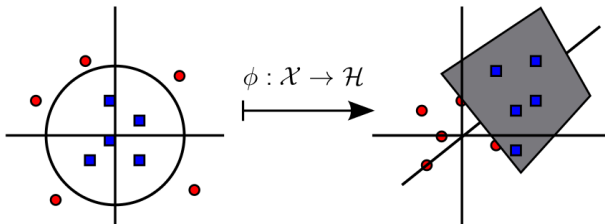
Two-sample testing in high dimensional distributions with complex structure:

- ▶ Classical methods are often not feasible.
- ▶ Strings/text-data (Websites, primary structure), Graphs (Protein networks, social media), etc
- ▶ Mapping to vector space needed.
- ▶ We also do not want to make assumptions to p and q (like t-test).

The answer: with kernels!



Reproducing Kernel Hilbert Spaces – the classic slide



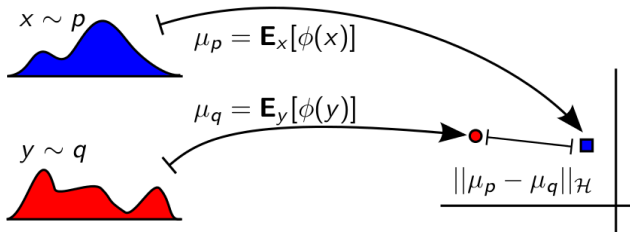
Positive semi-definite kernel: $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$

Reproducing Kernel Hilbert Spaces – mean embeddings

That was fun! Now let's do it with probability distributions

Reproducing Kernel Hilbert Spaces – mean embeddings

That was fun! Now let's do it with probability distributions



Maximum Mean Discrepancy

Let x, y be random variables with attached probability distributions p, q respectively. The kernel **Maximum Mean Discrepancy** is given by

$$\begin{aligned}\text{MMD}^2[\mathcal{F}, p, q] &= \|\mu_p - \mu_q\|_{\mathcal{H}}^2 \\ &= \mathbf{E}_{x, x', y, y'} [k(x, x') + k(y, y') - k(x, y') - k(x', y)]\end{aligned}$$

where x', y' are independent copies of x, y .

Maximum Mean Discrepancy

Let x, y be random variables with attached probability distributions p, q respectively. The kernel **Maximum Mean Discrepancy** is given by

$$\begin{aligned}\text{MMD}^2[\mathcal{F}, p, q] &= \|\mu_p - \mu_q\|_{\mathcal{H}}^2 \\ &= \mathbf{E}_{x, x', y, y'} [k(x, x') + k(y, y') - k(x, y') - k(x', y)]\end{aligned}$$

where x', y' are independent copies of x, y .

One can show

$$\|\mu_p - \mu_q\|_{\mathcal{H}}^2 = 0 \Leftrightarrow p = q$$

for certain kernels. **Any** pair of distributions can be distinguished.

Sounds good, let's compute it from data

Given data X, Y with $|X| = |Y| = m$, a **quadratic time estimate** is

$$\frac{1}{m(m-1)} \sum_{i=1}^m \sum_{j \neq i}^m k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i).$$

Powerful, a bit complicated to compute the test.

Sounds good, let's compute it from data

Given data X, Y with $|X| = |Y| = m$, a **quadratic time estimate** is

$$\frac{1}{m(m-1)} \sum_{i=1}^m \sum_{j \neq i}^m k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i).$$

Powerful, a bit complicated to compute the test. An alternative is the **linear time estimate**: divide data into two halves and compute

$$\frac{2}{m} \sum_{i=1}^{\frac{m}{2}} k(x_{2i}, x_{2i-1}) + k(y_{2i}, y_{2i-1}) - k(x_{2i}, y_{2i-1}) - k(x_{2i-1}, y_{2i}).$$

Convenient properties: Possible to stream (big data♥), easy to compute the test.

Sounds good, let's compute it from data

Given data X, Y with $|X| = |Y| = m$, a **quadratic time estimate** is

$$\frac{1}{m(m-1)} \sum_{i=1}^m \sum_{j \neq i}^m k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i).$$

Powerful, a bit complicated to compute the test. An alternative is the **linear time estimate**: divide data into two halves and compute

$$\frac{2}{m} \sum_{i=1}^{\frac{m}{2}} k(x_{2i}, x_{2i-1}) + k(y_{2i}, y_{2i-1}) - k(x_{2i}, y_{2i-1}) - k(x_{2i-1}, y_{2i}).$$

Convenient properties: Possible to stream (big data♥), easy to compute the test. Optimal kernel selection possible (!)

Independence Test: Hilbert Schmidt Independence Criterion

Idea (roughly, skipping many details): Compute MMD between $p(x)q(y)$ and $p(x, y)$. An estimate is

$$\text{HSIC} = \frac{1}{m^2} \text{trace}(KHLH),$$

where K, L are the kernel matrices on data X, Y respectively and $H = I - \frac{1}{n} \mathbf{1}\mathbf{1}^\top$.

Independence Test: Hilbert Schmidt Independence Criterion

Idea (roughly, skipping many details): Compute MMD between $p(x)q(y)$ and $p(x, y)$. An estimate is

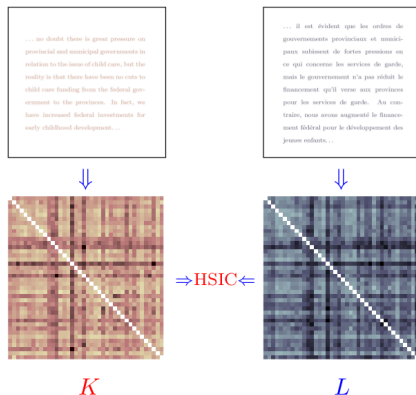
$$\text{HSIC} = \frac{1}{m^2} \text{trace}(KHLH),$$

where K, L are the kernel matrices on data X, Y respectively and $H = I - \frac{1}{n} \mathbf{1}\mathbf{1}^T$.


- ▶ One can also view this as a norm of the kernel-covariance operator between p and q
- ▶ The test, as for the quadratic time MMD, is a bit messy to compute (permutation/bootstrapping)

A nice application

Use a HSIC with a (spectrum) string kernel is able to detect significant dependence between EU parliament translations of the same text into two different languages.



Demo time!

-  Mark Girolami and Simon Rogers.
Variational Bayesian Multinomial Probit Regression with
Gaussian Process Priors.
[Neural Computation](#), 18:1790–1817, 2006.
-  Arthur Gretton, Kenji Fukumizu, CH Teo, and Le Song.
A kernel statistical test of independence.
[Advances in Neural Information Processing Systems](#), pages
1–8, 2008.
-  Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch,
Bernhard Schölkopf, and Alexander Smola.
A Kernel Two-Sample Test.
[Journal of Machine Learning Research](#), 13:671–721, 2012a.
-  Arthur Gretton, Bharath Sriperumbudur, Dino Sejdinovic,
Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano
Pontil, and Kenji Fukumizu.

Optimal kernel choice for large-scale two-sample tests.
In [Advances in Neural Information Processing Systems](#), 2012b.



Simon J D Prince.

[Computer vision : models , learning and inference.](#)
Cambridge University Press, 2012.



Carl Edward Rasmussen and Zoubin Ghahramani.

Occam's Razor.

[Advances in Neural Information Processing Systems](#), 13, 2001.



Carl Edward Rasmussen and Christopher K. I. Williams.

[Gaussian Processes in Machine Learning.](#)
MIT Press, 2006.



Heiko Strathmann.

M.Sc. Adaptive Large-Scale Kernel Two-Sample Testing, 2012.



C.K.I. Williams and D. Barber.

Bayesian classification with Gaussian processes.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(12):1342–1351, 1998.

ISSN 01628828.

Thank you for your attention!

Questions?