Statistical Data Mining and Machine Learning
Hilary Term 2016

Dino Sejdinovic
Department of Statistics
Oxford

Slides and other materials available at:
http://www.stats.ox.ac.uk/~sejdinov/sdmml

Bayesian Learning

Maximum Likelihood Principle

- A generative model for training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ given a parameter vector θ:

$$y_i \sim (\pi_1, \ldots, \pi_K), \quad x|y_i \sim g_{y_i}(x) = p(x|\phi_{y_i})$$

- k-th class conditional density assumed to have a parametric form for $g_k(x) = p(x|\phi_k)$ and all parameters are given by $\theta = (\pi_1, \ldots, \pi_K; \phi_1, \ldots, \phi_K)$
- Generative process defines the **likelihood function**: the joint distribution of all the observed data $p(\mathcal{D}|\theta)$ given a parameter vector θ .
- Process of generative learning consists of computing the MLE $\widehat{\theta}$ of θ based on $\mathcal{D}$:

$$\widehat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, p(\mathcal{D}|\theta)$$

- We then use a plug-in approach to perform classification

$$f_{\widehat{\theta}}(x) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\, \mathbb{P}_{\widehat{\theta}}(Y = k|X = x) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \frac{\widehat{\pi}_k p(x|\widehat{\phi}_k)}{\sum_{j=1}^K \widehat{\pi}_j p(x|\widehat{\phi}_j)}$$

The Bayesian Learning Framework

- Being Bayesian: **treat parameter vector θ as a random variable**: process of learning is then **computation of the posterior distribution** $p(\theta|\mathcal{D})$.
- In addition to the likelihood $p(\mathcal{D}|\theta)$ need to specify a **prior distribution** $p(\theta)$.
- Posterior distribution is then given by the **Bayes Theorem**:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$$

  - **Likelihood**: $p(\mathcal{D}|\theta)$
  - **Prior**: $p(\theta)$
  - **Posterior**: $p(\theta|\mathcal{D})$
  - **Marginal likelihood**: $p(\mathcal{D}) = \int_\Theta p(\mathcal{D}|\theta)p(\theta)d\theta$

- Summarizing the posterior:
  - **Posterior mode**: $\widehat{\theta}^{\text{MAP}} = \operatorname{argmax}_{\theta \in \Theta} p(\theta|\mathcal{D})$ (maximum a posteriori).
  - **Posterior mean**: $\widehat{\theta}^{\text{mean}} = \mathbb{E}\left[\theta|\mathcal{D}\right]$.
  - **Posterior variance**: $\text{Var}[\theta|\mathcal{D}]$.

Simple Example: Coin Tosses

- A simple example: We have a coin with probability ϕ of coming up heads. Model coin tosses as i.i.d. Bernoullis, 1 =head, 0 =tail.
- Estimate ϕ given a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$ of tosses.

$$p(\mathcal{D}|\phi) = \phi^{n_1}(1-\phi)^{n_0}$$

with $n_j = \sum_{i=1}^n \mathbb{1}(x_i = j)$.

- Maximum Likelihood estimate:

$$\hat{\phi}^{\text{ML}} = \frac{n_1}{n}$$

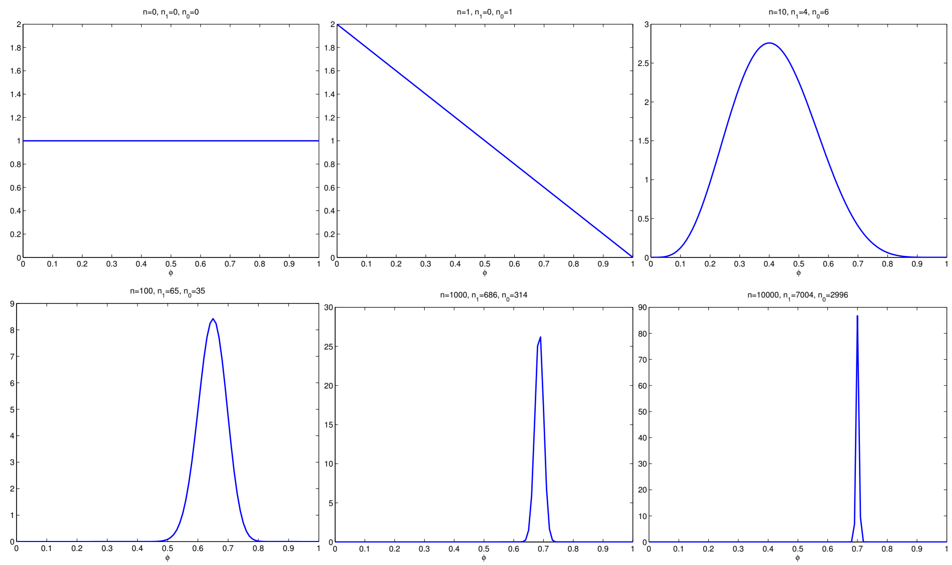
- Bayesian approach: treat the unknown parameter ϕ as a random variable. Simple prior: $\phi \sim \text{Uniform}[0,1]$, i.e., $p(\phi) = 1$ for $\phi \in [0,1]$. Posterior distribution:

$$p(\phi|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{\phi^{n_1}(1-\phi)^{n_0} \cdot 1}{p(\mathcal{D})}, \quad p(\mathcal{D}) = \int_0^1 \phi^{n_1}(1-\phi)^{n_0} d\phi = \frac{(n+1)!}{n_1! n_0!}$$

Posterior is a $\text{Beta}(n_1+1, n_0+1)$ distribution: $\widehat{\phi}^{\text{mean}} = \frac{n_1+1}{n+2}$.

Bayesian Learning | Review of Bayesian Inference

Simple Example: Coin Tosses



Posterior becomes behaves like the ML estimate as dataset grows and is peaked at true value $\phi^* = .7$.

Bayesian Learning | Review of Bayesian Inference

Simple Example: Coin Tosses

- All Bayesian reasoning is based on the posterior distribution.
  - Posterior mode: $\widehat{\phi}^{\text{MAP}} = \frac{n_1}{n}$
  - Posterior mean: $\widehat{\phi}^{\text{mean}} = \frac{n_1+1}{n+2}$
  - Posterior variance: $\text{Var}[\phi|\mathcal{D}] = \frac{1}{n+3}\widehat{\phi}^{\text{mean}}(1 - \widehat{\phi}^{\text{mean}})$
  - $(1-\alpha)$-credible regions: $(l, r) \subset [0,1]$ s.t. $\int_l^r p(\theta|\mathcal{D})d\theta = 1 - \alpha$.
- Consistency: Assuming that the true parameter value ϕ^* is given a non-zero density under the prior, the posterior distribution concentrates around the true value as $n \to \infty$.
- Rate of convergence?

Bayesian Learning | Review of Bayesian Inference

Simple Example: Coin Tosses

- The **posterior predictive distribution** is the conditional distribution of x_{n+1} given $\mathcal{D} = \{x_i\}_{i=1}^n$:

$$\begin{aligned} p(x_{n+1}|\mathcal{D}) &= \int_0^1 p(x_{n+1}|\phi, \mathcal{D})p(\phi|\mathcal{D})d\phi \\ &= \int_0^1 p(x_{n+1}|\phi)p(\phi|\mathcal{D})d\phi \\ &= (\widehat{\phi}^{\text{mean}})^{x_{n+1}}(1 - \widehat{\phi}^{\text{mean}})^{1-x_{n+1}} \end{aligned}$$

- We predict on new data by **averaging** the predictive distribution over the posterior. Accounts for uncertainty about ϕ.

Simple Example: Coin Tosses

- In this example, the posterior distribution has a known analytic form and is in the same Beta family as the prior: $\text{Uniform}[0,1] \equiv \text{Beta}(1,1)$.
- An example of a **conjugate prior**.
- A Beta distribution $\text{Beta}(a,b)$ with parameters $a, b > 0$ is an exponential family distribution with density

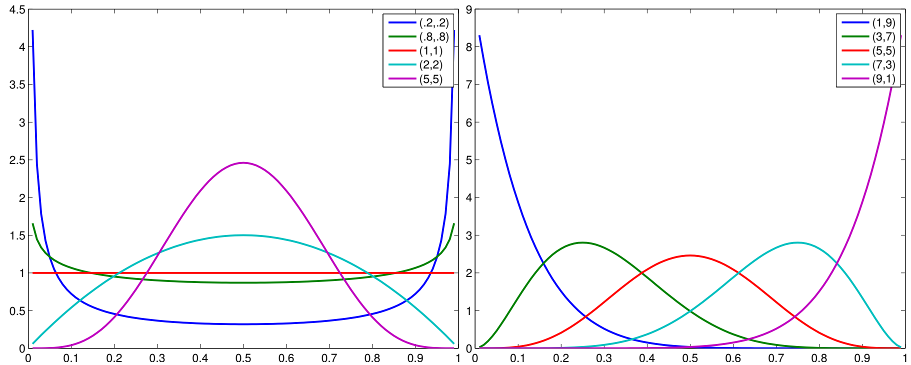
$$p(\phi|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^{a-1}(1-\phi)^{b-1}$$

  where $\Gamma(t) = \int_0^\infty u^{t-1}e^{-u}du$ is the gamma function.
- If the prior is $\phi \sim \text{Beta}(a,b)$, then the posterior distribution is

$$p(\phi|\mathcal{D},a,b) = \propto \phi^{a+n_1-1}(1-\phi)^{b+n_0-1}$$

  so is $\text{Beta}(a+n_1, b+n_0)$.
- Hyperparameters a and b are **pseudo-counts**, an imaginary initial sample that reflects our prior beliefs about ϕ.

Beta Distributions



Bayesian Inference on the Categorical Distribution

- Suppose we observe $\mathcal{D} = \{y_i\}_{i=1}^n$ with $y_i \in \{1,\ldots,K\}$, and model them as i.i.d. with pmf $\pi = (\pi_1,\ldots,\pi_K)$:

$$p(\mathcal{D}|\pi) = \prod_{i=1}^n \pi_{y_i} = \prod_{k=1}^K \pi_k^{n_k}$$

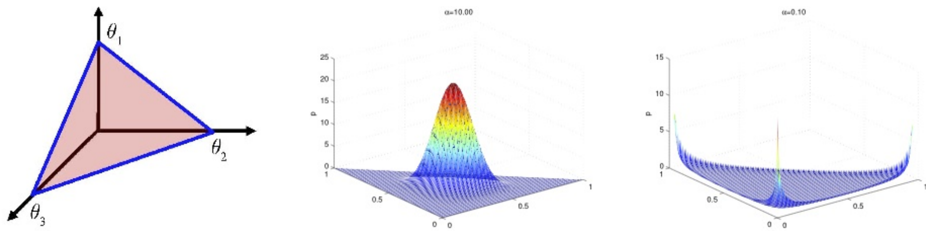
  with $n_k = \sum_{i=1}^n \mathbb{1}(y_i = k)$ and $\pi_k > 0$, $\sum_{k=1}^K \pi_k = 1$.
- The conjugate prior on π is the Dirichlet distribution $\text{Dir}(\alpha_1,\ldots,\alpha_K)$ with parameters $\alpha_k > 0$, and density

$$p(\pi) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1}$$

  on the probability simplex $\{\pi : \pi_k > 0, \sum_{k=1}^K \pi_k = 1\}$.
- The posterior is also Dirichlet $\text{Dir}(\alpha_1 + n_1, \ldots, \alpha_K + n_K)$.
- Posterior mean is

$$\hat{\pi}_k^{\text{mean}} = \frac{\alpha_k + n_k}{\sum_{j=1}^K \alpha_j + n_j}.$$

Dirichlet Distributions



(A) Support of the Dirichlet density for $K = 3$.
(B) Dirichlet density for $\alpha_k = 10$.
(C) Dirichlet density for $\alpha_k = 0.1$.

Naïve Bayes

- Return to the spam classification example with two-class naïve Bayes

$$p(x_i|\phi_k) = \prod_{j=1}^{p} \phi_{kj}^{x_i^{(j)}} (1-\phi_{kj})^{1-x_i^{(j)}}.$$

- Set $n_k = \sum_{i=1}^n \mathbb{1}\{y_i = k\}$, $n_{kj} = \sum_{i=1}^n \mathbb{1}(y_i = k, x_i^{(j)} = 1)$. MLE is:

$$\hat{\pi}_k = \frac{n_k}{n}, \qquad \hat{\phi}_{kj} = \frac{\sum_{i:y_i=k} x_i^{(j)}}{n_k} = \frac{n_{kj}}{n_k}.$$

- One problem: if the ℓ-th word did not appear in documents labelled as class k then $\hat{\phi}_{k\ell} = 0$ and

$$\mathbb{P}(Y = k|X = x \text{ with } \ell\text{-th entry equal to } 1)$$
$$\propto \hat{\pi}_k \prod_{j=1}^{p} \left(\hat{\phi}_{kj}\right)^{x^{(j)}} \left(1-\hat{\phi}_{kj}\right)^{1-x^{(j)}} = 0$$

  i.e. we will never attribute a new document containing word ℓ to class k (regardless of other words in it).

Bayesian Inference on Naïve Bayes model

- Under the Naïve Bayes model, the joint distribution of labels $y_i \in \{1, \ldots, K\}$ and data vectors $x_i \in \{0,1\}^p$ is

$$\prod_{i=1}^{n} p(x_i, y_i) = \prod_{i=1}^{n}\prod_{k=1}^{K} \left(\pi_k \prod_{j=1}^{p} \phi_{kj}^{x_i^{(j)}} (1-\phi_{kj})^{1-x_i^{(j)}} \right)^{\mathbb{1}(y_i=k)}$$
$$= \prod_{k=1}^{K} \pi_k^{n_k} \prod_{j=1}^{p} \phi_{kj}^{n_{kj}} (1-\phi_{kj})^{n_k - n_{kj}}$$

  where $n_k = \sum_{i=1}^n \mathbb{1}(y_i = k)$, $n_{kj} = \sum_{i=1}^n \mathbb{1}(y_i = k, x_i^{(j)} = 1)$.
- For conjugate prior, we can use $\mathrm{Dir}((\alpha_k)_{k=1}^K)$ for π, and $\mathrm{Beta}(a,b)$ for ϕ_{kj} independently.
- Because the likelihood factorizes, the posterior distribution over π and (ϕ_{kj}) also factorizes, and posterior for π is $\mathrm{Dir}((\alpha_k + n_k)_{k=1}^K)$, and for ϕ_{kj} is $\mathrm{Beta}(a + n_{kj}, b + n_k - n_{kj})$.

Bayesian Inference on Naïve Bayes model

- Given $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, want to predict a label $\tilde{y}$ for a new document $\tilde{x}$. We can calculate

$$p(\tilde{x}, \tilde{y} = k|\mathcal{D}) = p(\tilde{y} = k|\mathcal{D})p(\tilde{x}|\tilde{y} = k, \mathcal{D})$$

  with

$$p(\tilde{y} = k|\mathcal{D}) = \frac{\alpha_k + n_k}{\sum_{l=1}^K \alpha_l + n}$$
$$p(\tilde{x}^{(j)} = 1|\tilde{y} = k, \mathcal{D}) = \frac{a + n_{kj}}{a + b + n_k}$$

- Predicted class is

$$p(\tilde{y} = k|\tilde{x}, \mathcal{D}) = \frac{p(\tilde{y} = k|\mathcal{D})p(\tilde{x}|\tilde{y} = k, \mathcal{D})}{p(\tilde{x}|\mathcal{D})}$$

- Compared to ML plug-in estimator, pseudocounts help to “regularize” probabilities away from extreme values.

Bayesian Learning and Regularization

- Consider a Bayesian approach to logistic regression: introduce a multivariate normal prior for weight vector $w \in \mathbb{R}^p$, and a uniform (improper) prior for offset $b \in \mathbb{R}$. The prior density is:

$$p(b, w) = 1 \cdot (2\pi\sigma^2)^{-\frac{p}{2}} \exp\left(-\frac{1}{2\sigma^2}\|w\|_2^2\right)$$

- The posterior is

$$p(b, w|\mathcal{D}) \propto \exp\left(-\frac{1}{2\sigma^2}\|w\|_2^2 - \sum_{i=1}^{n} \log(1 + \exp(-y_i(b + w^\top x_i)))\right)$$

- The posterior mode is equivalent to minimizing the L_2-regularized empirical risk.
- Regularized empirical risk minimization is (often) equivalent to having a prior and finding a MAP estimate of the parameters.
  - L_2 regularization - multivariate normal prior.
  - L_1 regularization - multivariate Laplace prior.
- From a Bayesian perspective, the MAP parameters are just one way to summarize the posterior distribution.

Bayesian Model Selection

- A model $\mathcal{M}$ with a given set of parameters $\theta_{\mathcal{M}}$ consists of both the likelihood $p(\mathcal{D}|\theta_{\mathcal{M}})$ and the prior distribution $p(\theta_{\mathcal{M}})$.
  - One example model would consist of all Gaussian mixtures with K components and equal covariance (LDA): $\theta_{\text{LDA}} = (\pi_1, \dots, \pi_K; \mu_1, \dots, \mu_K; \Sigma)$, along with a prior on θ; another would allow different covariances (QDA) $\theta_{\text{QDA}} = (\pi_1, \dots, \pi_K; \mu_1, \dots, \mu_K; \Sigma_1, \dots, \Sigma_K)$.
- The posterior distribution

$$p(\theta_{\mathcal{M}}|\mathcal{D}, \mathcal{M}) = \frac{p(\mathcal{D}|\theta_{\mathcal{M}}, \mathcal{M})p(\theta_{\mathcal{M}}|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})}$$

- Marginal probability of the data under $\mathcal{M}$ (**Bayesian model evidence**):

$$p(\mathcal{D}|\mathcal{M}) = \int_{\Theta} p(\mathcal{D}|\theta_{\mathcal{M}}, \mathcal{M})p(\theta_{\mathcal{M}}|\mathcal{M})d\theta$$

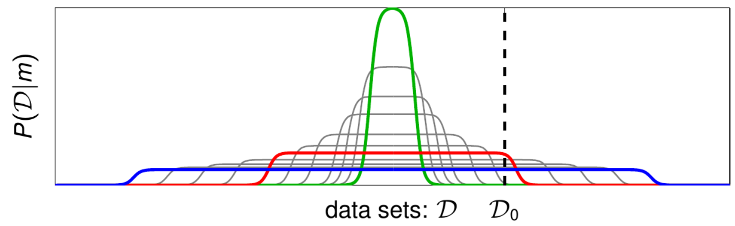
- Compare models using their **Bayes factors** $\frac{p(\mathcal{D}|\mathcal{M})}{p(\mathcal{D}|\mathcal{M}')}$

Bayesian Occam's Razor

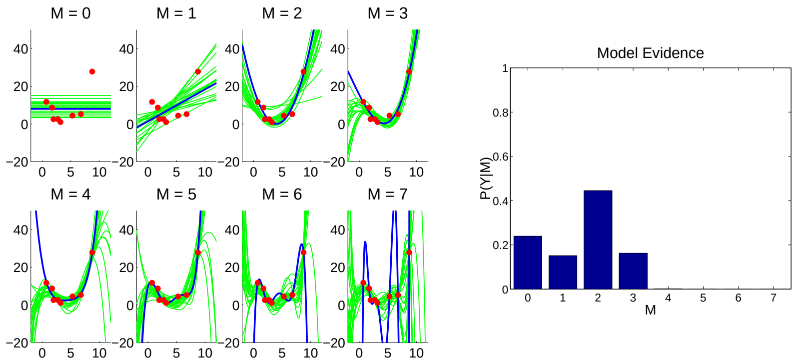
- **Occam's Razor**: of two explanations adequate to explain the same set of observations, the simpler should be preferred.

$$p(\mathcal{D}|\mathcal{M}) = \int_{\Theta} p(\mathcal{D}|\theta_{\mathcal{M}}, \mathcal{M})p(\theta_{\mathcal{M}}|\mathcal{M})d\theta$$

- Model evidence $p(\mathcal{D}|\mathcal{M})$ is the probability that a set of randomly selected parameter values inside the model would generate dataset $\mathcal{D}$.
- Models that are too simple are unlikely to generate the observed dataset.
- Models that are too complex can generate many possible dataset, so again, they are unlikely to generate that particular dataset at random.



Bayesian model comparison: Occam's razor at work



figures by M.Sahani

Bayesian Learning – Discussion

- Use probability distributions to reason about uncertainties of parameters (latent variables and parameters are treated in the same way).
- Model consists of the likelihood function **and** the prior distribution on parameters: allows to integrate prior beliefs and domain knowledge.
- Bayesian computation — most posteriors are intractable, and posterior needs to be approximated by:
  - Monte Carlo methods (MCMC and SMC).
  - Variational methods (variational Bayes, belief propagation etc).
- Prior usually has hyperparameters, i.e., $p(\theta) = p(\theta|\psi)$. How to choose ψ?
  - Be Bayesian about ψ as well — choose a hyperprior $p(\psi)$ and compute $p(\psi|\mathcal{D})$.
  - Maximum Likelihood II — $\hat{\psi} = \text{argmax}_{\psi \in \Psi} p(\mathcal{D}|\psi)$.

$$p(\mathcal{D}|\psi) = \int p(\mathcal{D}|\theta)p(\theta|\psi)d\theta$$

$$p(\psi|\mathcal{D}) = \frac{p(\mathcal{D}|\psi)p(\psi)}{p(\mathcal{D})}$$

Bayesian Learning – Further Reading

- Videolectures by Zoubin Ghahramani:
  Bayesian Learning and Graphical models.
- Gelman et al. Bayesian Data Analysis.
- Kevin Murphy. Machine Learning: a Probabilistic Perspective.
- E. T. Jaynes. Probability Theory: The Logic of Science.

Gaussian Processes

Parametric vs Nonparametric models

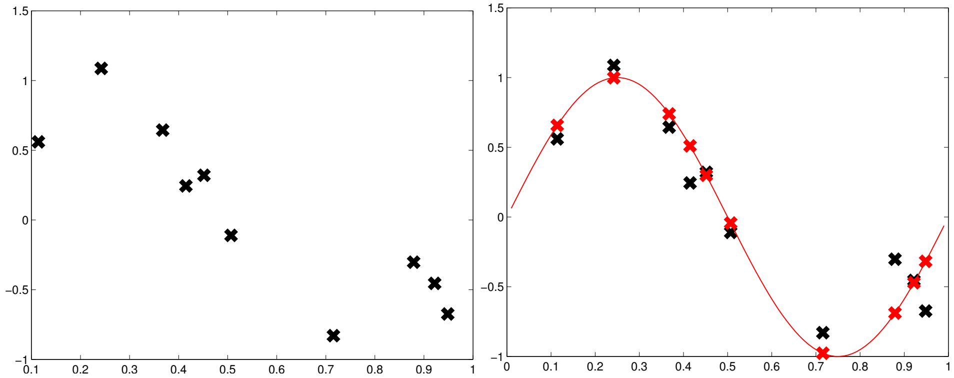
- **Parametric models** have a fixed finite number of parameters, regardless of the dataset size. In the Bayesian setting, given the parameter vector θ, the predictions are independent of the data $\mathcal{D}$.

  $$p(\tilde{x}, \theta | \mathcal{D}) = p(\theta|\mathcal{D}) p(\tilde{x}|\theta)$$

  Parameters can be thought of as a data summary: communication channel flows from data to the predictions through the parameters.
  Model-based learning (e.g., mixture of K multivariate normals)

- **Nonparametric models** allow the number of "parameters" to grow with the dataset size. Alternatively, predictions depend on the data (and the hyperparameters).
  Memory-based learning (e.g., kernel density estimation)

Regression



- We are given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Regression: learn the underlying real-valued function $f(x)$.

Different Flavours of Regression

- We can model response y_i as a noisy version of the underlying function f evaluated at input x_i:

$$y_i | f, x_i \sim \mathcal{N}(f(x_i), \sigma^2)$$

Appropriate loss: $L(y, f(x)) = (y - f(x))^2$

- **Frequentist Parametric** approach: model f as f_θ for some parameter vector θ. Fit θ by ML / ERM with squared loss (linear regression).
- **Frequentist Nonparametric** approach: model f as the unknown parameter taking values in an infinite-dimensional space of functions. Fit f by regularized ML / ERM with squared loss (kernel ridge regression)
- **Bayesian Parametric** approach: model f as f_θ for some parameter vector θ. Put a prior on θ and compute a posterior $p(\theta|\mathcal{D})$ (Bayesian linear regression).
- **Bayesian Nonparametric** approach: treat f as the random variable taking values in an infinite-dimensional space of functions. Put a prior over functions $f \in \mathcal{F}$, and compute a posterior $p(f|\mathcal{D})$ (Gaussian Process regression).

- Just work with the function values at the inputs $\mathbf{f} = (f(x_1), \ldots, f(x_n))^\top$
- What properties of the function can we incorporate?
  - Multivariate normal prior on $\mathbf{f}$:

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$$

  - Use a kernel function k to define $\mathbf{K}$:

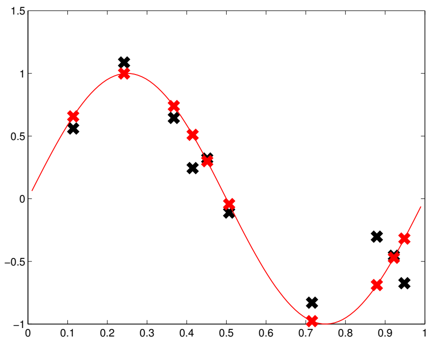
$$\mathbf{K}_{ij} = k(x_i, x_j)$$

  - Expect regression functions to be smooth: If x and x' are close by, then $f(x)$ and $f(x')$ have similar values, i.e. strongly correlated.

$$\begin{pmatrix} f(x) \\ f(x') \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} k(x,x) & k(x,x') \\ k(x',x) & k(x',x') \end{pmatrix} \right)$$

In particular, want $k(x,x') \approx k(x,x) = k(x',x')$.

The prior $p(\mathbf{f})$ encodes our prior knowledge about the function.



- Model:

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$$
$$y_i | f_i \sim \mathcal{N}(f_i, \sigma^2)$$

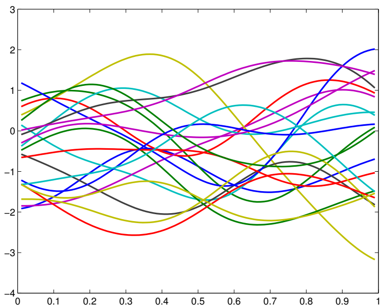
Gaussian Processes

- What does a multivariate normal prior mean?
- Imagine $\mathbf{x}$ forms an infinitesimally dense grid of data space. Simulate prior draws

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$$

Plot f_i vs x_i for $i = 1, \ldots, n$.

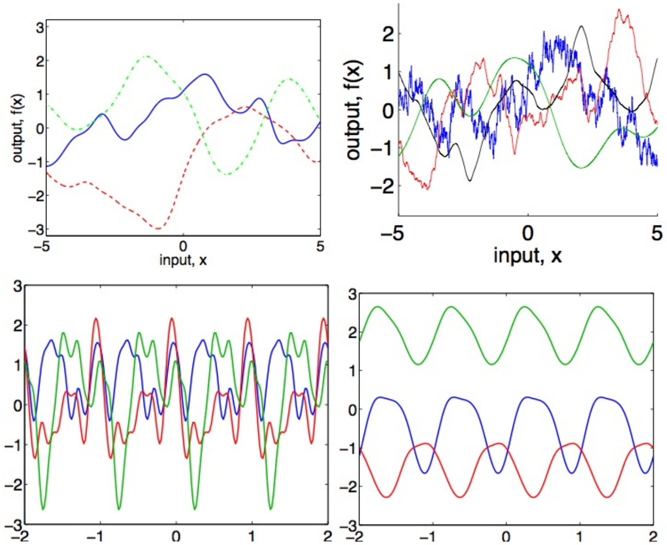
- The corresponding prior over functions is called a **Gaussian Process** (GP): any finite number of evaluations of which follow a Gaussian distribution.



http://www.gaussianprocess.org/

Gaussian Processes

- Different kernels lead to different function characteristics.



Carl Rasmussen. Tutorial on Gaussian Processes at NIPS 2006.

Gaussian Processes

$$\mathbf{f}|\mathbf{x} \sim \mathcal{N}(0, \mathbf{K})$$
$$\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

- Posterior distribution:

$$\mathbf{f}|\mathbf{y} \sim \mathcal{N}(\mathbf{K}(\mathbf{K}+\sigma^2 I)^{-1}\mathbf{y}, \mathbf{K}-\mathbf{K}(\mathbf{K}+\sigma^2 I)^{-1}\mathbf{K})$$

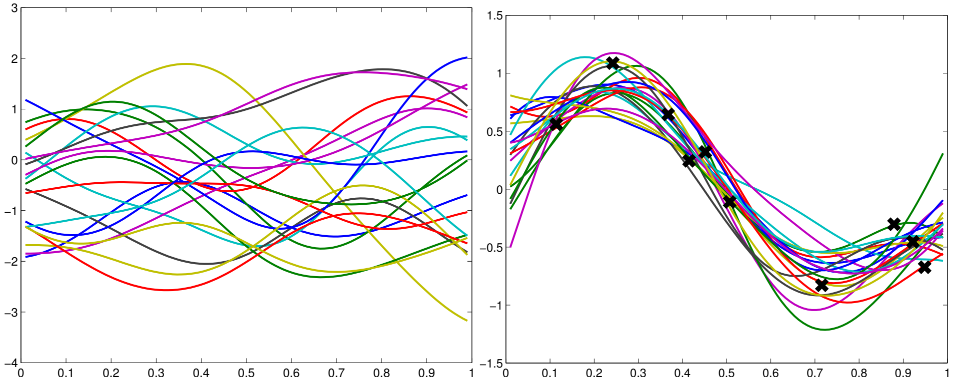
- Posterior predictive distribution: Suppose $\mathbf{x}'$ is a test set. We can extend our model to include the function values $\mathbf{f}'$ at the test set:

$$\begin{pmatrix}\mathbf{f}\\ \mathbf{f}'\end{pmatrix}|\mathbf{x},\mathbf{x}' \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\mathbf{K_{xx}} & \mathbf{K_{xx'}}\\ \mathbf{K_{x'x}} & \mathbf{K_{x'x'}}\end{pmatrix}\right)$$
$$\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

  where $\mathbf{K_{xx'}}$ is matrix with (i,j)-th entry $k(x_i, x'_j)$.
- Some manipulation of multivariate normals gives:

$$\mathbf{f}'|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K_{x'x}}(\mathbf{K_{xx}}+\sigma^2 I)^{-1}\mathbf{y}, \mathbf{K_{x'x'}}-\mathbf{K_{x'x}}(\mathbf{K_{xx}}+\sigma^2 I)^{-1}\mathbf{K_{xx'}}\right)$$

Gaussian Processes



GP regression demo: http://www.tmpl.fi/gp/

- A whirlwind journey through data mining and machine learning techniques:
  - **Unsupervised learning**: PCA, MDS, Isomap, Hierarchical clustering, K-means, mixture modelling, EM algorithm.
  - **Supervised learning**: LDA, QDA, naïve Bayes, logistic regression, SVMs, kernel methods, kNN, deep neural networks, Gaussian processes, decision trees, ensemble methods: random forests, bagging, stacking, dropout and boosting.
  - **Conceptual frameworks**: prediction, performance evaluation, generalization, overfitting, regularization, model complexity, validation and cross-validation, bias-variance tradeoff.
  - **Theory**: decision theory, statistical learning theory, convex optimization, Bayesian vs. frequentist learning, parametric vs non-parametric learning.
- **Further resources**:
  - Machine Learning Summer Schools, videolectures.net.
  - Conferences: NIPS, ICML, UAI, AISTATS.
  - Mailing list: ml-news.

Thank You!