# Lecture 8:
# Gaussian processes

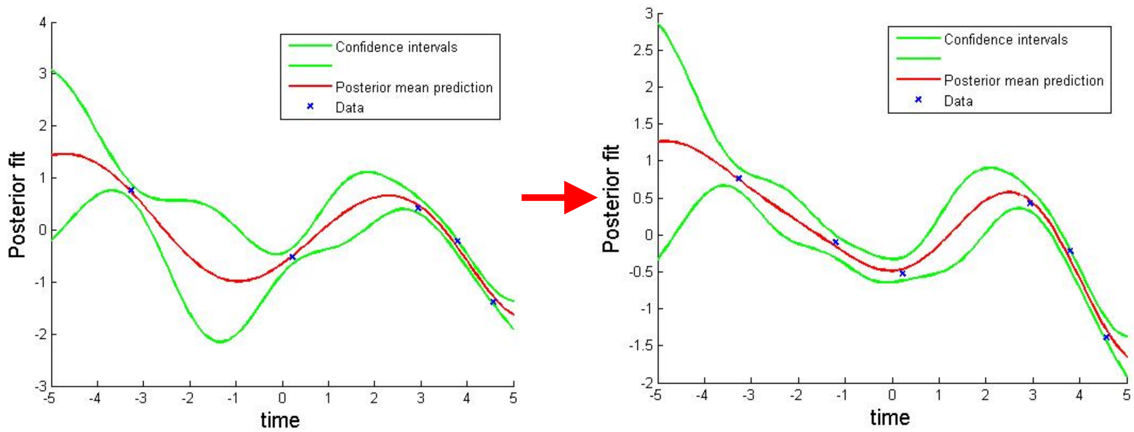
Nando de Freitas
*www.cs.ubc.ca/~nando/340-2008/*
*September 2008*

# Outline

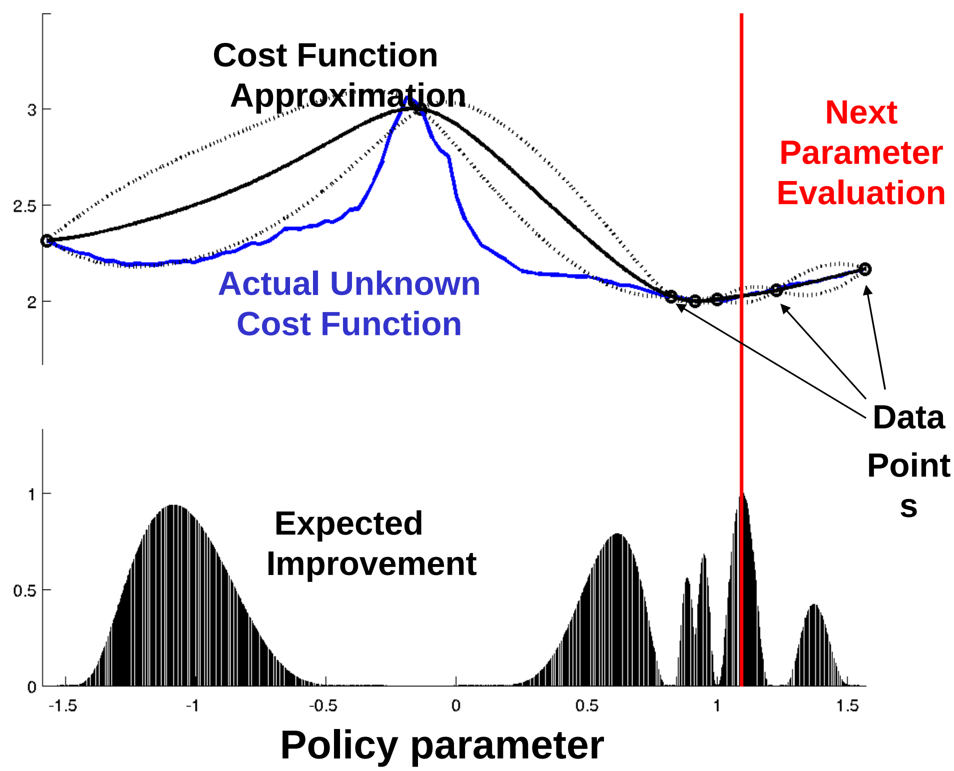
In this lecture we will learn to combine ideas of Bayesian learning, maximum likelihood and decision theory to deal with complex nonlinear problems.

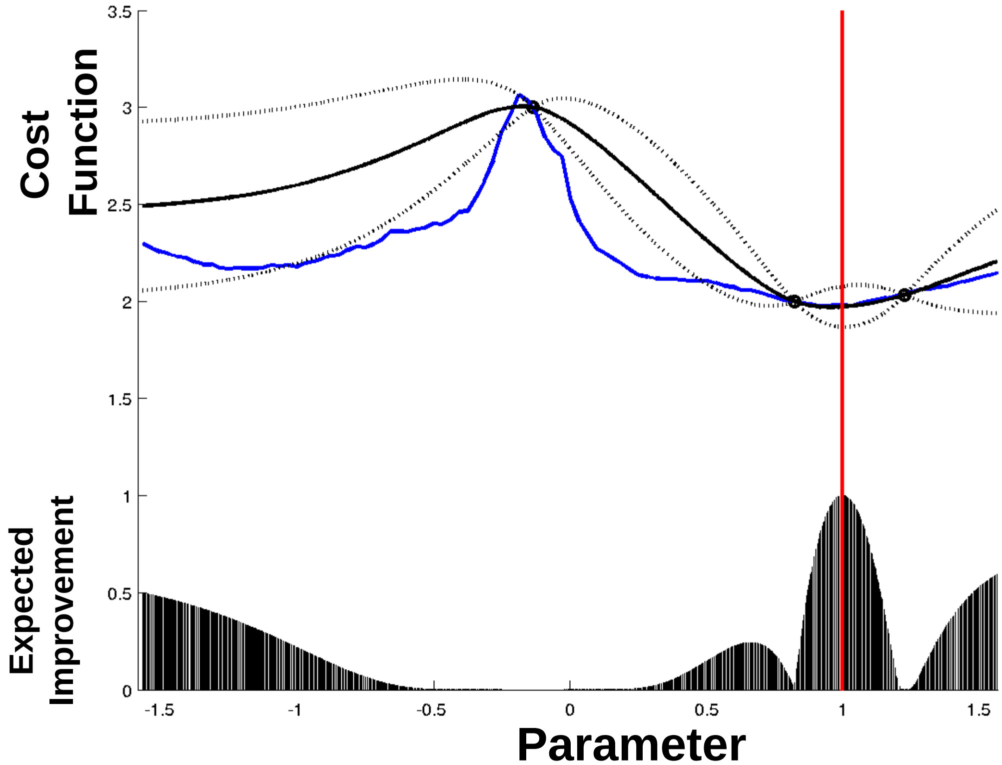
The lecture uses material from the paper: "*Gaussian processes in machine learning*" by Carl Rasmussen.

# Nonlinear regression and active learning



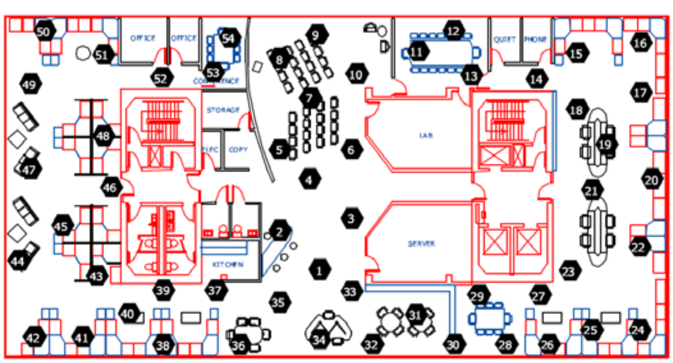


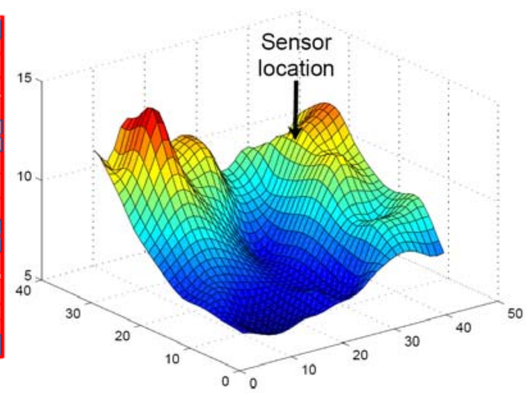








# Sensor placement application by Andreas Krause and Carlos Guestrin





# Gaussian processes prior over functions

**Definition 1.** *A Gaussian Process is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions.*

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

For example:

$$m(x) = \tfrac{1}{4}x^2, \text{ and } k(x, x') = \exp(-\tfrac{1}{2}(x - x')^2)$$

Now, given *n* input data points, we have: $\mathbf{f} \sim \mathcal{N}(\mu, \Sigma)$

$$\mu_i = m(x_i) = \tfrac{1}{4}x_i^2, \quad i = 1, \ldots, n \text{ and}$$
$$\Sigma_{ij} = k(x_i, x_j) = \exp(-\tfrac{1}{2}(x_i - x_j)^2), \quad i, j = 1, \ldots, n,$$



# Posterior distribution

$\mathbf{f}$ = known values of the function in the training set

$\mathbf{f}_*$ = unknown values of the function in the test set $X_*$

Jointly, they come from a multivariate Gaussian:

$$\begin{bmatrix}\mathbf{f} \\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu} \\ \boldsymbol{\mu}_*\end{bmatrix}, \begin{bmatrix}\Sigma & \Sigma_* \\ \Sigma_*^\top & \Sigma_{**}\end{bmatrix}\right)$$

where $\boldsymbol{\mu} = m(x_i), i = 1, \ldots, n$

The posterior can be obtained analytically (see homework 5):

$$\mathbf{f}_*|\mathbf{f} \sim \mathcal{N}\big(\mu_* + \Sigma_*^\top \Sigma^{-1}(\mathbf{f} - \boldsymbol{\mu}),\ \Sigma_{**} - \Sigma_*^\top \Sigma^{-1}\Sigma_*\big)$$

# Noise in the function obervations: Gaussian likelihood

$$y(x) = f(x) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma_n^2),$$

$$f \sim \mathcal{GP}(m, k), \qquad y \sim \mathcal{GP}(m, k + \sigma_n^2 \delta_{ii'})$$

$\delta_{ii'} = 1$ iff $i = i'$ is the Kronecker's delta



# Parameter learning for GPs: Can do it by maximum likelihood

$$\begin{aligned} L &= \log p(\mathbf{y}|\mathbf{x}, \theta) \\ &= -\tfrac{1}{2}\log|\Sigma| - \tfrac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu}) - \tfrac{n}{2}\log(2\pi) \end{aligned}$$

For example, we can parameterize the mean and covariance:

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

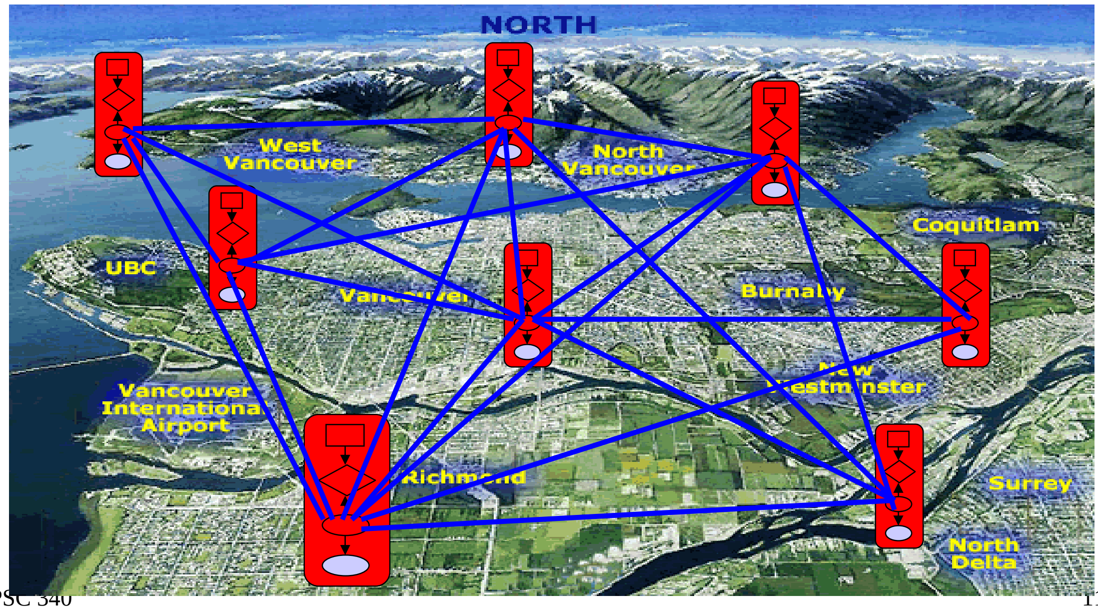
$$m(x) = ax^2 + bx + c$$

$$k(x, x') = \sigma_y^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right) + \sigma_n^2 \delta_{ii'}$$

$$\theta = \{a, b, c, \sigma_y, \sigma_n, \ell\}$$

# Sensor network scheduling

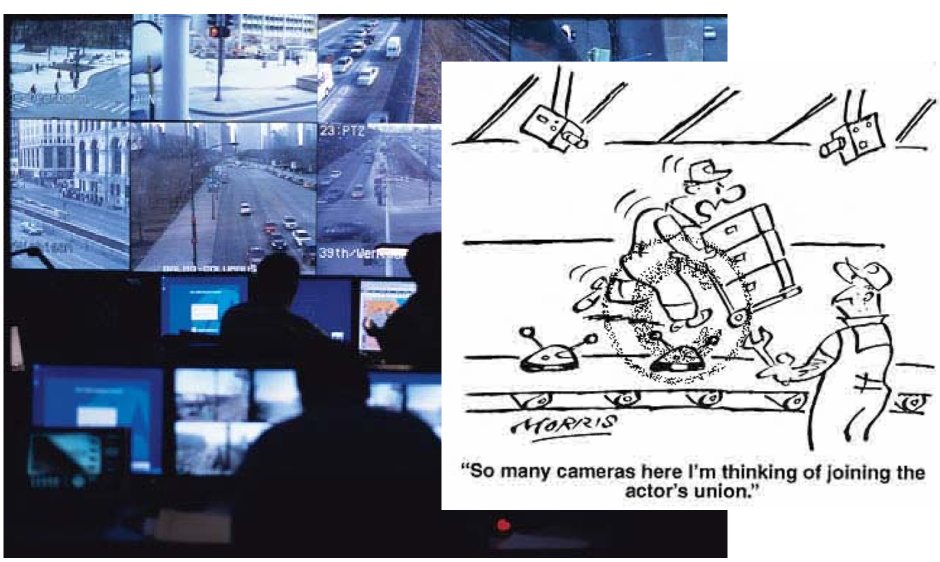
Automatically schedule sensors to obtain the best understanding of the environment while minimizing resource expenditure (power, bandwidth, need for human intervention)?



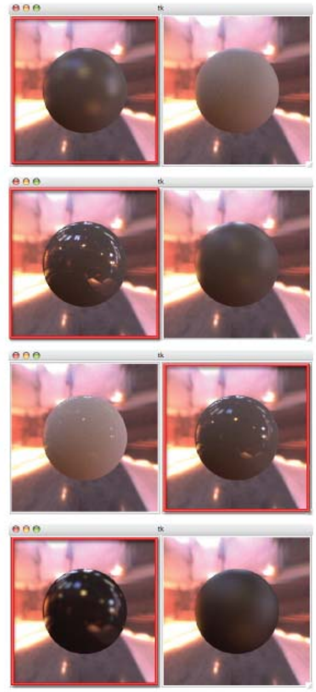



# Active learning and surveillance

**In network with thousands of cameras, which camera views should be presented to the human operator?**

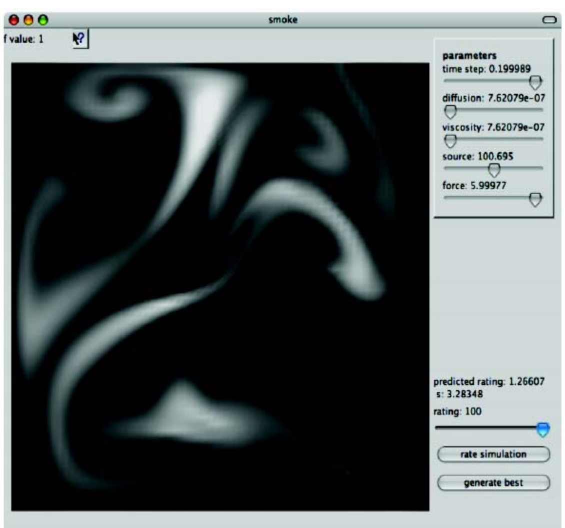


# Intelligent user interfaces





# Other Active Learning Problems




- Which sites should a crawler visit?
- Which tests to conduct in active diagnosis?
- What is a good animated walk?
- Interactive video search.
- Relevance feedback systems.
- Optimizing spatial and temporal allocation of sensors. How do we adapt to target maneuvers?
- Learning opponent's strategies.

# Active computer vision