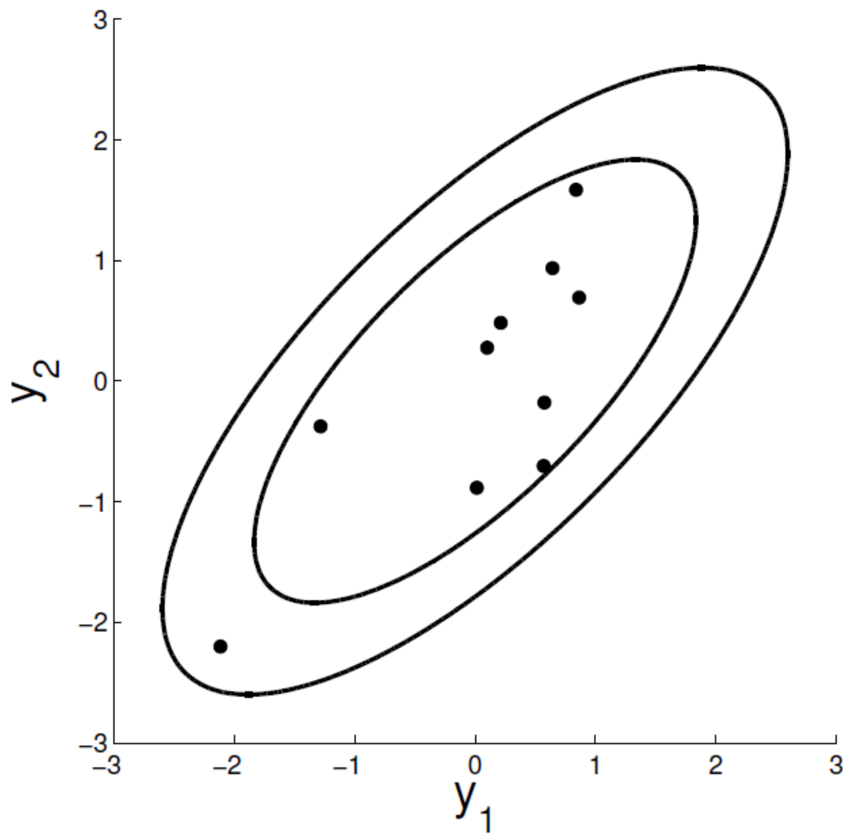
Gaussian Process

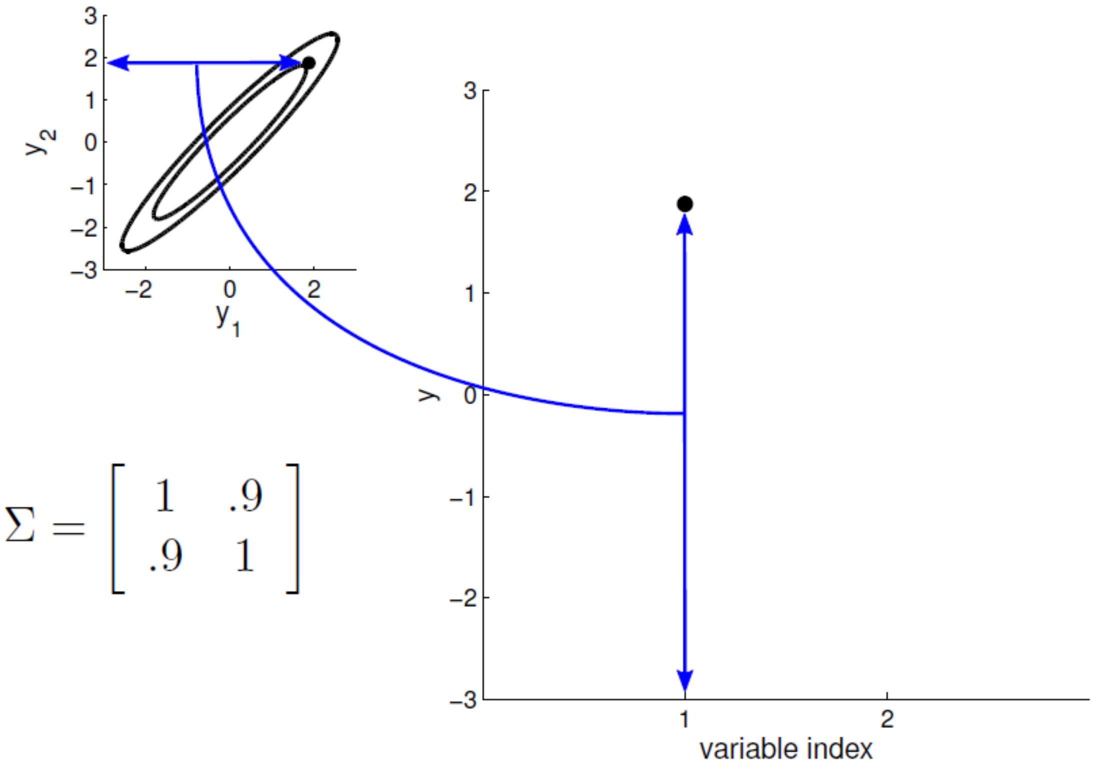
Multivariate Gaussian

$$p(\mathbf{y}|\Sigma) \propto \exp\left(-\frac{1}{2}\mathbf{y}^\top\Sigma^{-1}\mathbf{y}\right)$$

$$\Sigma = \begin{bmatrix} 1 & .7 \\ .7 & 1 \end{bmatrix}$$

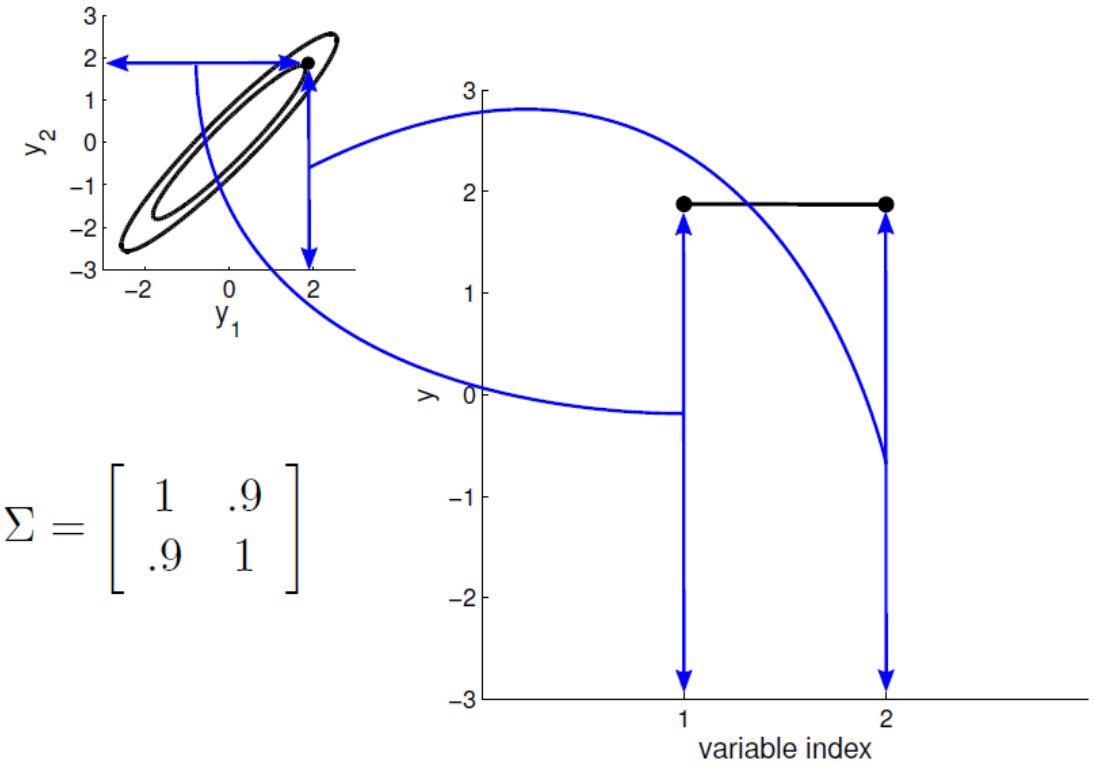


Alternate Visualization



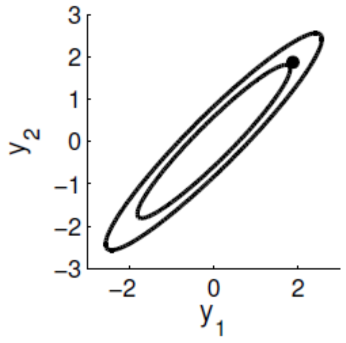
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

Alternate Visualization

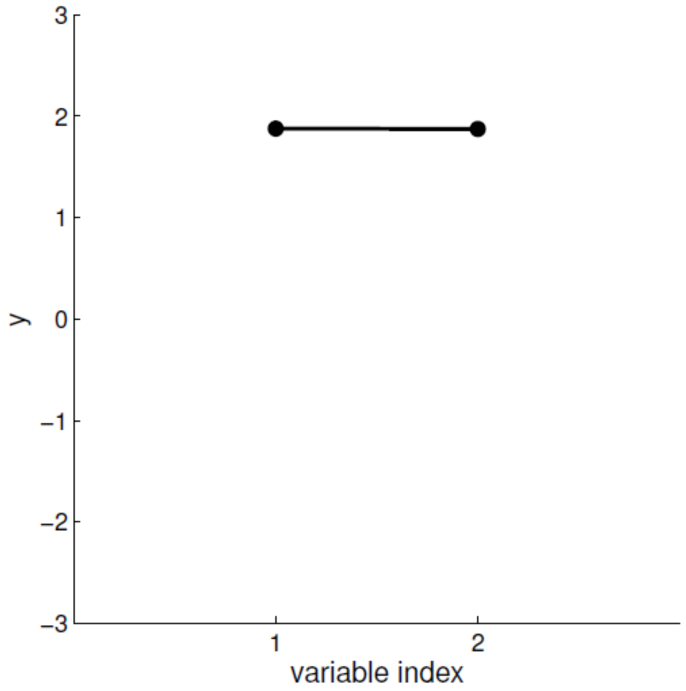


$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

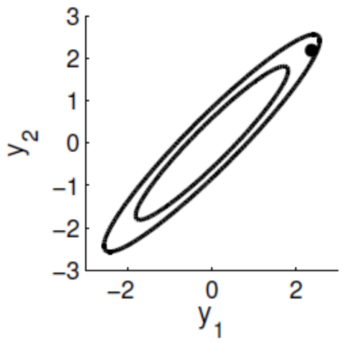
Alternate Visualization



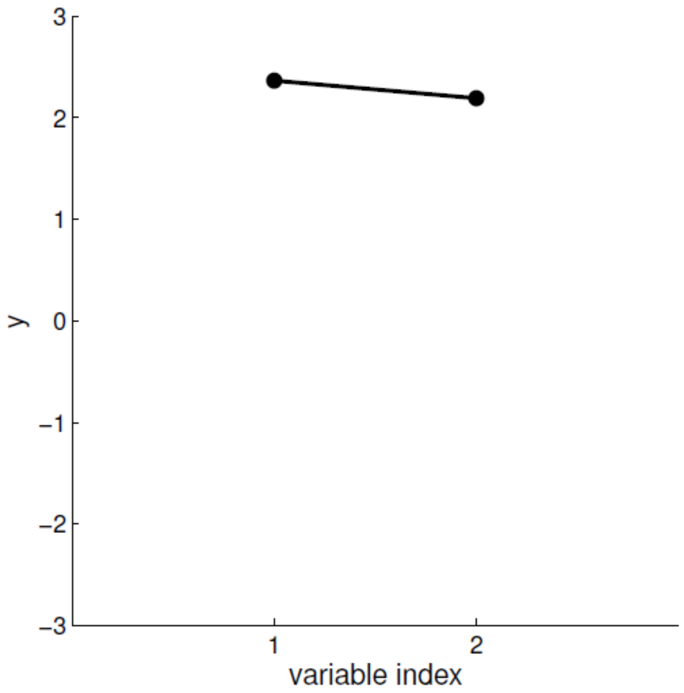
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



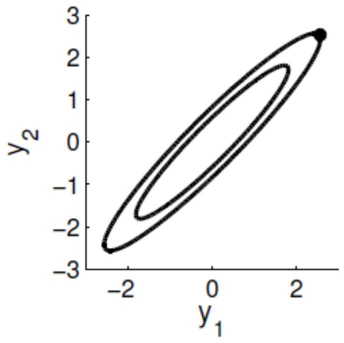
Alternate Visualization



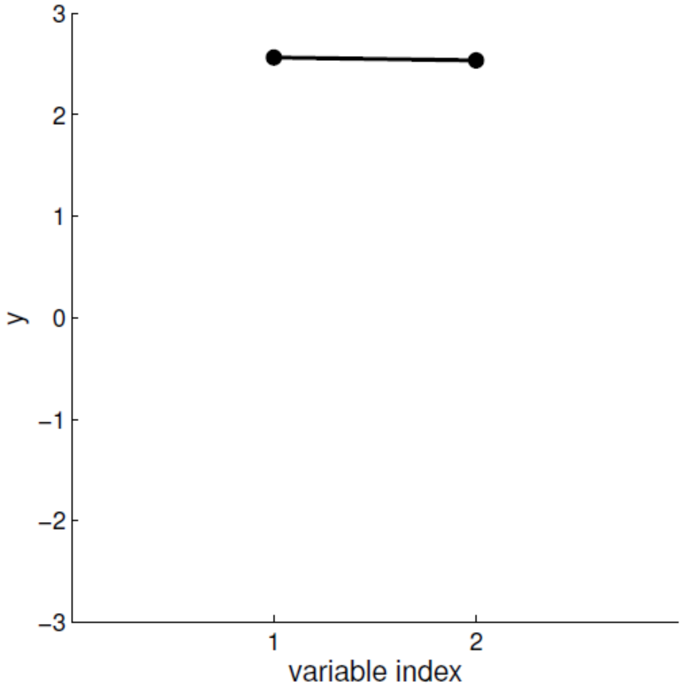
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



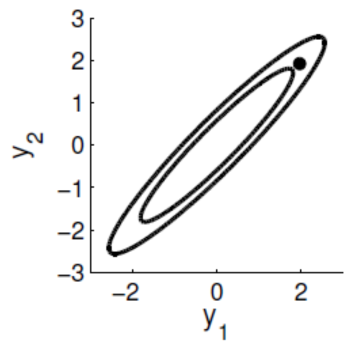
Alternate Visualization



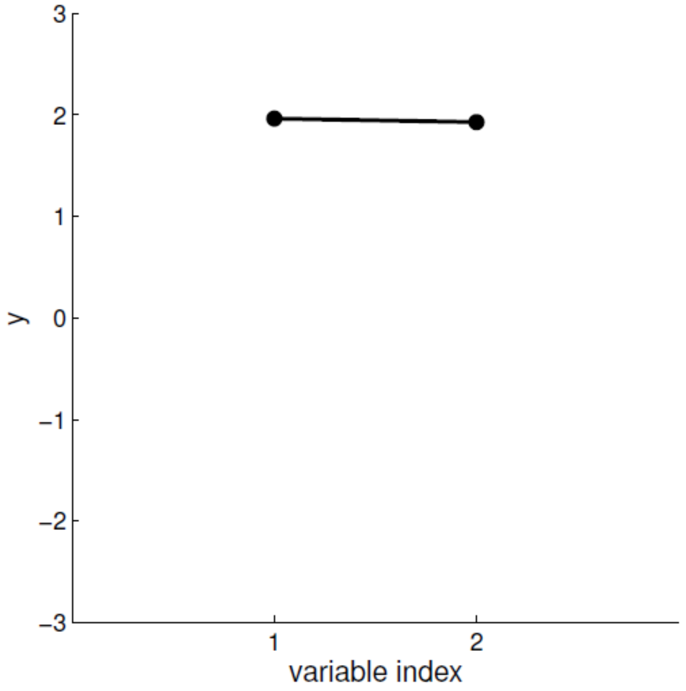
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



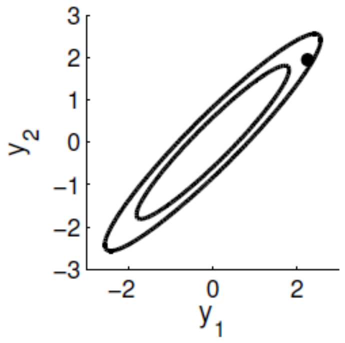
Alternate Visualization



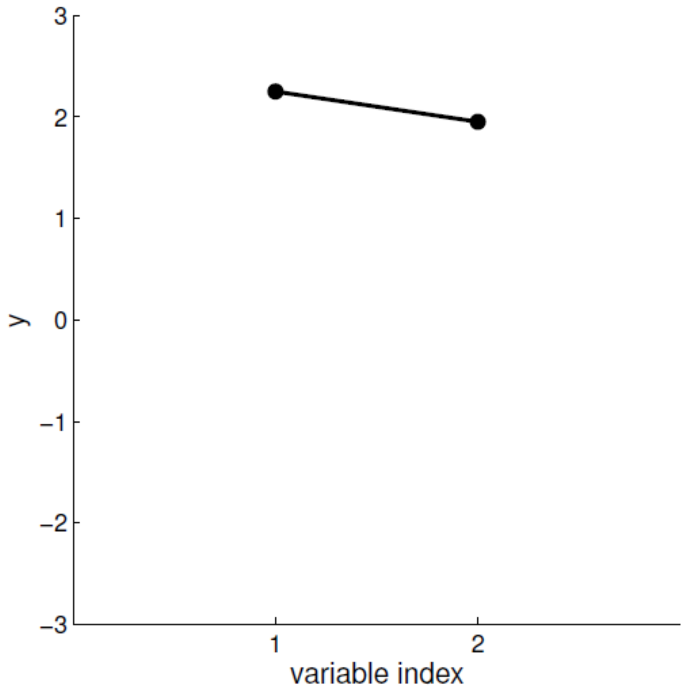
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



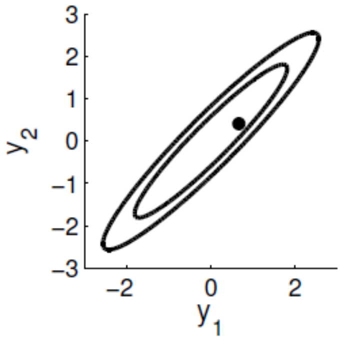
Alternate Visualization



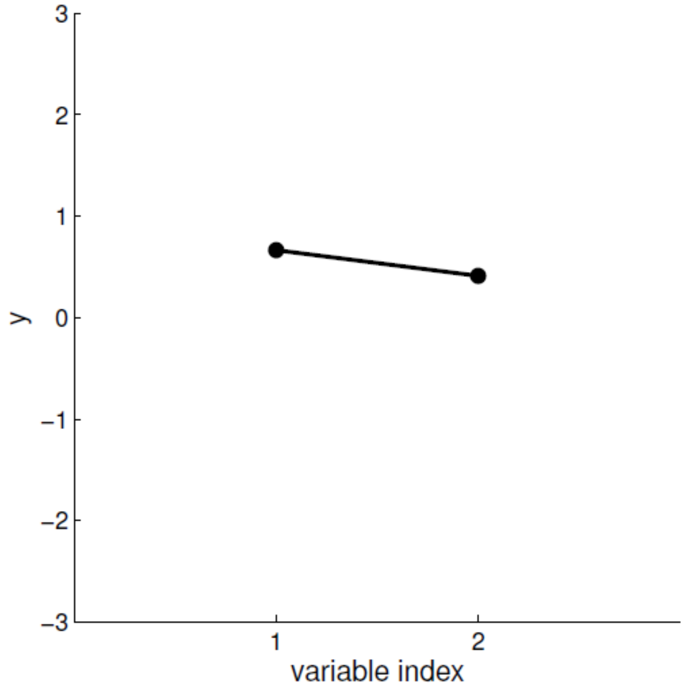
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



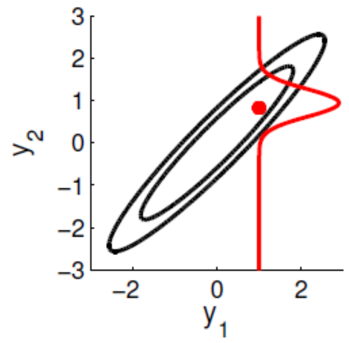
Alternate Visualization



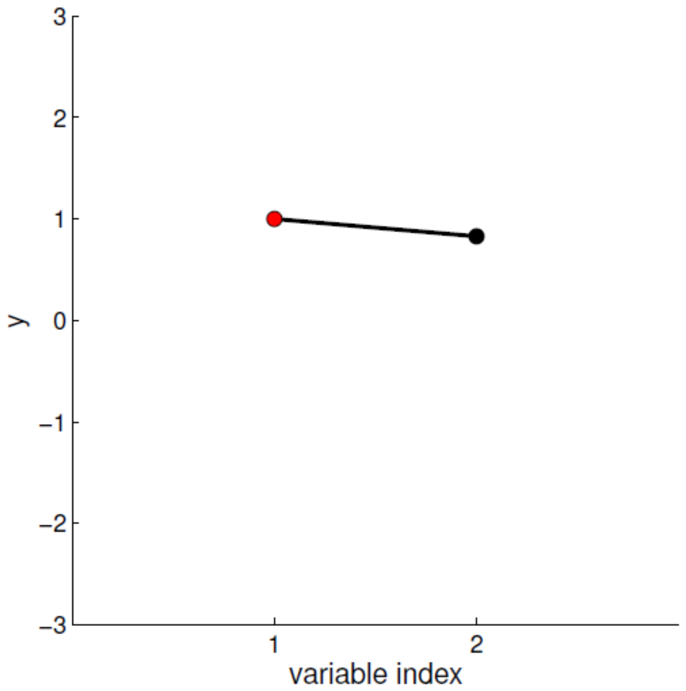
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



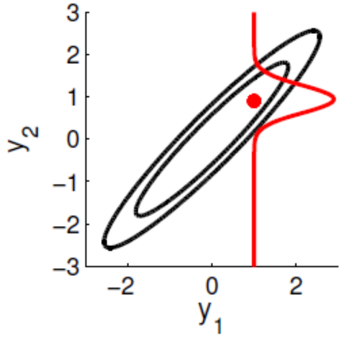
Alternate Visualization



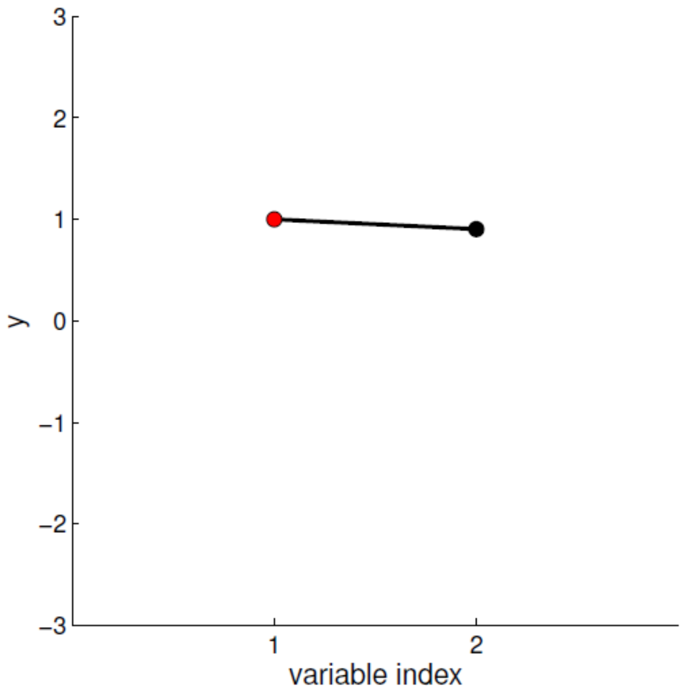
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



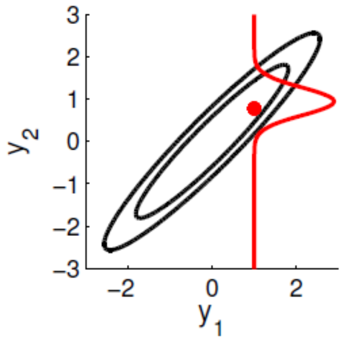
Alternate Visualization



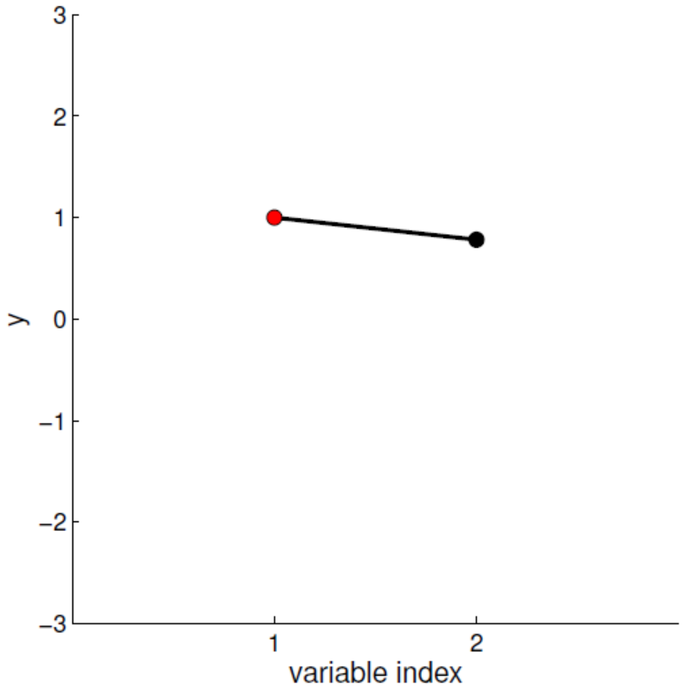
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



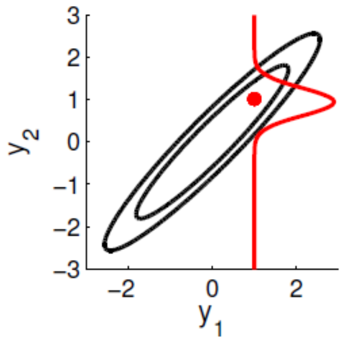
Alternate Visualization



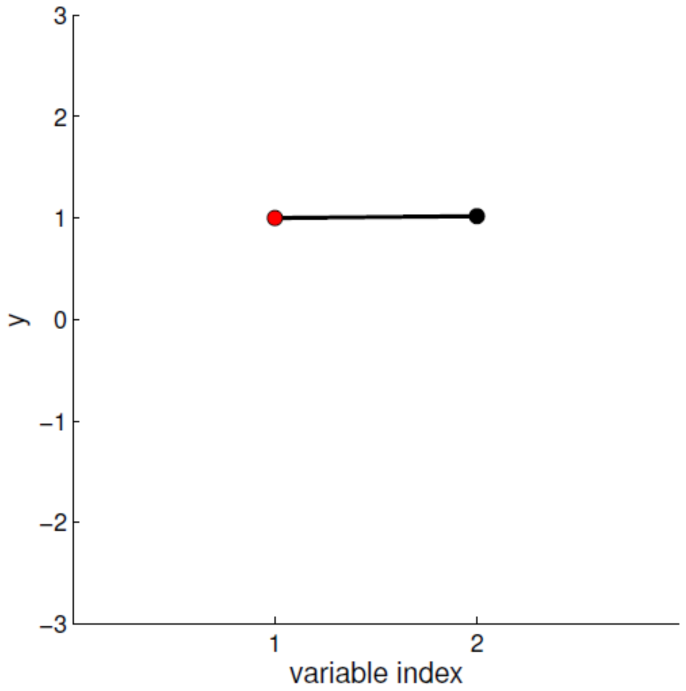
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



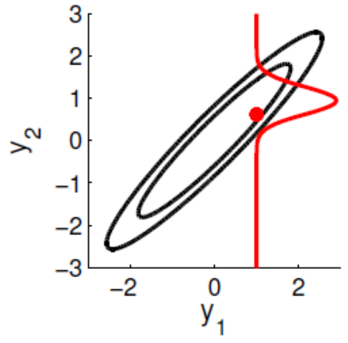
Alternate Visualization



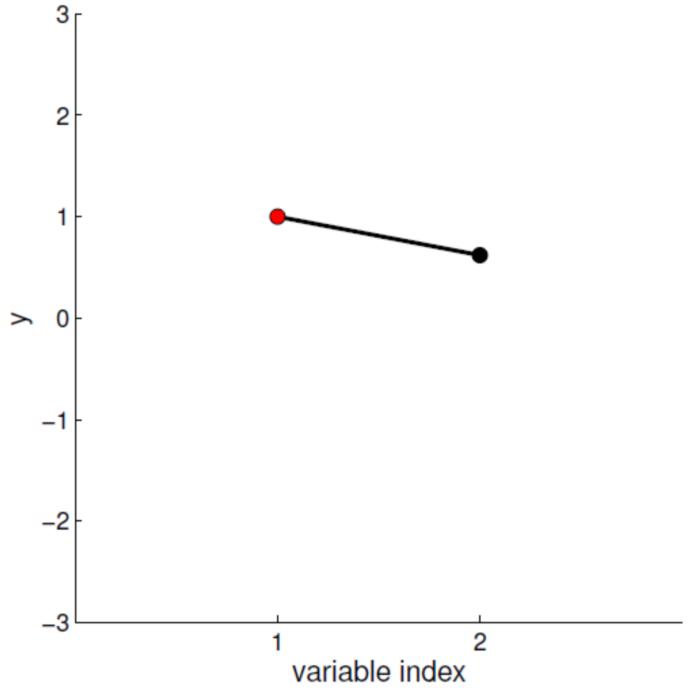
$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



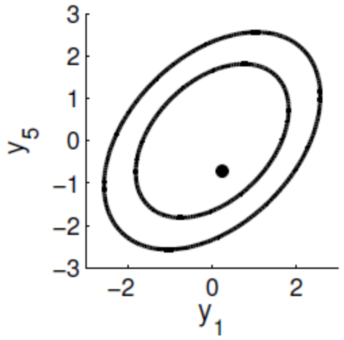
Alternate Visualization

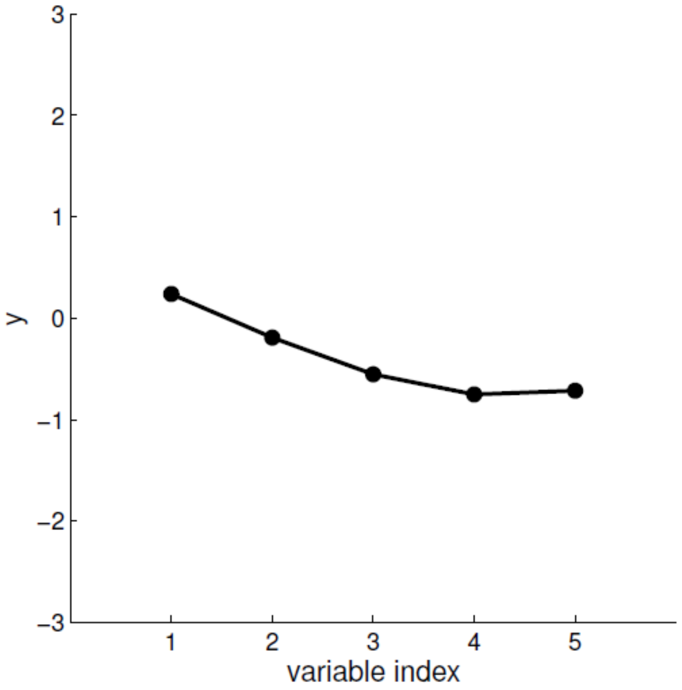


$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$



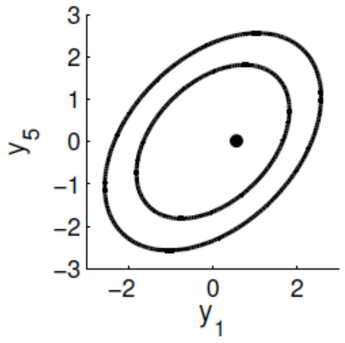
Alternate Visualization

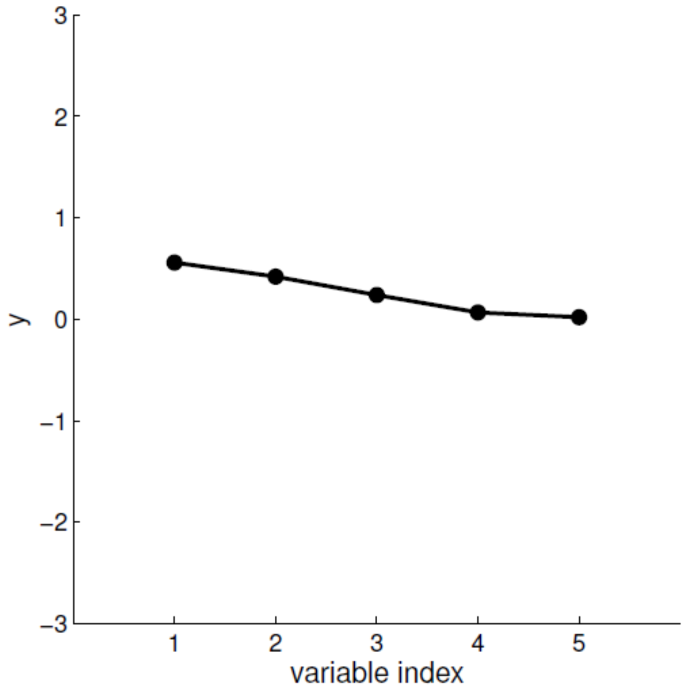




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

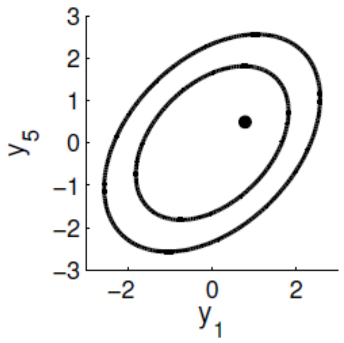
Alternate Visualization

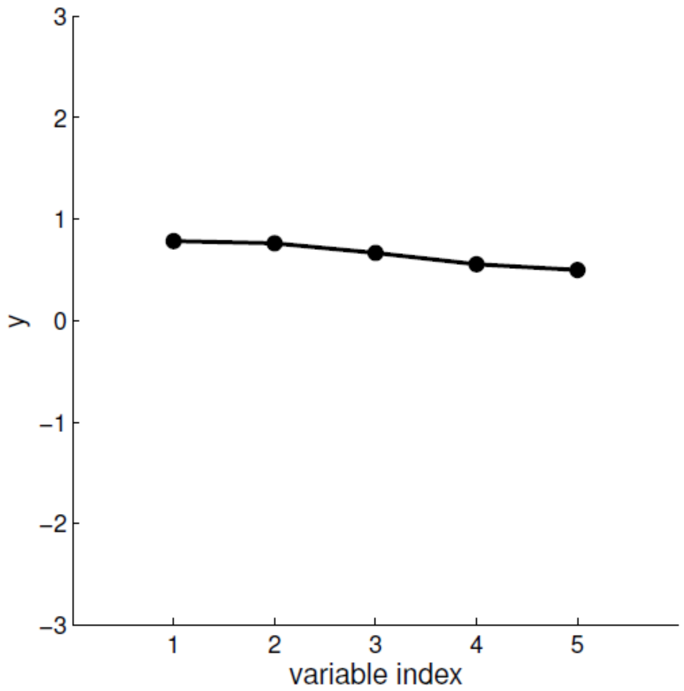




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

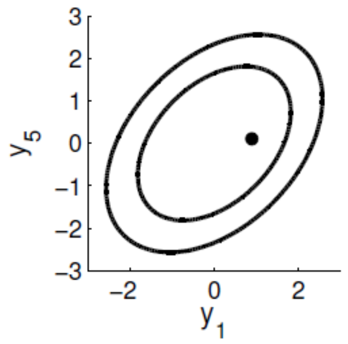
Alternate Visualization

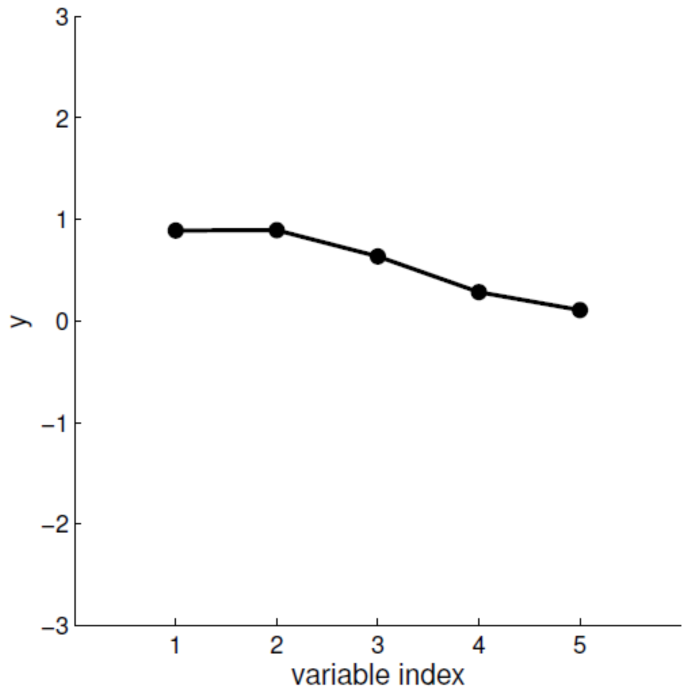




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

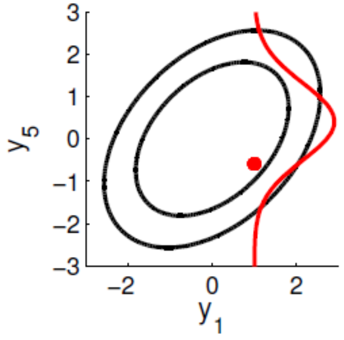
Alternate Visualization

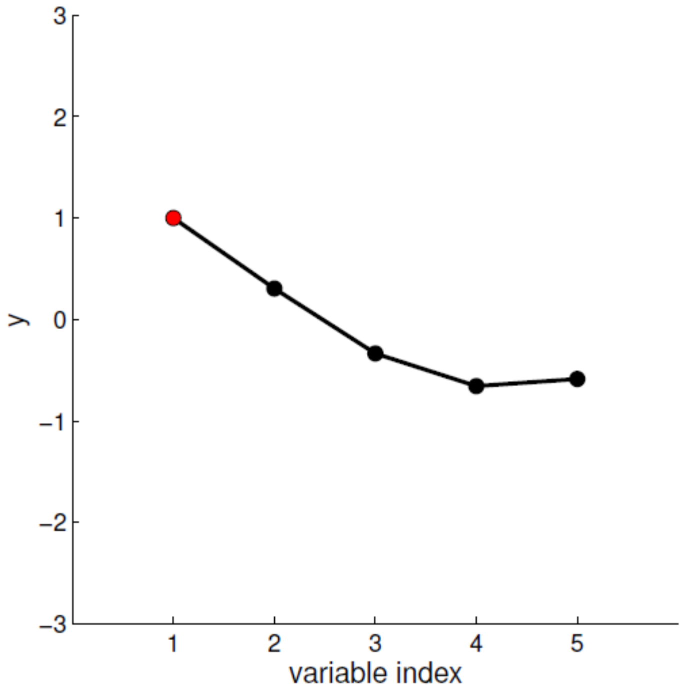




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

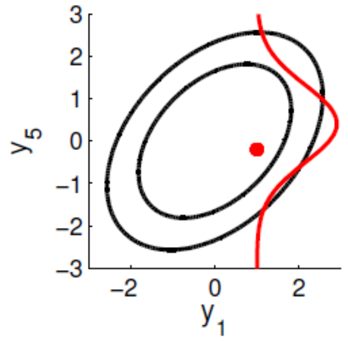
Alternate Visualization



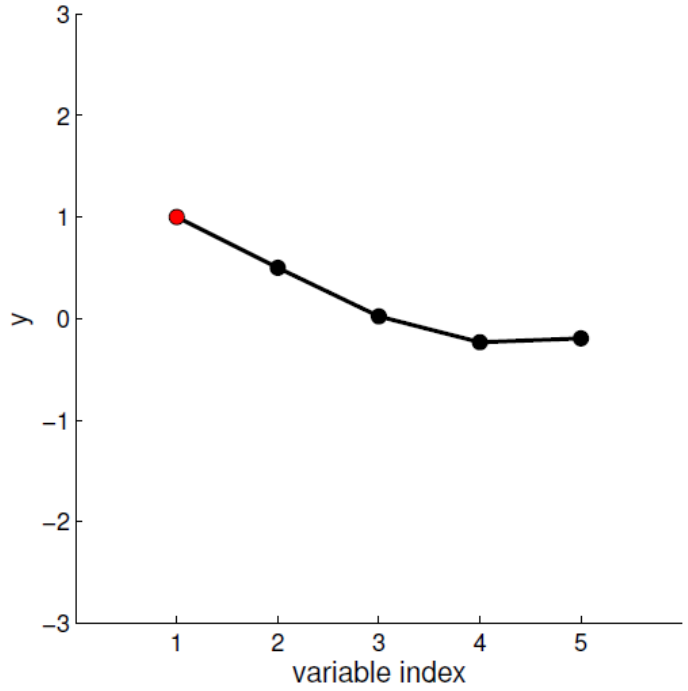


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

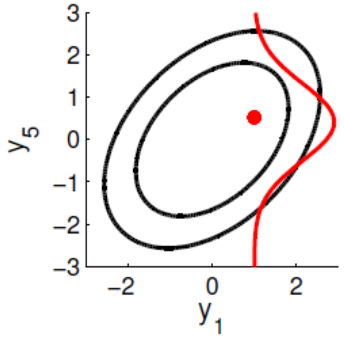
Alternate Visualization

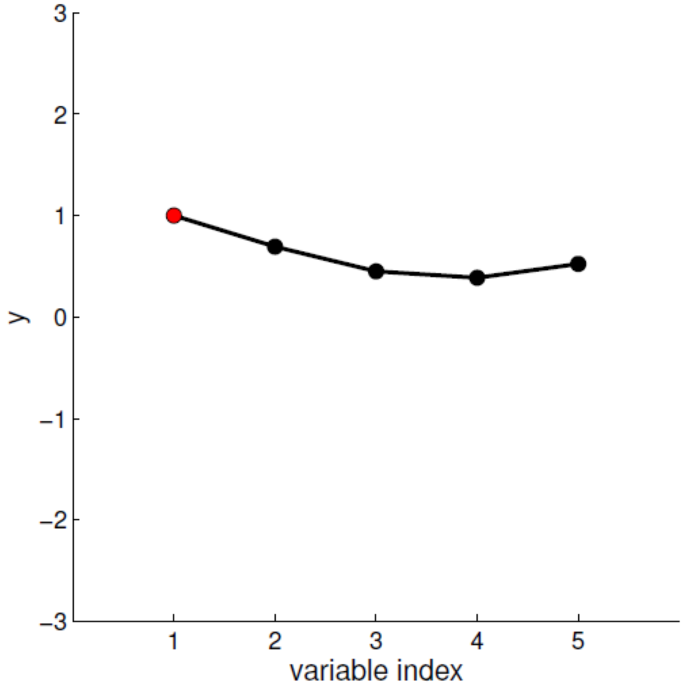


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$



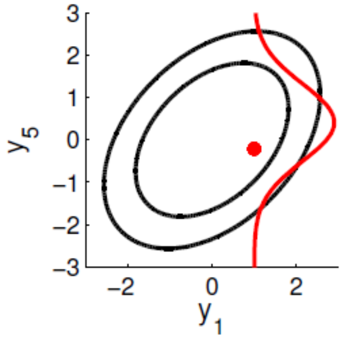
Alternate Visualization

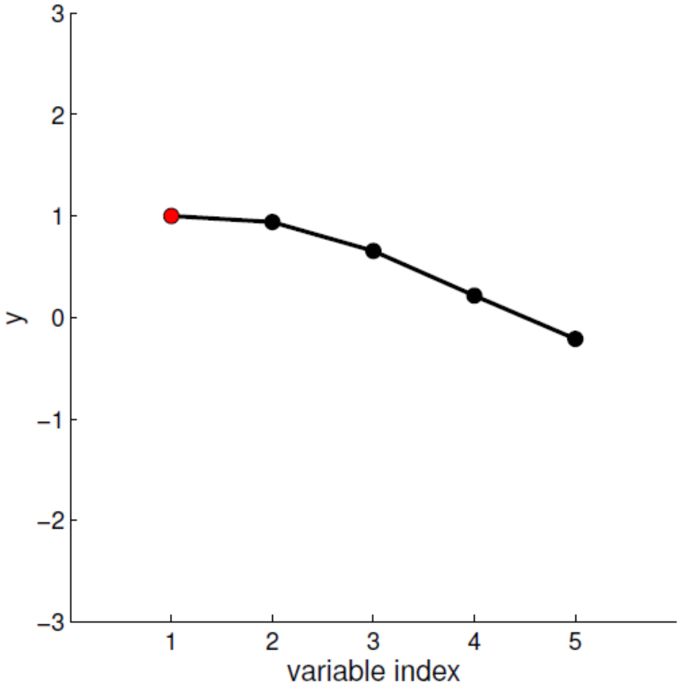




$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

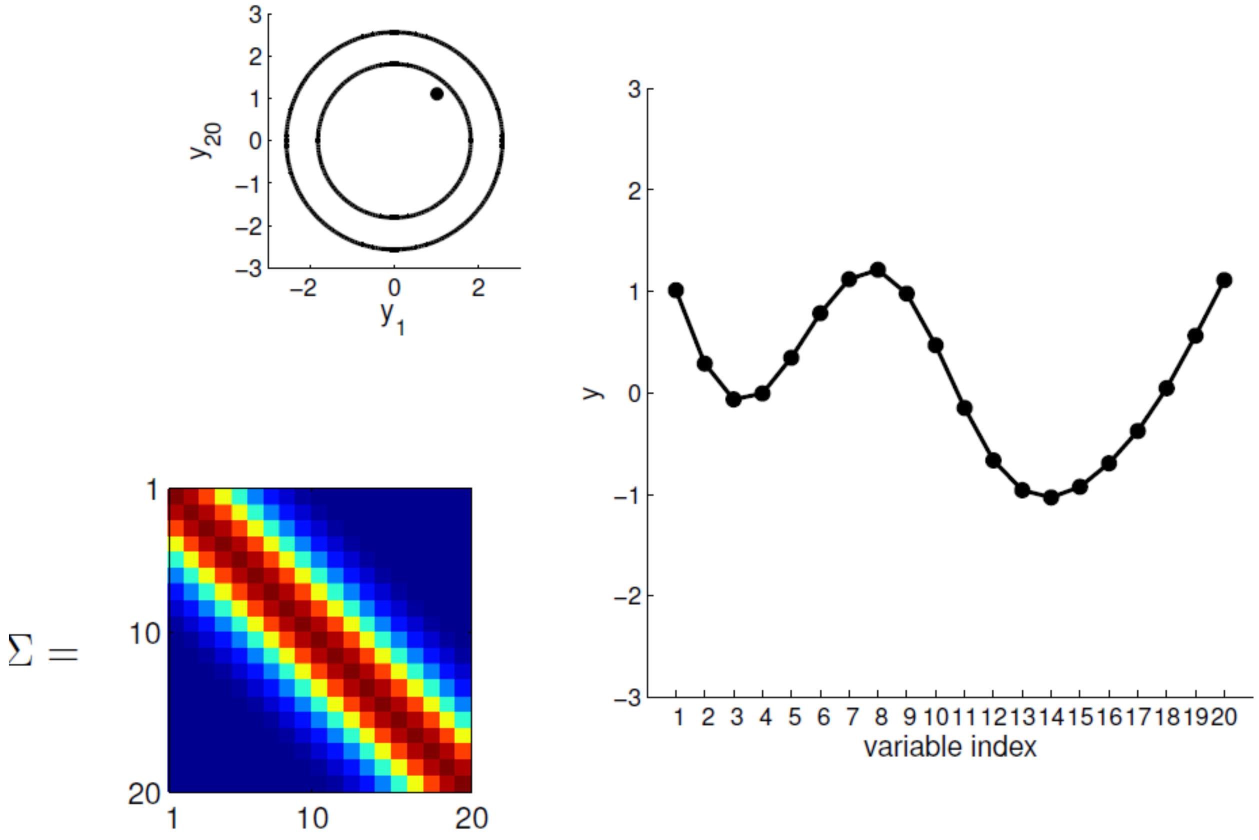
Alternate Visualization



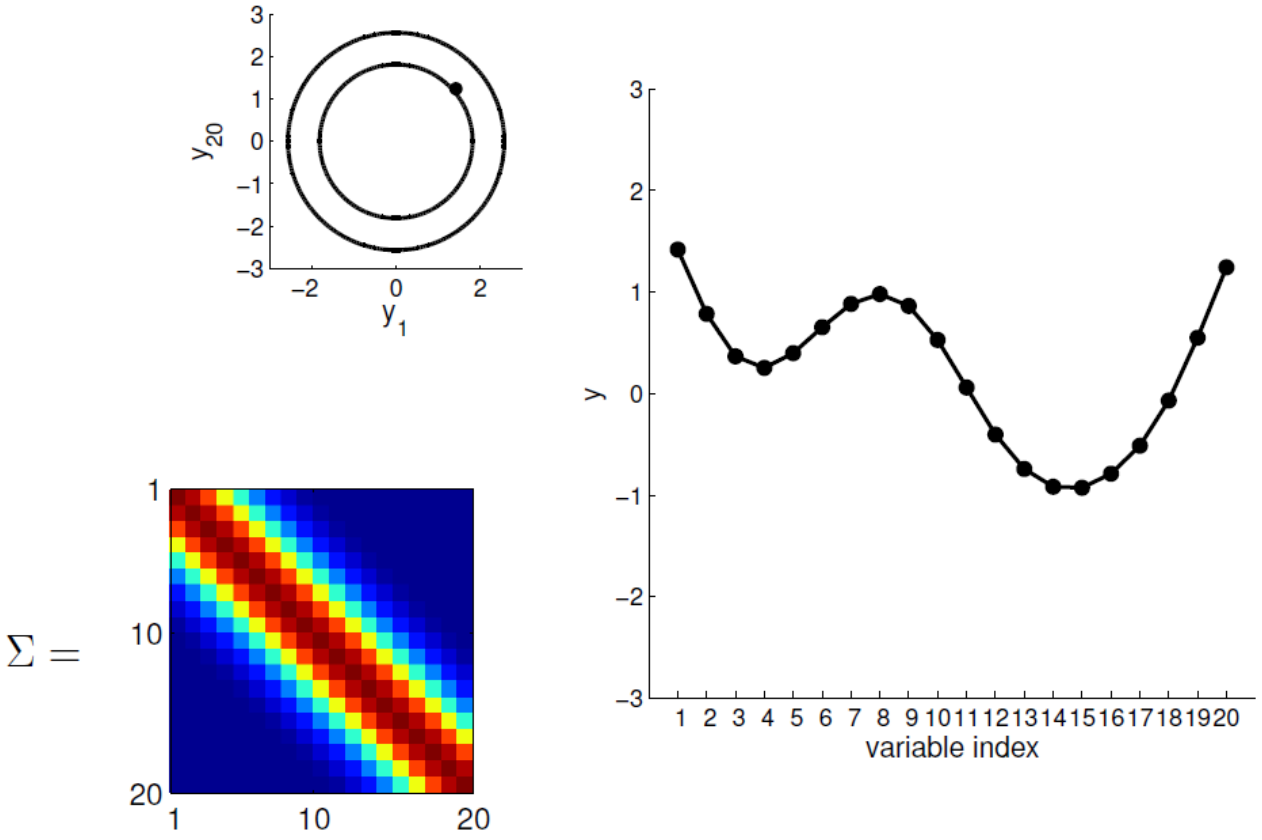


$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$

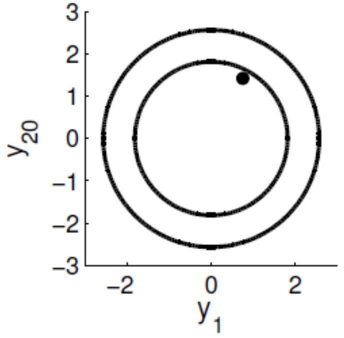
Alternate Visualization



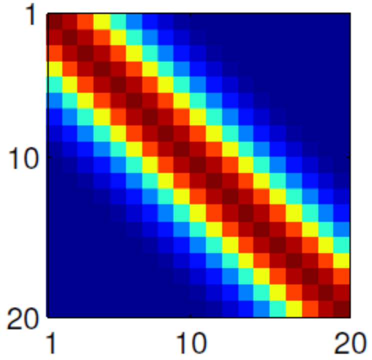
Alternate Visualization

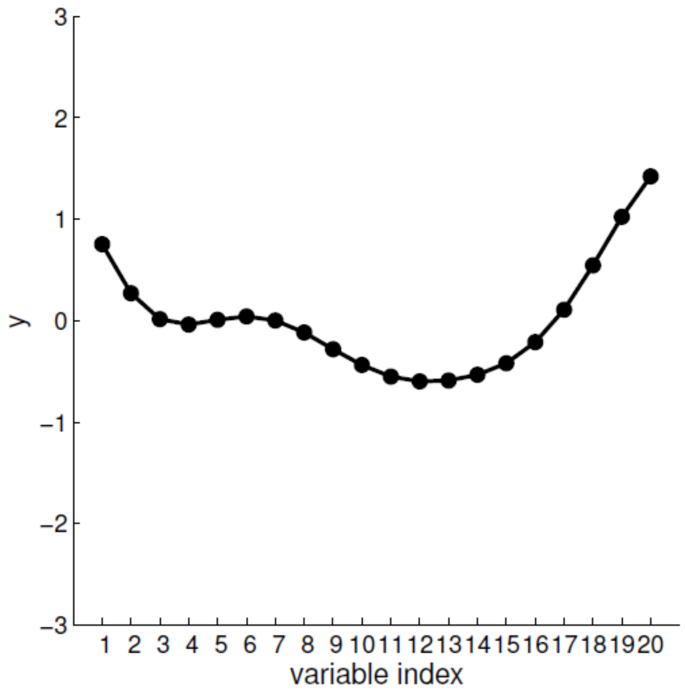


Alternate Visualization

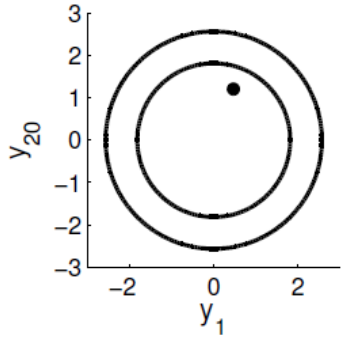


$\Sigma =$

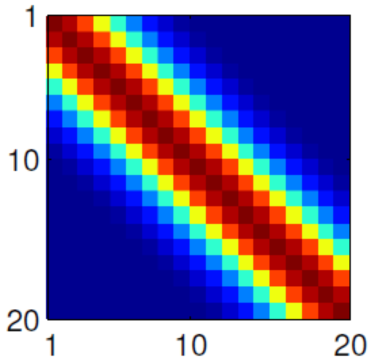


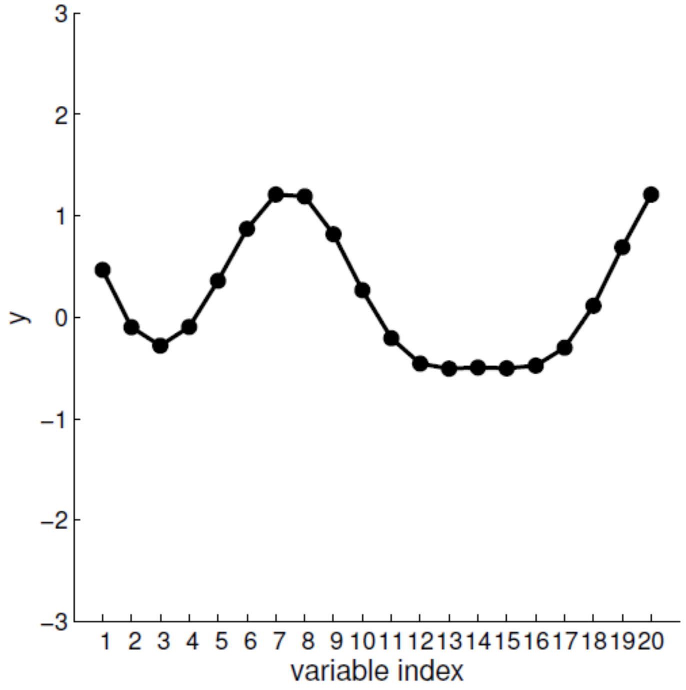


Alternate Visualization

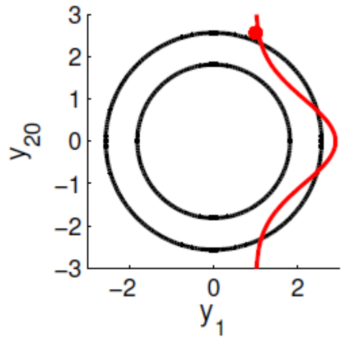


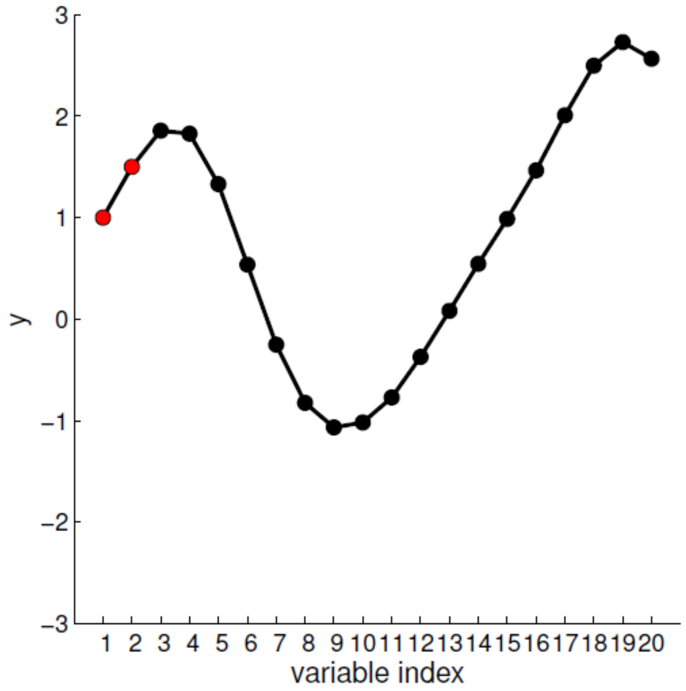
$\Sigma =$



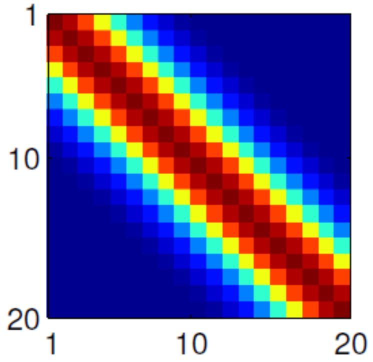


Alternate Visualization



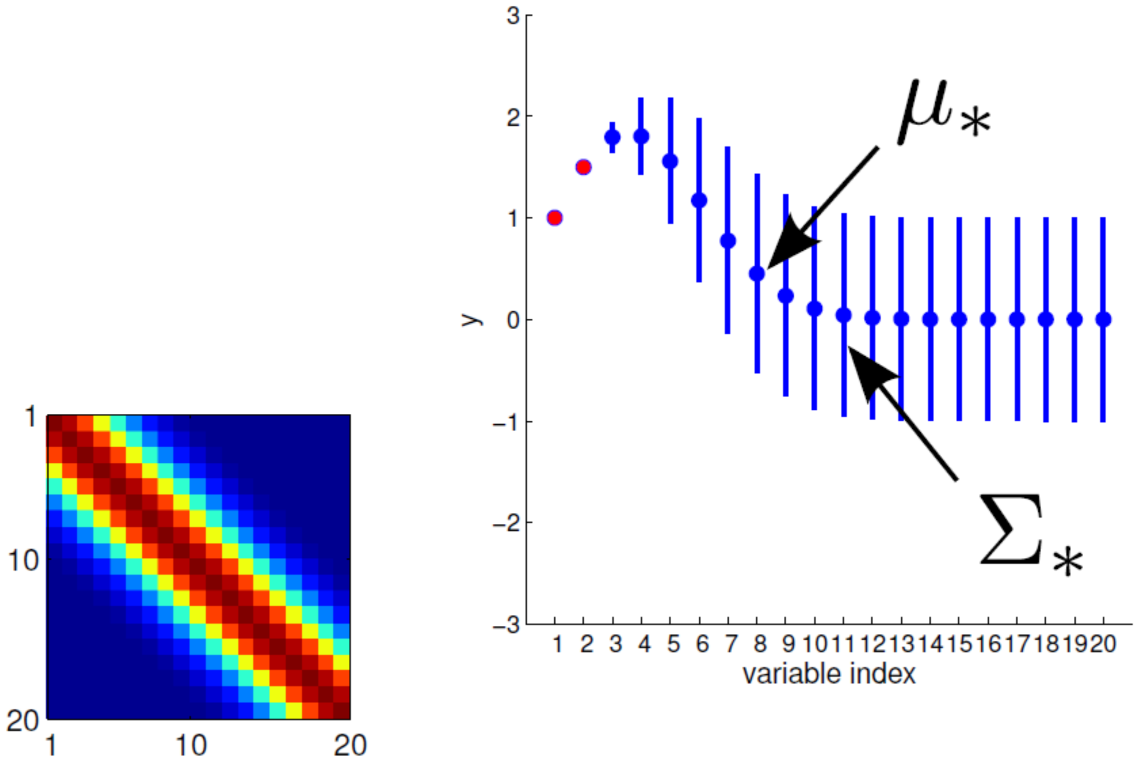


$\Sigma =$

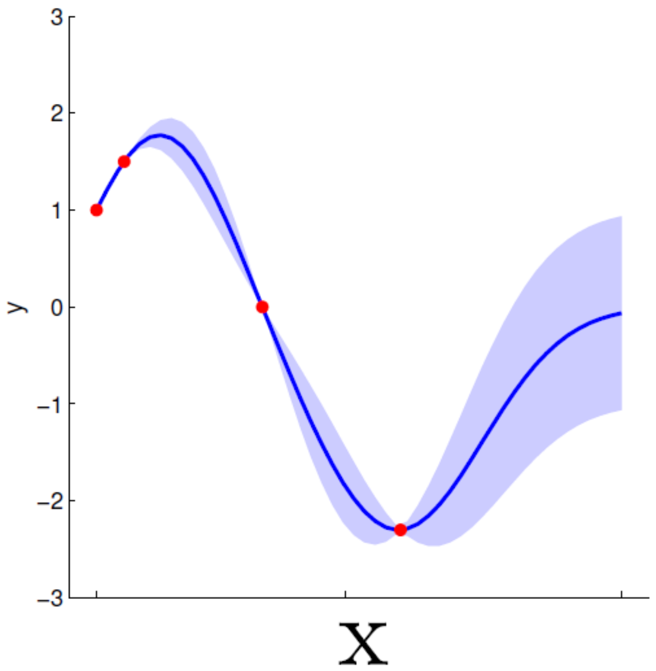


Alternate Visualization

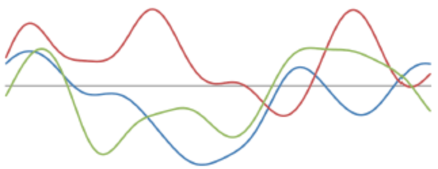
$\Sigma =$



Distribution over Function Space: Stochastic Process



- Draw from a $\mathcal{GP}(\mu, \kappa)$ will give us a random function f (imagine it as an infinite dim. vector)



Definition: Gaussian Process

Gaussian process = generalisation of multivariate Gaussian distribution to infinitely many variables.

> **Definition**: a Gaussian process is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.

Stochastic Process:
A collection of random variables with an associated index x

A Gaussian distribution is fully specified by a mean vector, $\boldsymbol{\mu}$, and covariance matrix Σ:

$$\mathbf{f} = (\mathrm{f}_1, \ldots, \mathrm{f}_n) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma), \quad \text{indices} \quad i = 1, \ldots, n$$

A Gaussian process is fully specified by a mean function $m(\mathbf{x})$ and covariance function $\mathrm{K}(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), \mathrm{K}(\mathbf{x}, \mathbf{x}')\right), \quad \text{indices} \quad \mathbf{x}$$

Definition: Gaussian Process

- f is said to be drawn from a $\mathcal{GP}(\mu, \kappa)$ if its finite dim. version is the following joint Gaussian

$$\begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{x}_1) \\ \mu(\boldsymbol{x}_2) \\ \vdots \\ \mu(\boldsymbol{x}_N) \end{bmatrix}, \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) \dots \kappa(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \kappa(\boldsymbol{x}_2, \boldsymbol{x}_1) \dots \kappa(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots \quad \ddots \quad \vdots \\ \kappa(\boldsymbol{x}_N, \boldsymbol{x}_1) \dots \kappa(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \right)$$

- The above means that f's values at any finite set of inputs are jointly Gaussian
- We can also write the above more compactly as $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ where

$$\mathbf{f} = \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_N) \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \mu(\boldsymbol{x}_1) \\ \mu(\boldsymbol{x}_2) \\ \vdots \\ \mu(\boldsymbol{x}_N) \end{bmatrix}, \mathbf{K} = \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) \dots \kappa(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \kappa(\boldsymbol{x}_2, \boldsymbol{x}_1) \dots \kappa(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots \quad \ddots \quad \vdots \\ \kappa(\boldsymbol{x}_N, \boldsymbol{x}_1) \dots \kappa(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

Advantageous Properties of Gaussians

- Sum of Gaussians is a Gaussian
- Product of Gaussians is a Gaussian
- Scaled Gaussian is a Gaussian

- If P(A, B) is Gaussian –
  - P(A), P(B) marginals are Gaussians
  - P(A|B) conditionals are Gaussian

Kernel Function: Example

Example kernel (squared exponential or SE):

$$k(t_i, t_j) = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\}$$

From kernel to covariance matrix

- Choose some *hyperparameters*: $\sigma_f = 7$, $\ell = 100$

$$t = \begin{bmatrix} 0700 \\ 0800 \\ 1029 \end{bmatrix} \qquad K(t,t) = \{k(t_i,t_j)\}_{i,j} = \begin{bmatrix} 49.0 & 29.7 & 00.2 \\ 29.7 & 49.0 & 03.6 \\ 00.2 & 03.6 & 49.0 \end{bmatrix}$$

Kernel to Covariance Matrix

- Choose some *hyperparameters*: $\sigma_f = 7$, $\ell = 500$

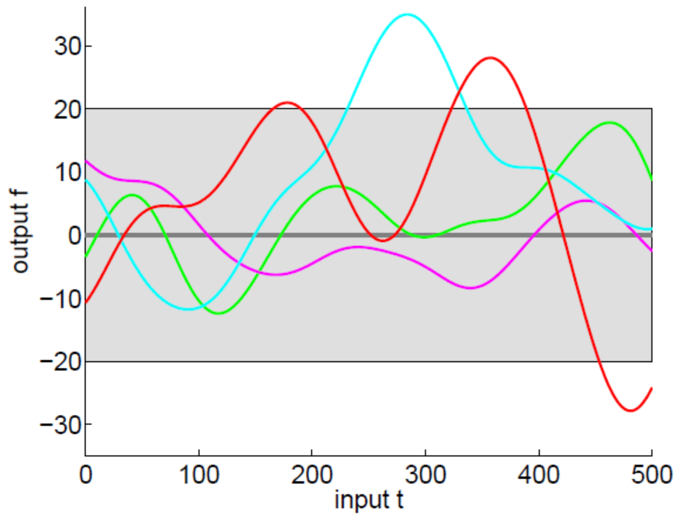
$$t = \begin{bmatrix} 0700 \\ 0800 \\ 1029 \end{bmatrix} \qquad K(t,t) = \{k(t_i, t_j)\}_{i,j} = \begin{bmatrix} 49.0 & 48.0 & 39.5 \\ 48.0 & 49.0 & 44.1 \\ 39.5 & 44.1 & 49.0 \end{bmatrix}$$

- Choose some *hyperparameters*: $\sigma_f = 14$, $\ell = 50$

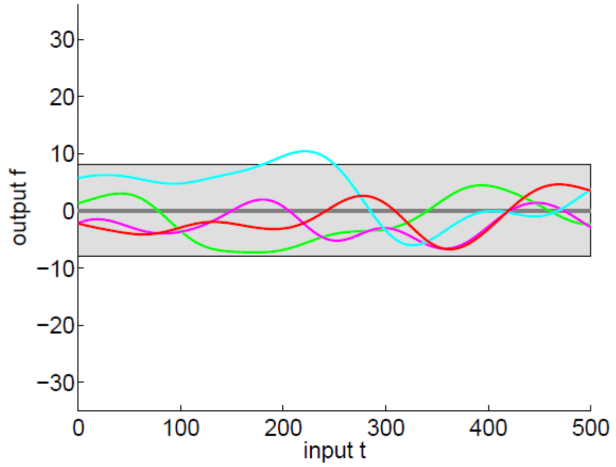
$$t = \begin{bmatrix} 0700 \\ 0800 \\ 1029 \end{bmatrix} \qquad K(t,t) = \{k(t_i, t_j)\}_{i,j} = \begin{bmatrix} 196 & 26.5 & 00.0 \\ 26.5 & 196 & 0.01 \\ 00.0 & 0.01 & 196 \end{bmatrix}$$

Samples from GP

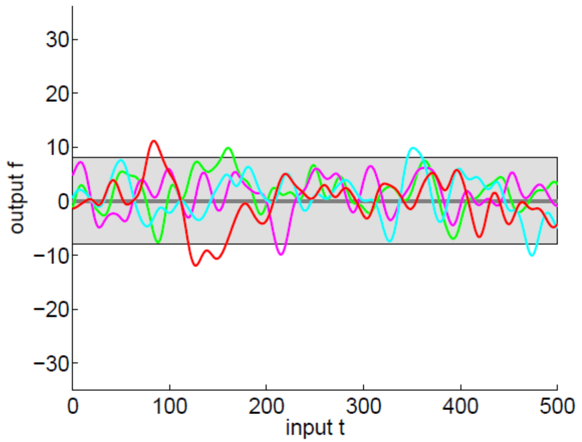
- $\sigma_f = 10$, $\ell = 50$



- $\sigma_f = 4$, $\ell = 50$



- $\sigma_f = 4$, $\ell = 10$



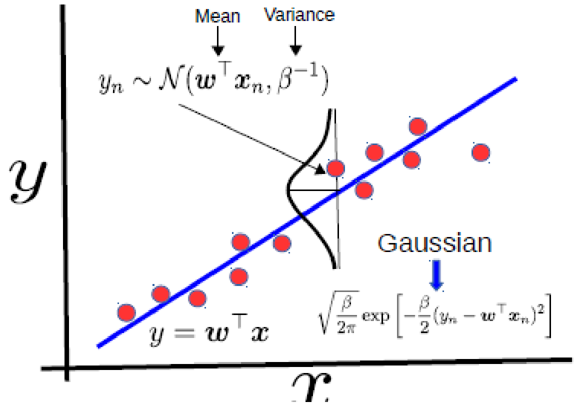
Bayesian Linear Regression

- Given: N training examples $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, features: $\boldsymbol{x}_n \in \mathbb{R}^D$, response $y_n \in \mathbb{R}$
- Assume a "noisy" linear model with regression weight vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_D] \in \mathbb{R}^D$

$$y_n = \boldsymbol{w}^\top \boldsymbol{x}_n + \epsilon_n$$

where $\epsilon_n \sim \mathcal{N}(0, \beta^{-1})$, β: precision (inverse variance) of Gaussian (assumed known)

- Therefore $p(y_n|\boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1})$



PPD: Linear Regression

- Let's first consider the (probabilistic) linear regression model

$$
\begin{aligned}
p(\boldsymbol{w}) &= \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_0, \Sigma_0) && \text{(Prior)} \\
p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) &= \mathcal{N}(\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N) && \text{(Likelihood w.r.t. } N \text{ obs.)} \\
p(\boldsymbol{y}|\mathbf{X}) &= \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w} = \mathcal{N}(\mathbf{X}\boldsymbol{\mu}_0, \beta^{-1}\mathbf{I}_N + \mathbf{X}\boldsymbol{\Sigma}_0\mathbf{X}^\top) && \text{(Marginal likelihood)} \\
p(\boldsymbol{y}|\mathbf{X}) &= \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I}_N + \mathbf{X}\mathbf{X}^\top) && \text{(if } \boldsymbol{\mu}_0 = 0 \text{ and } \boldsymbol{\Sigma}_0 = \mathbf{I}) \\
p(\boldsymbol{y}|\mathbf{X}) &= \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^\top) && \text{(if } \beta^{-1} = \infty \text{, i.e., zero noise)}
\end{aligned}
$$

- Thus the joint marginal distr. of $\boldsymbol{y}$ conditioned on $\mathbf{X}$ is the following multivariate Gaussian

$$
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{x}_1^\top \boldsymbol{x}_1 \dots \boldsymbol{x}_1^\top \boldsymbol{x}_N \\ \boldsymbol{x}_2^\top \boldsymbol{x}_1 \dots \boldsymbol{x}_2^\top \boldsymbol{x}_N \\ \vdots \quad \ddots \quad \vdots \\ \boldsymbol{x}_N^\top \boldsymbol{x}_1 \dots \boldsymbol{x}_N^\top \boldsymbol{x}_N \end{bmatrix} \right)
$$

Non-linear Regression

- Training data: $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$. $\boldsymbol{x}_n \in \mathbb{R}^D$, $y_n \in \mathbb{R}$
- Assume the responses to be a noisy function of the inputs

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n = f_n + \epsilon_n$$

- Assume a zero-mean Gaussian noise: $\epsilon_n \sim \mathcal{N}(\epsilon_n|0, \sigma^2)$
- This implies the following likelihood model: $p(y_n|f_n) = \mathcal{N}(y_n|f_n, \sigma^2)$
- Denote $\boldsymbol{f} = [f_1, \ldots, f_N]$ and $\boldsymbol{y} = [y_1, \ldots, y_N]$. For i.i.d. responses, the joint likelihood will be

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}_N)$$

- We now need a prior on the function f that enables us to model a nonlinear f
- Let's choose zero mean Gaussian Process prior $\mathcal{GP}(0, \kappa)$ on f, which is equivalent to

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

GP Regression

- The likelihood model: $p(\boldsymbol{y}|\mathbf{f}) = \mathcal{N}(\boldsymbol{y}|\mathbf{f}, \sigma^2\mathbf{I}_N)$. The prior distribution: $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$

  The posterior $p(\mathbf{f}|\boldsymbol{y}) \propto p(\mathbf{f})p(\boldsymbol{y}|\mathbf{f})$, which will be another Gaussian

- What's the posterior predictive $p(y_*|\boldsymbol{x}_*, \boldsymbol{y}, \mathbf{X})$ or $p(y_*|\boldsymbol{y})$ (skipping $\mathbf{X}, \boldsymbol{x}_*$ from the notation)?

$$p(y_*|\boldsymbol{y}) = \int p(y_*|f_*)p(f_*|\boldsymbol{y})df_*$$

  where $p(f_*|\boldsymbol{y}) = \int p(f_*|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{y})d\boldsymbol{f}$ and note that $p(f_*|\boldsymbol{f})$ must be Gaussian for GP

- For this case (GP regression), we actually don't need to compute $p(y_*|\boldsymbol{y})$ using the above method

Partitioned Multivariate Gaussian

- Consider a multi-variate Gaussian and partition random vector into (X,Y).

$$\mathcal{N}(\mu, \Sigma) = \mathcal{N}\left(\begin{bmatrix}\mu_X \\ \mu_Y\end{bmatrix}, \begin{bmatrix}\Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY}\end{bmatrix}\right)$$

Then

$$\begin{aligned} X &\sim \mathcal{N}(\mu_X, \Sigma_{XX}) \\ Y &\sim \mathcal{N}(\mu_Y, \Sigma_{YY}) \end{aligned}$$

$$\begin{aligned} X|Y = y_0 &\sim \mathcal{N}(\mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(y_0 - \mu_Y), \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}) \\ Y|X = x_0 &\sim \mathcal{N}(\mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x_0 - \mu_X), \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}) \end{aligned}$$

- Mean moved according to correlation and variance on measurement
- Covariance $\Sigma_{XX\,|\,Y=y0}$ does not depend on y_0

Simpler Solution for PPD in GP

Let $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ be a training set of i.i.d. examples from some unknown distribution. In the Gaussian process regression model,

$$y^{(i)} = f(x^{(i)}) + \varepsilon^{(i)}, \qquad i = 1, \ldots, m$$

$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot))$ Gaussian Process Prior

PPD

Now, let $T = \{(x_*^{(i)}, y_*^{(i)})\}_{i=1}^{m_*}$ be a set of i.i.d. testing points

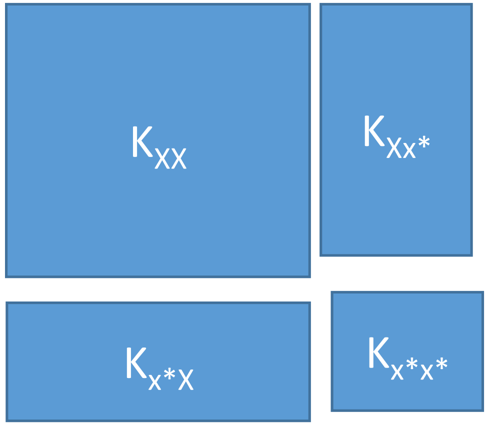
$$X = \begin{bmatrix} — & (x^{(1)})^T & — \\ — & (x^{(2)})^T & — \\ & \vdots & \\ — & (x^{(m)})^T & — \end{bmatrix} \in \mathbf{R}^{m \times n} \quad \vec{f} = \begin{bmatrix} f(x^{(1)}) \\ f(x^{(2)}) \\ \vdots \\ f(x^{(m)}) \end{bmatrix}, \quad \vec{\varepsilon} = \begin{bmatrix} \varepsilon^{(1)} \\ \varepsilon^{(2)} \\ \vdots \\ \varepsilon^{(m)} \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbf{R}^{m},$$

$$X_* = \begin{bmatrix} — & (x_*^{(1)})^T & — \\ — & (x_*^{(2)})^T & — \\ & \vdots & \\ — & (x_*^{(m_*)})^T & — \end{bmatrix} \in \mathbf{R}^{m_* \times n} \quad \vec{f}_* = \begin{bmatrix} f(x_*^{(1)}) \\ f(x_*^{(2)}) \\ \vdots \\ f(x_*^{(m_*)}) \end{bmatrix}, \quad \vec{\varepsilon}_* = \begin{bmatrix} \varepsilon_*^{(1)} \\ \varepsilon_*^{(2)} \\ \vdots \\ \varepsilon_*^{(m_*)} \end{bmatrix}, \quad \vec{y}_* = \begin{bmatrix} y_*^{(1)} \\ y_*^{(2)} \\ \vdots \\ y_*^{(m_*)} \end{bmatrix} \in \mathbf{R}^{m_*}.$$

PPD

Recall that for any function $f(\cdot)$ drawn from our zero-mean Gaussian process prior with covariance function $k(\cdot, \cdot)$, the marginal distribution over any set of input points belonging to $\mathcal{X}$ must have a joint multivariate Gaussian distribution. In particular, this must hold for the training and test points, so we have

$$\begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} \Bigg| X, X_* \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right),$$



From our i.i.d. noise assumption, we have that

$$\begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}_* \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} \sigma^2 I & \vec{0} \\ \vec{0}^T & \sigma^2 I \end{bmatrix}\right).$$

PPD

The sums of independent Gaussian random variables is also Gaussian, so

$$\begin{bmatrix} \vec{y} \\ \vec{y}_* \end{bmatrix} \Bigg| X, X_* = \begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} + \begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}_* \end{bmatrix} \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) + \sigma^2 I \end{bmatrix}\right).$$

Now, using the rules for conditioning Gaussians, it follows that

$$\vec{y_*} \mid \vec{y}, X, X_* \sim \mathcal{N}(\mu^*, \Sigma^*)$$

$$\mu^* = K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} \vec{y}$$

$$\Sigma^* = K(X_*, X_*) + \sigma^2 I - K(X_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} K(X, X_*).$$

GP Regression (Single Test Point)

- Reason: The marginal distribution of the training data responses $\boldsymbol{y}$

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}_N) = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \mathbf{C}_N)$$

- Using the same result, the marginal distribution $p(y_*) = \mathcal{N}(y_*|0, \kappa(\boldsymbol{x}_*, \boldsymbol{x}_*) + \sigma^2)$

  - Let's consider the joint distr. of N training responses $\boldsymbol{y}$ and test response y_*

$$p\left(\begin{bmatrix} \boldsymbol{y} \\ y_* \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y} \\ y_* \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \mathbf{C}_{N+1}\right)$$

where the $(N+1) \times (N+1)$ matrix $\mathbf{C}_{N+1}$ is given by

$$\mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{k}_* \\ \mathbf{k}_*{}^\top & c \end{bmatrix}$$

and $\mathbf{k}_* = [\kappa(\boldsymbol{x}_*, \boldsymbol{x}_1), \ldots, \kappa(\boldsymbol{x}_*, \boldsymbol{x}_N)]^\top$, $c = \kappa(\boldsymbol{x}_*, \boldsymbol{x}_*) + \sigma^2$

GP Regression (Single Test Point)

- Let's look at the predictions made by GP regression

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \mathcal{N}(y_*|\mu_*, \sigma_*^2) \\
\mu_* &= \mathbf{k}_*{}^\top \mathbf{C}_N^{-1} \mathbf{y} \\
\sigma_*^2 &= k(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2 - \mathbf{k}_*{}^\top \mathbf{C}_N^{-1} \mathbf{k}_*
\end{aligned}
$$

Interpretation of GP Regression

- Two interpretations for the mean prediction μ_*
  - A kernel SVM like interpretation

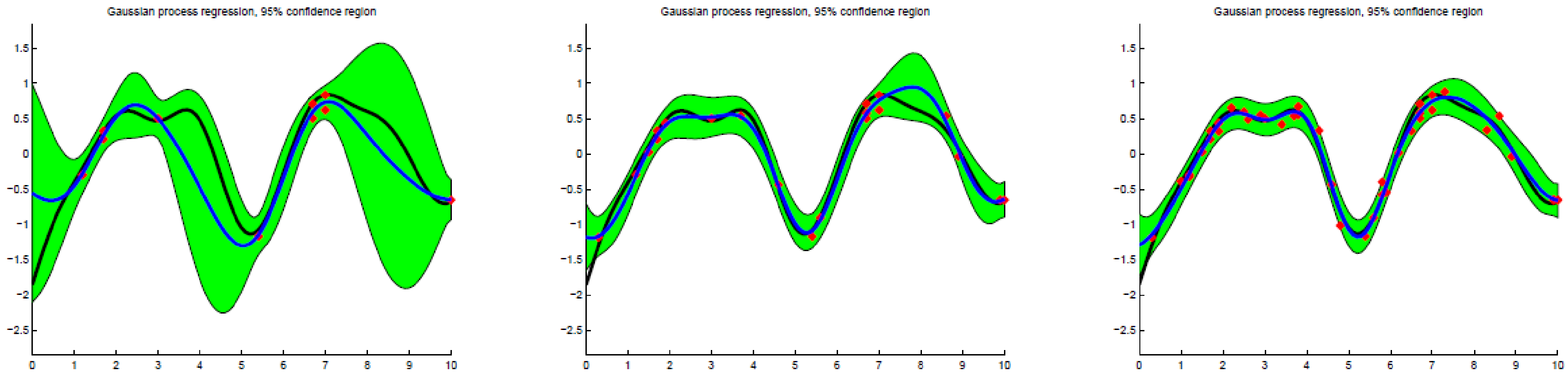
$$\mu_* = \mathbf{k}_*^\top \mathbf{C}_N^{-1} \boldsymbol{y} = \mathbf{k}_*^\top \boldsymbol{\alpha} = \sum_{n=1}^{N} k(\boldsymbol{x}_*, \boldsymbol{x}_n)\alpha_n$$

    where $\boldsymbol{\alpha}$ is akin to the weights of support vectors

  - A nearest neighbors interpretation

$$\mu_* = \mathbf{k}_*^\top \mathbf{C}_N^{-1} \boldsymbol{y} = \boldsymbol{w}^\top \boldsymbol{y} = \sum_{n=1}^{N} w_n y_n$$

Updates on PPD with more observations



(a) (b) (c)

GP Regression: Design Choices

- Revisit the model and see what can be hacked:

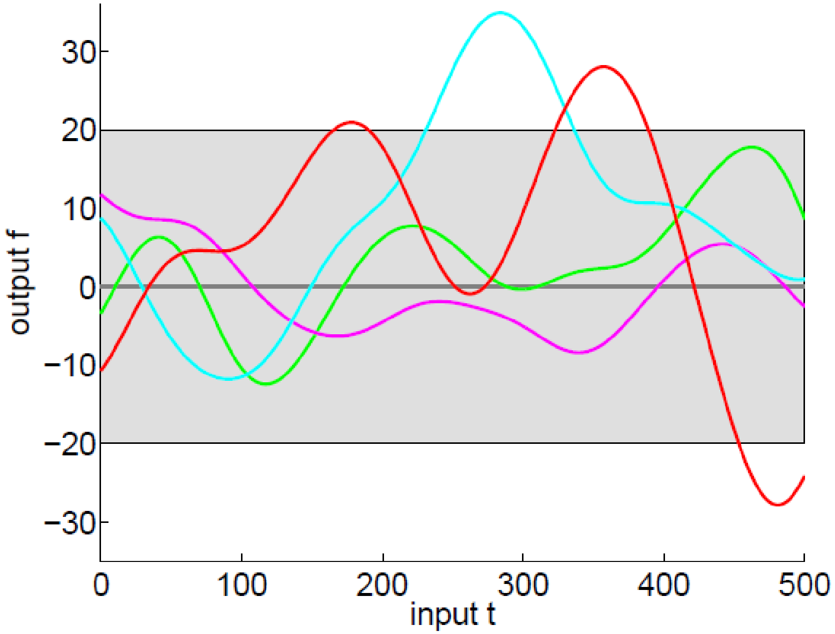
$$f \sim \mathcal{GP}(0, k_{ff}), \quad \text{where} \quad k_{ff}(t_i, t_j) = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\}$$

$$y_i | f_i \sim \mathcal{N}(f_i, \sigma_n^2 I)$$

- Option 1: hyperparameters $\rightarrow$ model selection.
- Option 2: functional form of k_{ff} $\rightarrow$ kernel choices.
- Option 3: the GP?
- Option 4: the data distribution $\rightarrow$ likelihood choices.

Kernel: Squared Exponential

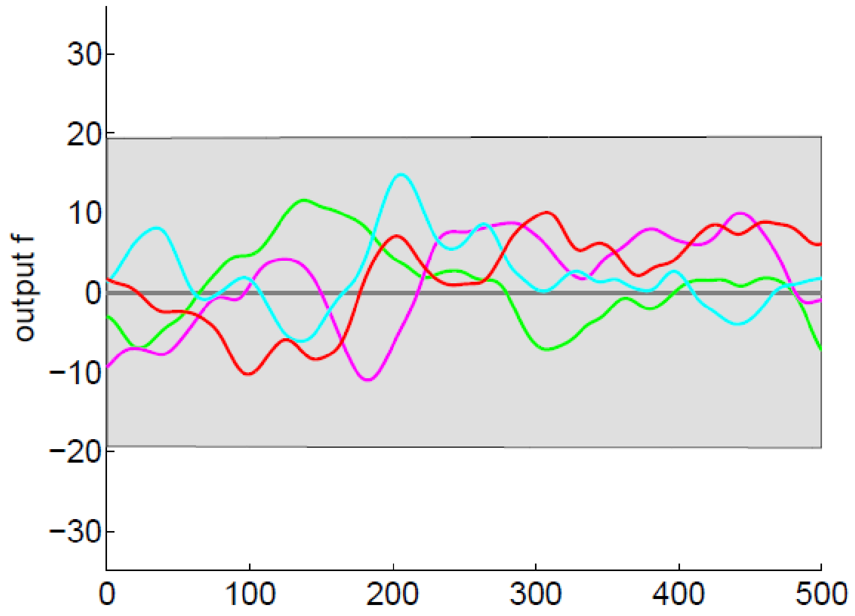
$$k(t_i, t_j) = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\}$$



Kernel: Rational Quadratic

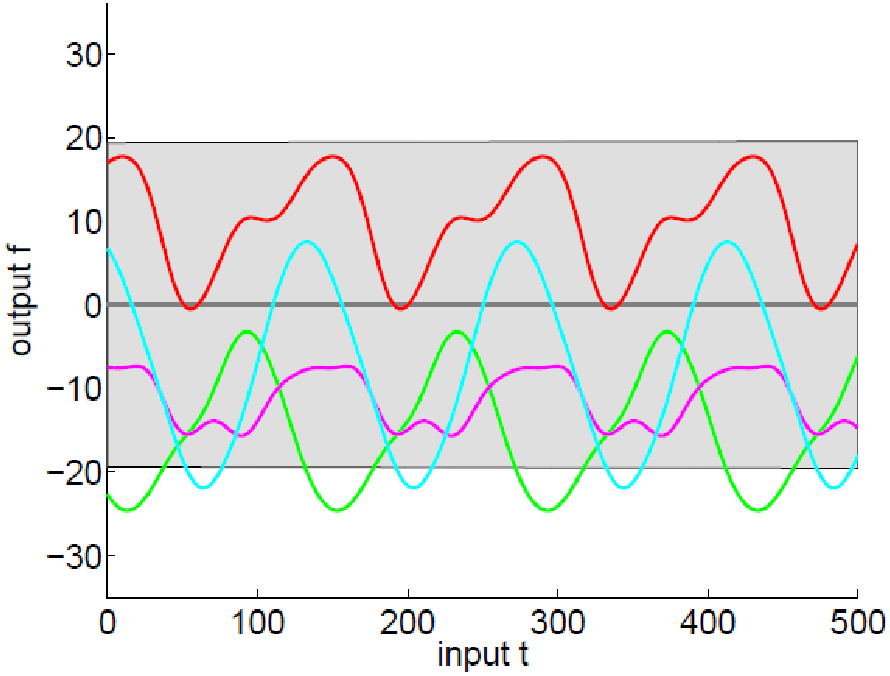
$$k(t_i, t_j) = \sigma_f^2 \left(1 + \frac{1}{2\alpha\ell^2}(t_i - t_j)^2\right)^{-\alpha}$$

$$\propto \quad \sigma_f^2 \int z^{\alpha-1} \exp\left(-\frac{\alpha z}{\beta}\right) \exp\left(-\frac{z(t_i - t_j)^2}{2}\right) dz$$



Kernel: Periodic

$$k(t_i, t_j) = \sigma_f^2 \exp\left\{-\frac{2}{\ell^2}\sin^2\left(\frac{\pi}{p}|t_i - t_j|\right)\right\}$$



Kernel Compositions

- Linear: $k(t_i, t_j) = \alpha k_1(t_i, t_j) + \beta k_2(t_i, t_j)$ (for $\alpha, \beta \geq 0$)

  or $k\left(x^{(i)}, x^{(j)}\right) = k_a\left(x_1^{(i)}, x_1^{(j)}\right) + k_b\left(x_2^{(i)}, x_2^{(j)}\right)$

- Products: $k(t_i, t_j) = k_1(t_i, t_j)k_2(t_i, t_j)$

- Integration: $z(t) = \int g(u, t)f(u)du \quad \leftrightarrow$

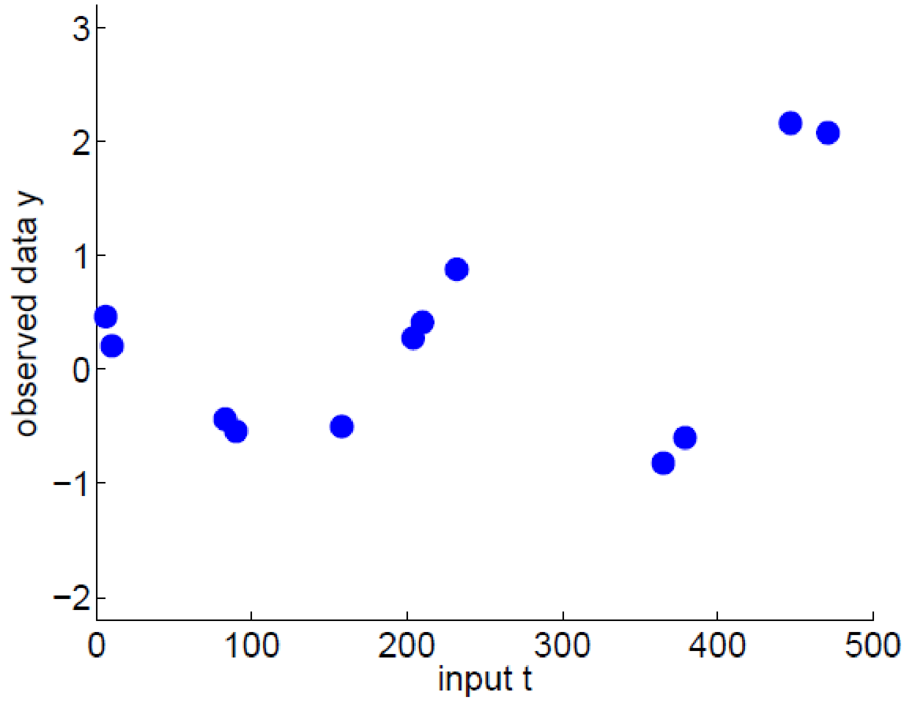
  $k_z(t_i, t_j) = \int \int g(u, t_1)k_f(t_i, t_j)g(v, t_j)dudv$

- Differentiation: $z(t) = \frac{\partial}{\partial t}f(t) \quad \leftrightarrow \quad k_z(t_i, t_j) = \frac{\partial^2}{\partial t_i \partial t_j}k_f(t_i, t_j)$

- Warping: $z(t) = f\left(h(t)\right) \quad \leftrightarrow \quad k_z(t_i, t_j) = k_f\left(h(t_i), h(t_j)\right)$

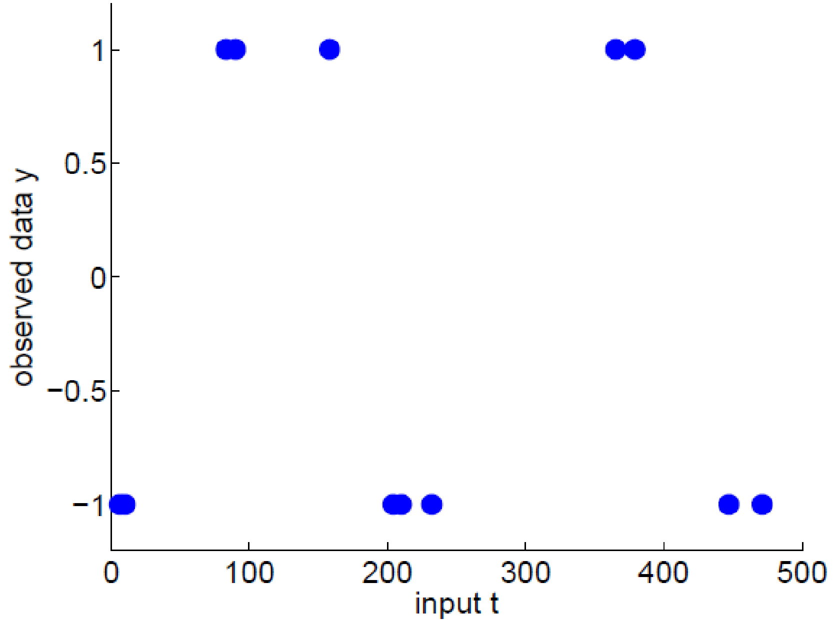
Likelihood Models: Regression

- data likelihood model: $y_i | f_i \sim \mathcal{N}(f_i, \sigma_n^2 I)$



Binary Label Data

- Classification (not regression) setting
- $y_i | f_i \sim \mathcal{N}(f_i, \sigma_n^2 I)$ is inappropriate

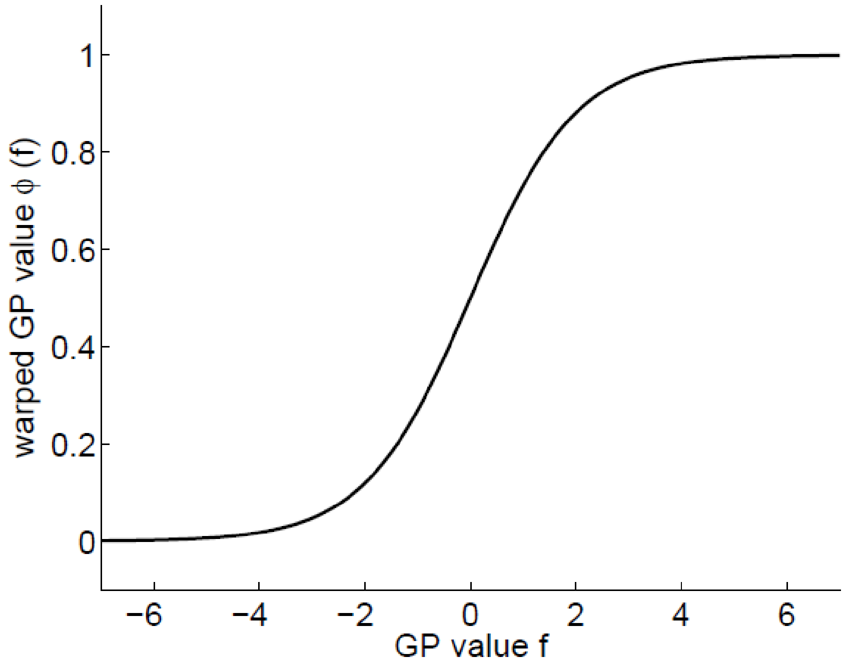


GP Classification

- Probit or Logistic "regression" model on $y_i \in \{-1, +1\}$:

$$p(y_i|f_i) = \phi(y_i f_i) = \frac{1}{1 + \exp(-y_i f_i)}$$

- Warps f onto the $[0, 1]$ interval

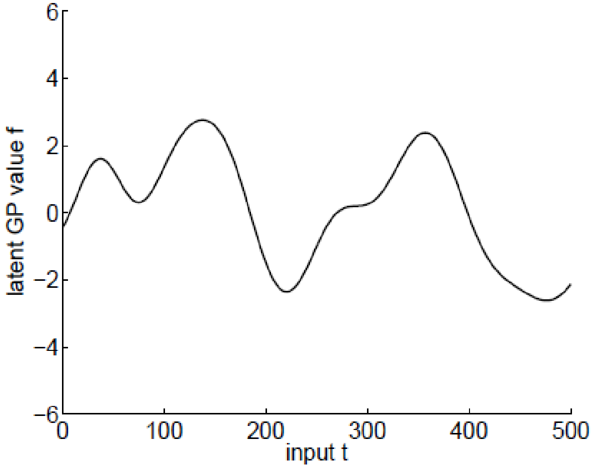


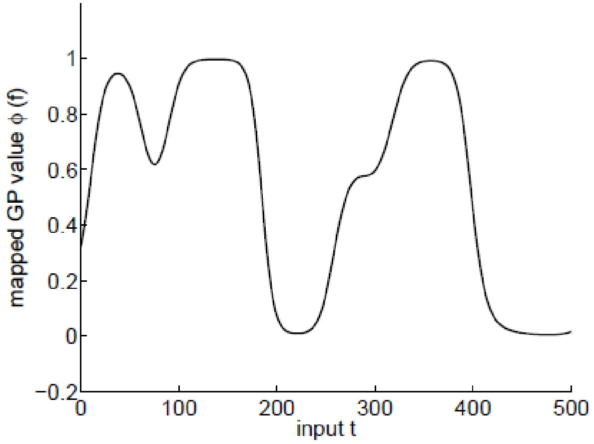
GP Classification

- Probit or Logistic "regression" model on $y_i \in \{-1, +1\}$:

$$p(y_i|f_i) = \phi(y_i f_i) = \frac{1}{1 + \exp(-y_i f_i)}$$

- Warps f onto the $[0, 1]$ interval





Scalability of GP

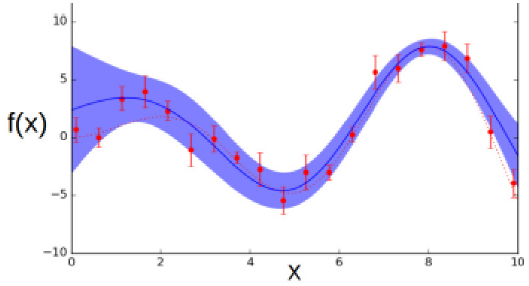
- Computational costs in some steps of GP based models scale in the size of training data
  - E.g., test time prediction in GP regression takes $O(N)$ time

$$
\begin{aligned}
p(y_*|\mathbf{y}) &= \mathcal{N}(y_*|\mu_*, \sigma_*^2) \\
\mu_* &= \mathbf{k}_*{}^\top \mathbf{C}_N^{-1}\mathbf{y} \qquad\qquad (O(N) \text{ cost assuming } \mathbf{C}_N^{-1} \text{ is pre-computed}) \\
\sigma_*^2 &= k(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2 - \mathbf{k}_*{}^\top \mathbf{C}_N^{-1}\mathbf{k}_*
\end{aligned}
$$

- GP models often require matrix inversions - takes $O(N^3)$ time. Storage also requires $O(N^2)$ space

Summary

- GPs enable us to learn nonlinear functions while also capturing the uncertainty



- Uncertainty can tell us where to acquire more training data to improve the function's estimate
  - Especially useful if we can't get too many training examples (e.g., expensive inputs and/or labels)

References

- Rasmussen and Williams, *Gaussian Processes for Machine Learning*
- Bishop, *Pattern Recognition and Machine Learning*
- www.gaussianprocess.org (better updated/kept than .com)

Programming Assignment I

- We have a data set consisting of the number of COVID-19 infections in World and in India for each day since 31-12-2109
  - https://ourworldindata.org/coronavirus-source-data
- Choose any GP prior (kernel function as well as hyperparameters)
  - You may use domain knowledge for this choice
- Obtain the mean and variance for the predictions for the last 15 days (as test points)
- Bonus: hyperparameter optimization, use of non-stationery kernels like Wiener Process
- Submit your program in python/Julia/C/C++, and a report by Oct. 2, 2020