MACHINE LEARNING
DEPARTMENT

Introduction to Machine Learning

12. Gaussian Processes

Alex Smola
Carnegie Mellon University

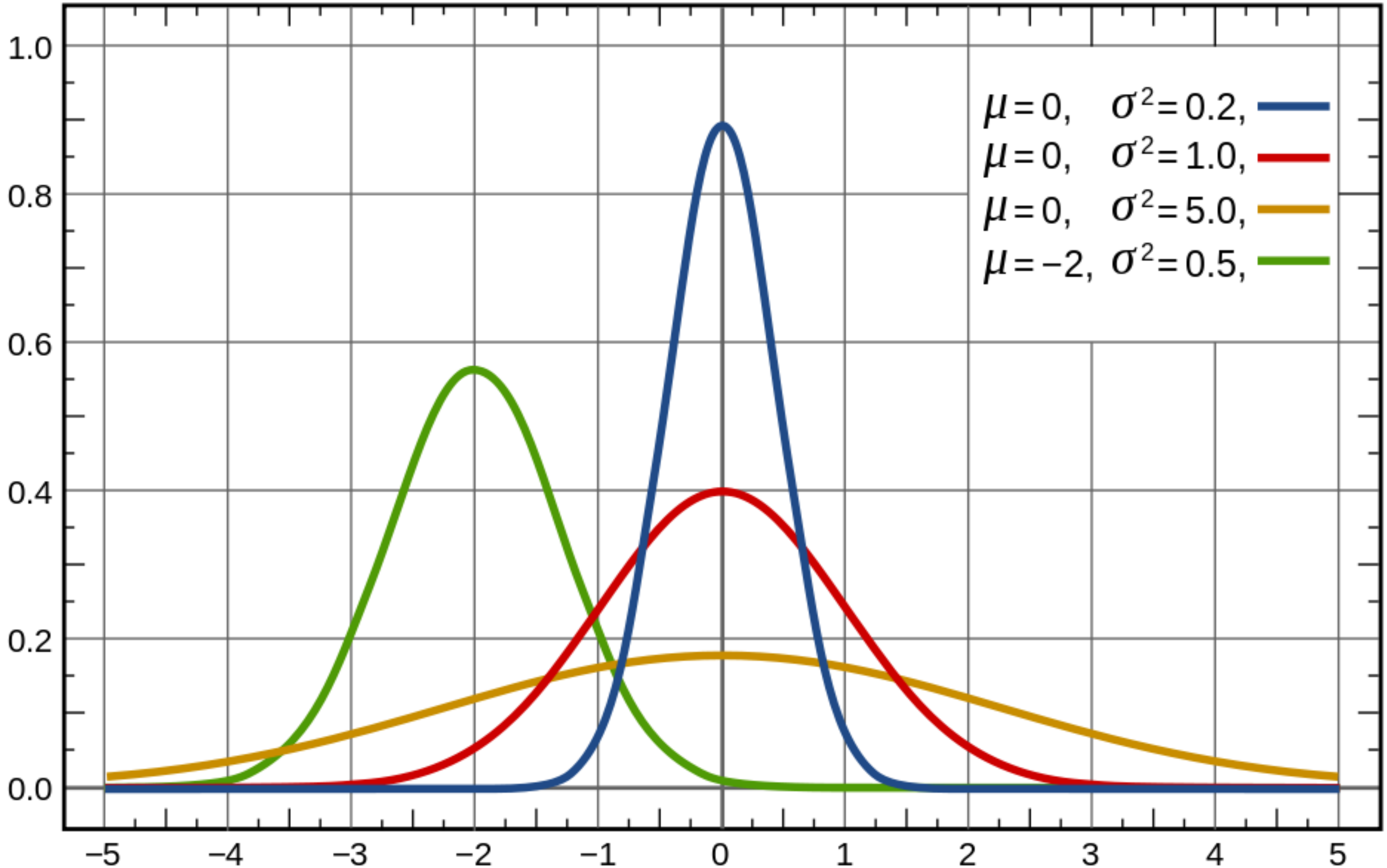
http://alex.smola.org/teaching/cmu2013-10-701
10-701

The Normal Distribution

http://www.gaussianprocess.org/gpml/chapters/

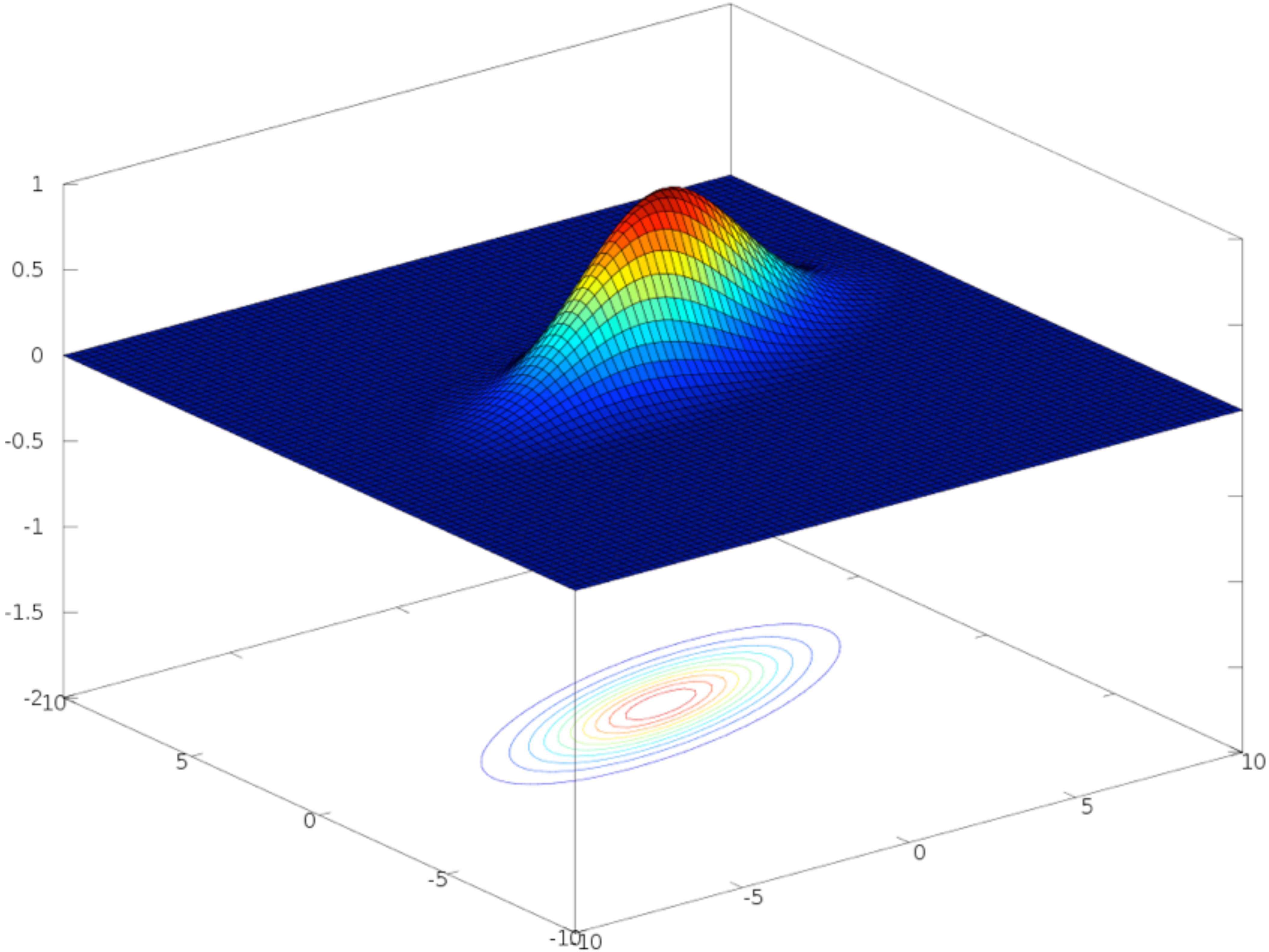
Carnegie Mellon University

The Normal Distribution



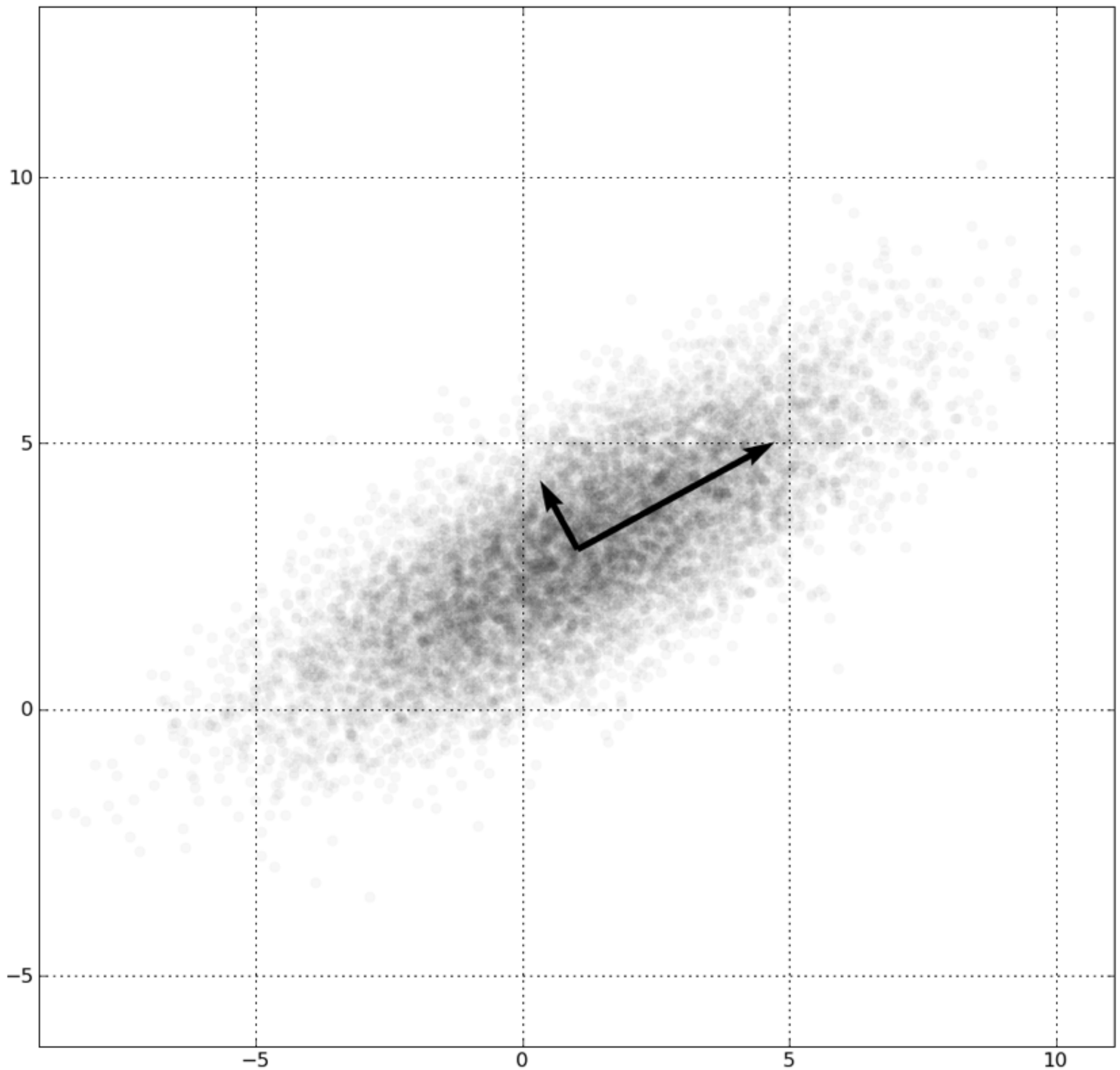
Carnegie Mellon University

Gaussians in Space



University

Gaussians in Space



samples in R^2

The Normal Distribution

- Density for scalar variables

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)}$$

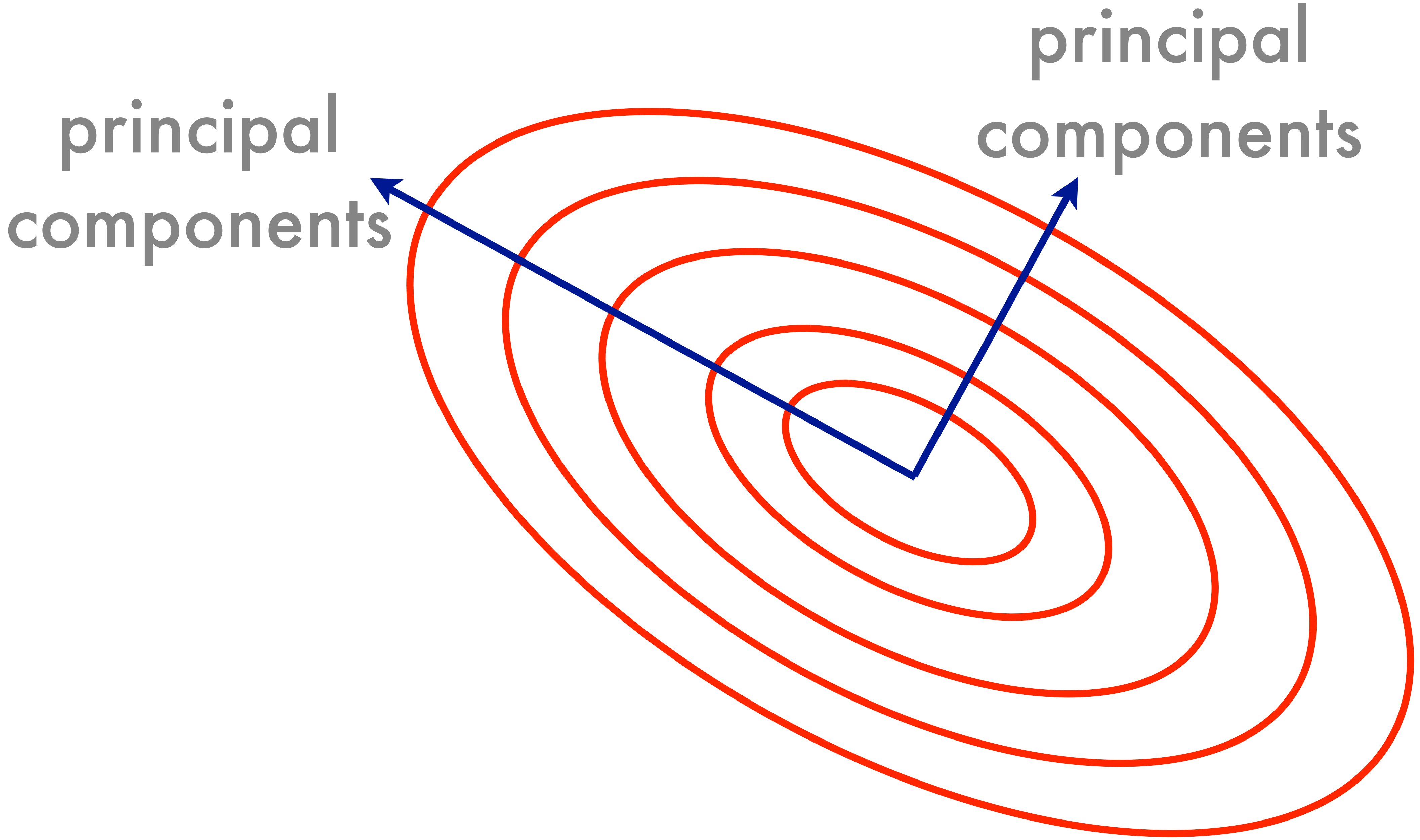
- Density in d dimensions

$$p(x) = (2\pi)^{-\frac{d}{2}} \left|\Sigma\right|^{-1} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$$

- Principal components
  - Eigenvalue decomposition $\Sigma = U^\top \Lambda U$
  - Product representation

$$p(x) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}(U(x-\mu))^\top \Lambda^{-1} U(x-\mu)}$$

The Normal Distribution



$$\Sigma = U^\top \Lambda U$$

$$p(x) = (2\pi)^{-\frac{d}{2}} \prod_{i=1}^{d} \Lambda_{ii}^{-1} e^{-\frac{1}{2}(U(x-\mu))^\top \Lambda^{-1} U(x-\mu)}$$

Why do we care?

- Central limit theorem shows that in the limit all averages behave like Gaussians
- Easy to estimate parameters (MLE)

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ and } \Sigma = \frac{1}{m}\sum_{i=1}^{m} x_i x_i^\top - \mu\mu^\top$$

- Distribution with largest uncertainty (entropy) for a given mean and covariance.
- Works well even if the assumptions are wrong

Why do we care?

- Central limit theorem shows that in the limit all averages behave like Gaussians
- Easy to estimate parameters (MLE)

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ and } \Sigma = \frac{1}{m}\sum_{i=1}^{m} x_i x_i^\top - \mu\mu^\top$$

```
X: data
m: sample size

mu    = (1/m)*sum(X,2)
sigma = (1/m)*X*X'- mu*mu'
```

Sampling from a Gaussian

- Case 1 - We have a normal distribution (randn)
  - We want $x \sim \mathcal{N}(\mu, \Sigma)$
  - Recipe: $x = \mu + Lz$ where $z \sim \mathcal{N}(0, \mathbf{1})$ and $\Sigma = LL^\top$
  - Proof: $\mathbf{E}\left[(x-\mu)(x-\mu)^\top\right] = \mathbf{E}\left[Lzz^\top L^\top\right]$

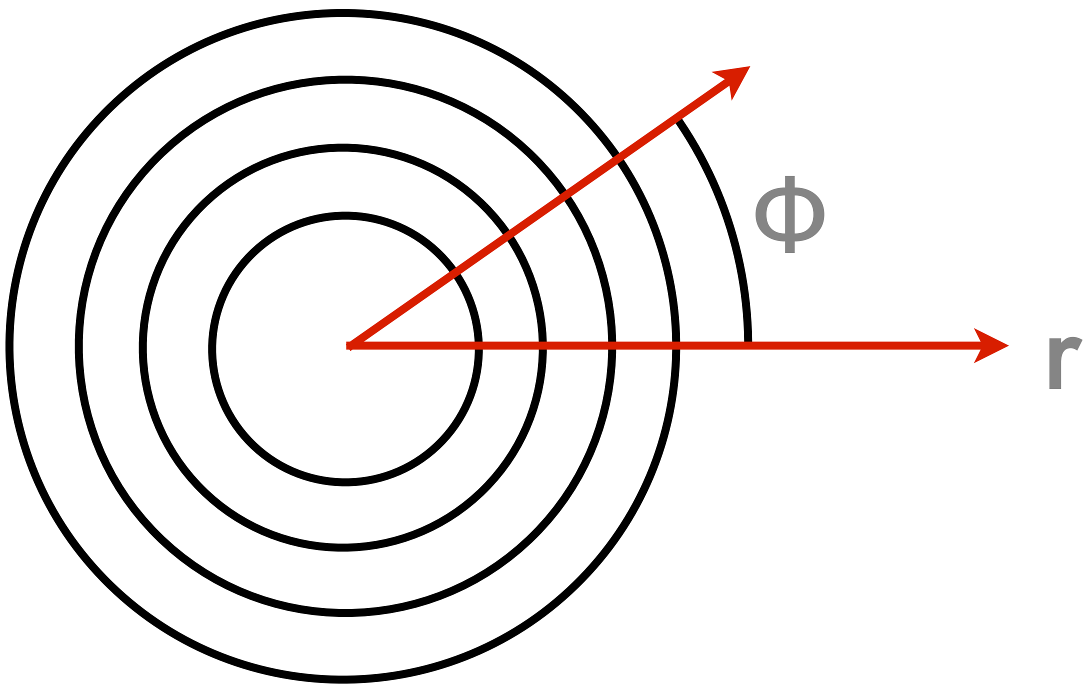
$$= L\mathbf{E}\left[zz^\top\right]L^\top = LL^\top = \Sigma$$

- Case 2 - Box-Müller transform for U[0,1]

$$p(x) = \frac{1}{2\pi}e^{-\frac{1}{2}\|x\|^2} \implies p(\phi, r) = \frac{1}{2\pi}e^{-\frac{1}{2}r^2}$$

$$F(\phi, r) = \frac{\phi}{2\pi} \cdot \left[1 - e^{-\frac{1}{2}r^2}\right]$$

Sampling from a Gaussian



$$p(x) = \frac{1}{2\pi} e^{-\frac{1}{2}\|x\|^2} \Longrightarrow p(\phi, r) = \frac{1}{2\pi} e^{-\frac{1}{2}r^2}$$

$$F(\phi, r) = \frac{\phi}{2\pi} \cdot \left[1 - e^{-\frac{1}{2}r^2}\right]$$

Sampling from a Gaussian

- Cumulative distribution function

$$F(\phi, r) = \frac{\phi}{2\pi} \cdot \left[1 - e^{-\frac{1}{2}r^2}\right]$$

Draw radial and angle component separately

```
tmp1 = rand()
tmp2 = rand()
r    = sqrt(-2*log(tmp1))
x1   = r*sin(tmp2/(2*pi))
x2   = r*cos(tmp2/(2*pi))
```

Sampling from a Gaussian

- Cumulative distribution function

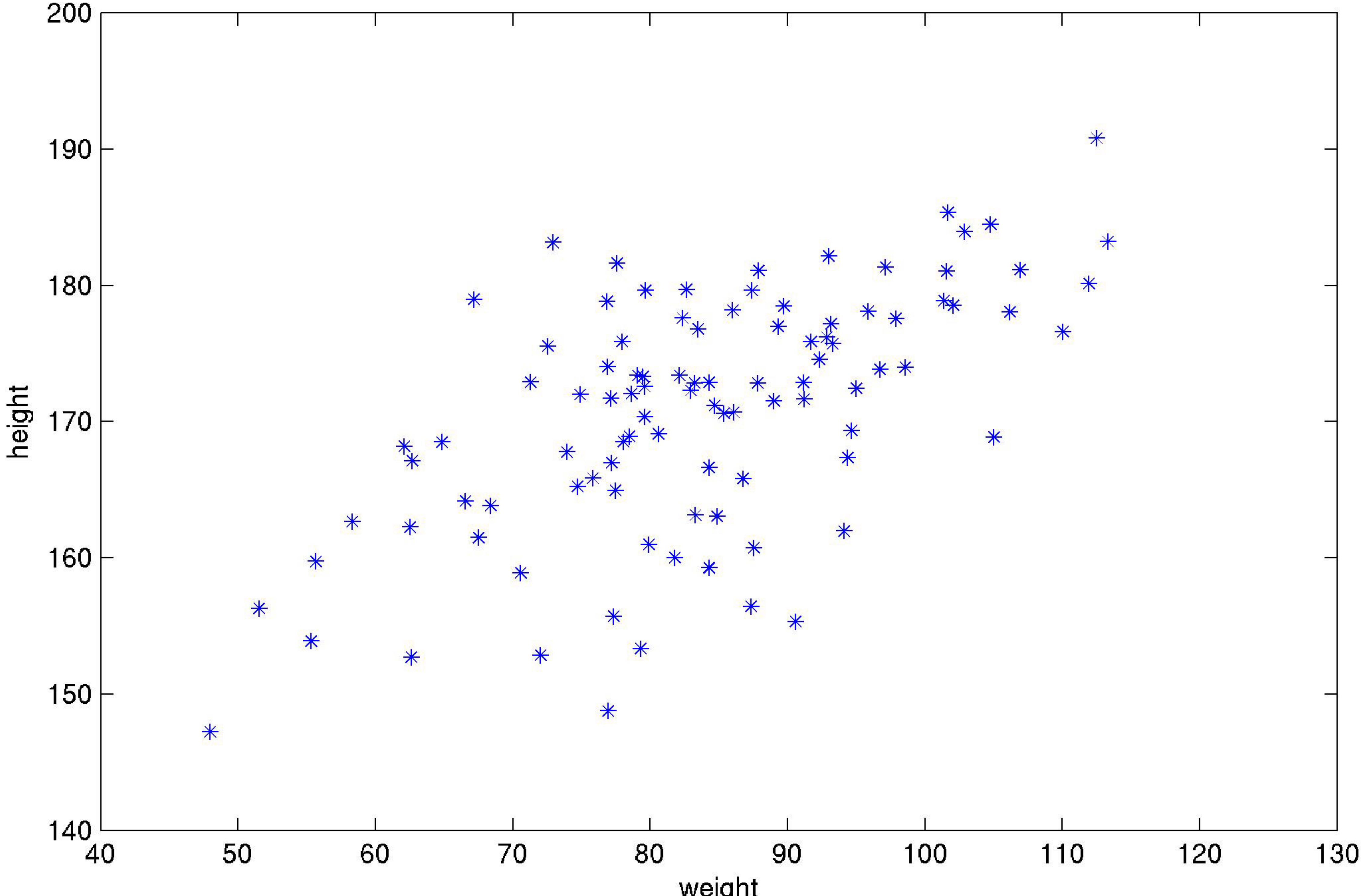
$$F(\phi, r) = \frac{\phi}{2\pi} \cdot \left[1 - e^{-\frac{1}{2}r^2}\right]$$

Draw radial and angle component separately

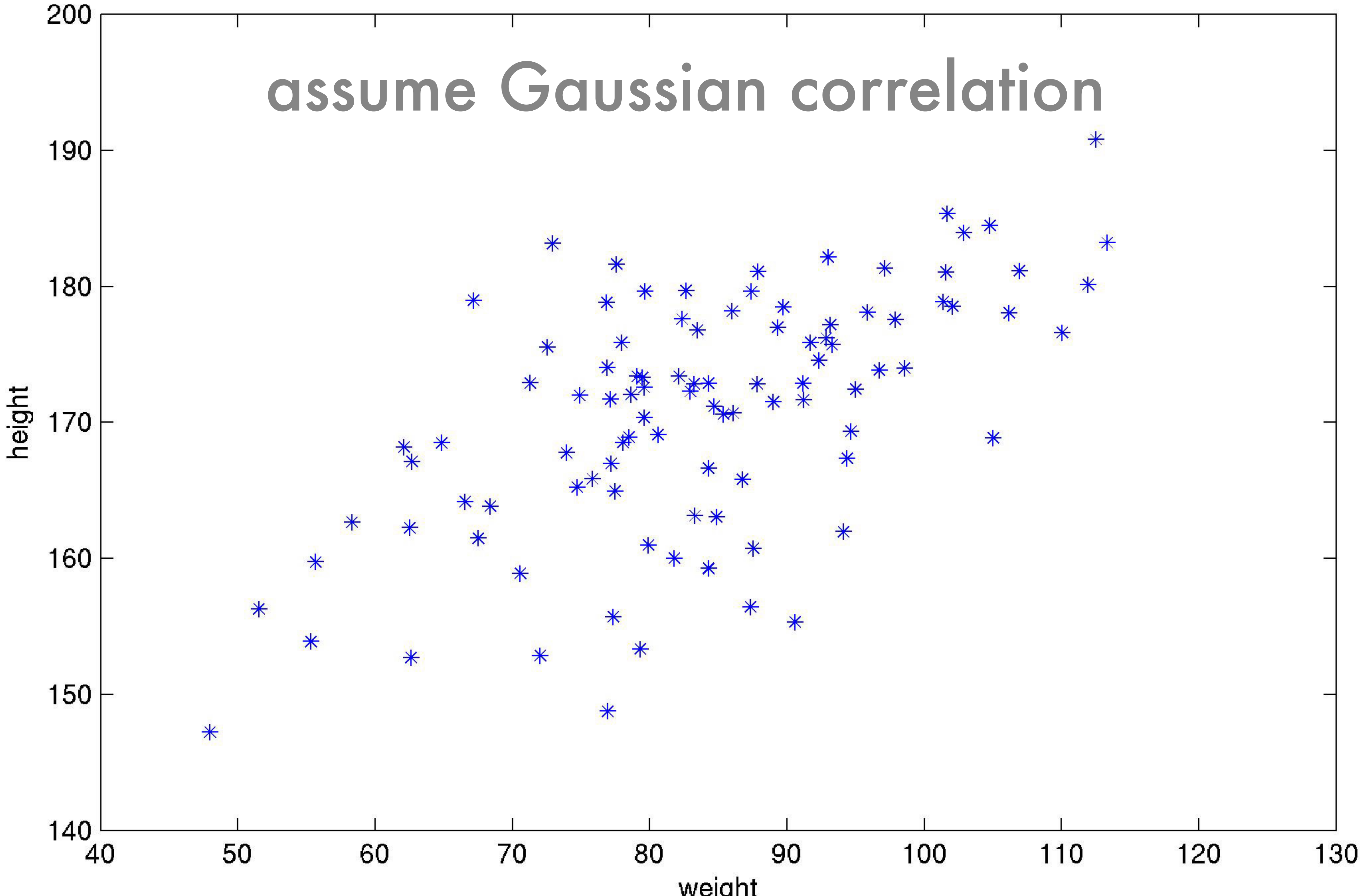
```
tmp1 = rand()
tmp2 = rand()
r    = sqrt(-2*log(tmp1))
x1   = r*sin(tmp2/(2*pi))
x2   = r*cos(tmp2/(2*pi))
```

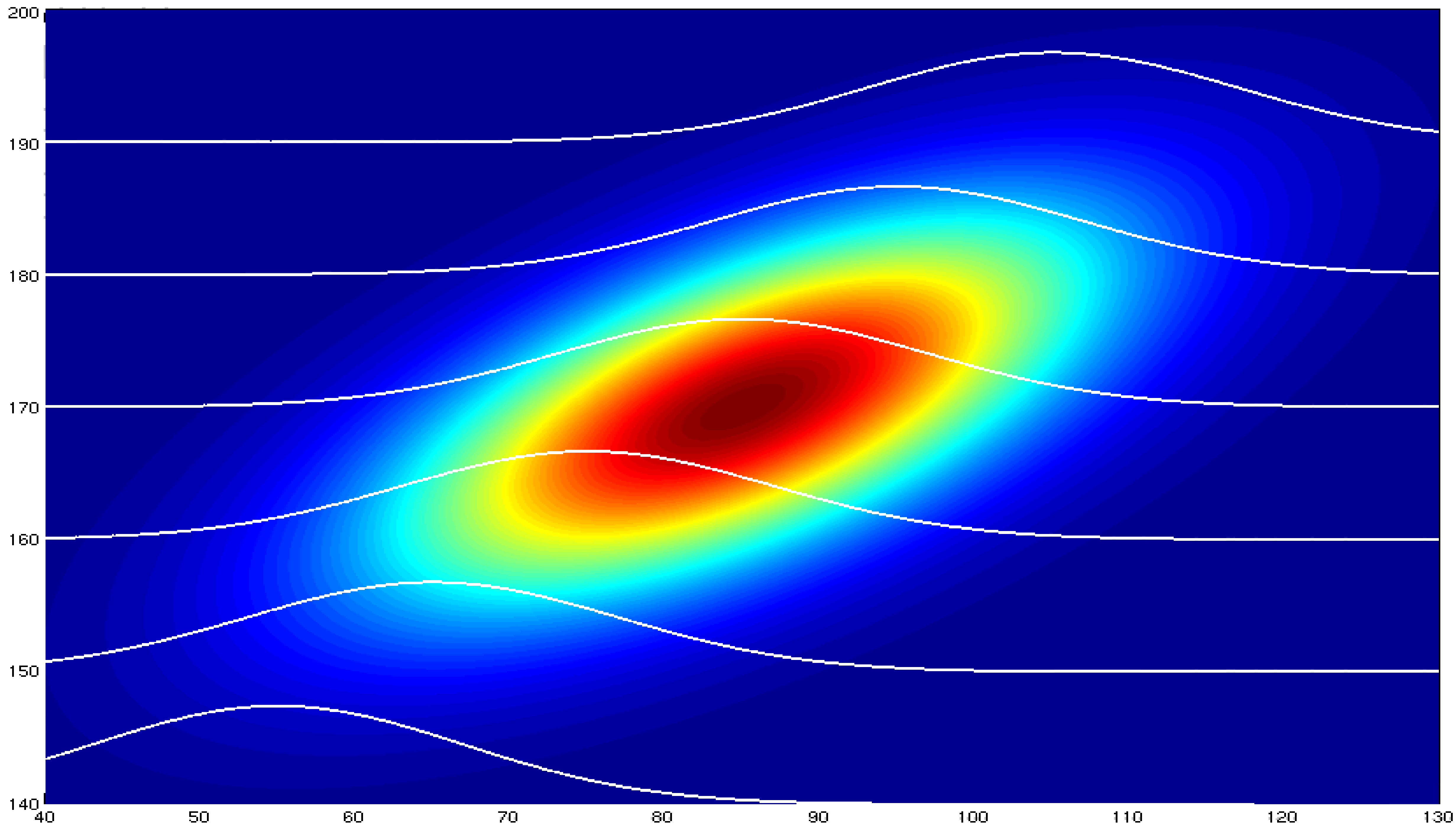
Why can we use tmp1 instead of 1-tmp1?

Example: correlating weight and height

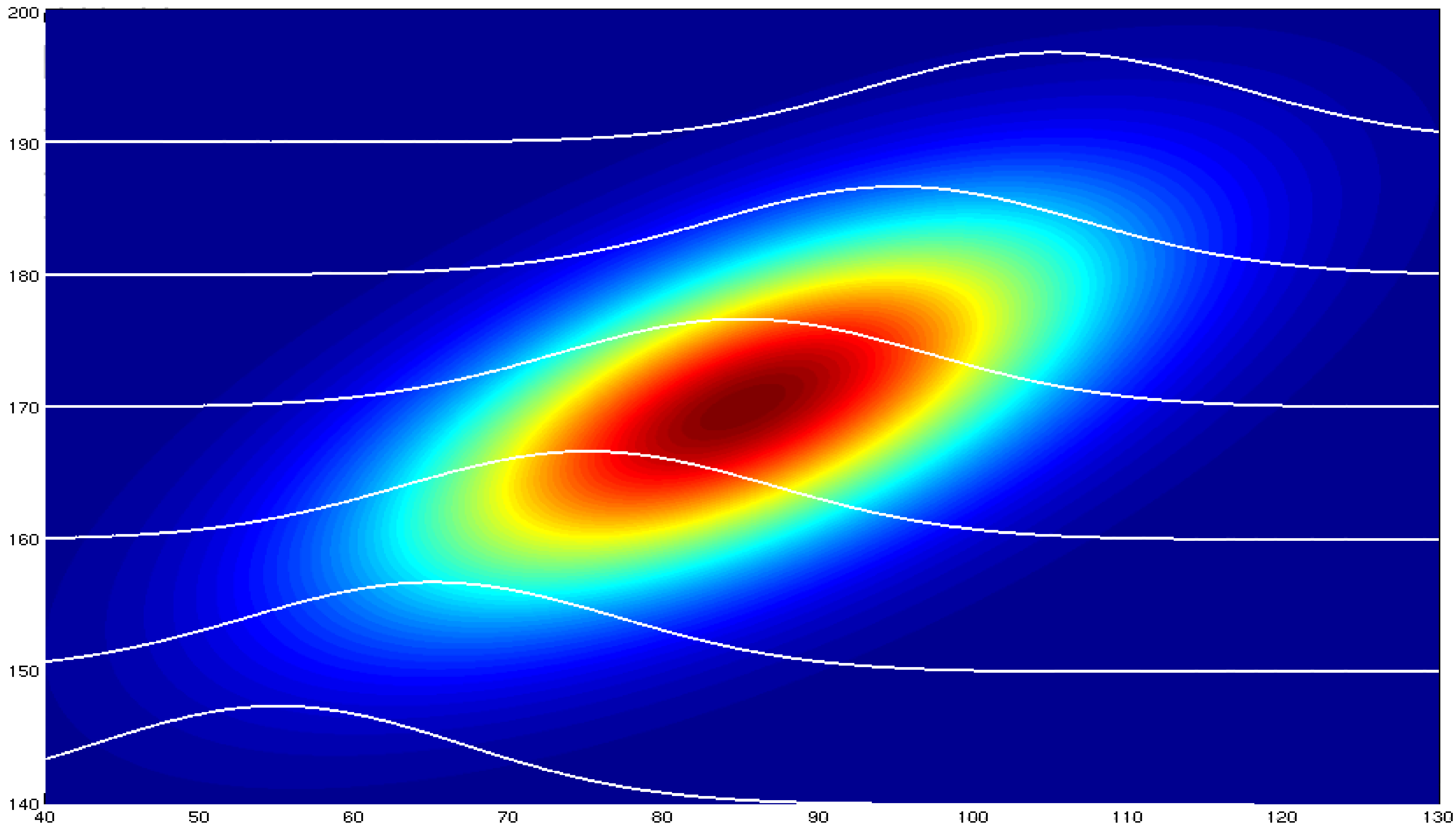


Example: correlating weight and height





$$p(\text{weight}|\text{height}) = \frac{p(\text{height}, \text{weight})}{p(\text{height})} \propto p(\text{height}, \text{weight})$$



$$p(x_2|x_1) \propto \exp\left[-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^\top \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{22} \end{bmatrix}^{-1} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right]$$

keep linear and quadratic terms of exponent

The gory math

Correlated Observations

Assume that the random variables $t \in \mathbb{R}^n, t' \in \mathbb{R}^{n'}$ are jointly normal with mean (μ, μ') and covariance matrix K

$$p(t, t') \propto \exp\left(-\frac{1}{2}\begin{bmatrix} t-\mu \\ t'-\mu' \end{bmatrix}^\top \begin{bmatrix} K_{tt} & K_{tt'} \\ K_{tt'}^\top & K_{t't'} \end{bmatrix}^{-1} \begin{bmatrix} t-\mu \\ t'-\mu' \end{bmatrix}\right).$$

Inference

Given t, estimate t' via $p(t'|t)$. Translation into machine learning language: we learn t' from t.

Practical Solution

Since $t'|t \sim \mathcal{N}(\tilde{\mu}, \tilde{K})$, we only need to collect all terms in $p(t, t')$ depending on t' by matrix inversion, hence

$$\tilde{K} = K_{t't'} - K_{tt'}^\top K_{tt}^{-1} K_{tt'} \text{ and } \tilde{\mu} = \mu' + K_{tt'}^\top \underbrace{\left[K_{tt}^{-1}(t-\mu)\right]}_{\text{independent of } t'}$$

Handbook of Matrices, Lütkepohl 1997 (big timesaver)

Mini Summary

- Normal distribution

$$p(x) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-1} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$$

- Sampling from $x \sim \mathcal{N}(\mu, \Sigma)$
  Use $x = \mu + Lz$ where $z \sim \mathcal{N}(0, 1)$ and $\Sigma = LL^\top$
- Estimating mean and variance

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ and } \Sigma = \frac{1}{m}\sum_{i=1}^{m} x_i x_i^\top - \mu\mu^\top$$

- Conditional distribution is Gaussian, too!

$$p(x_2|x_1) \propto \exp\left[-\frac{1}{2}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^\top \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{22} \end{bmatrix}^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}\right]$$

Gaussian Processes

Gaussian Process

Key Idea

Instead of a fixed set of random variables t, t' we assume a stochastic process $t : \mathcal{X} \to \mathbb{R}$, e.g. $\mathcal{X} = \mathbb{R}^n$.

Previously we had $\mathcal{X} = \{\text{age}, \text{height}, \text{weight}, \ldots\}$.

Definition of a Gaussian Process

A stochastic process $t : \mathcal{X} \to \mathbb{R}$, where all $(t(x_1), \ldots, t(x_m))$ are normally distributed.

Parameters of a GP

Mean $\mu(x) := \mathbf{E}[t(x)]$

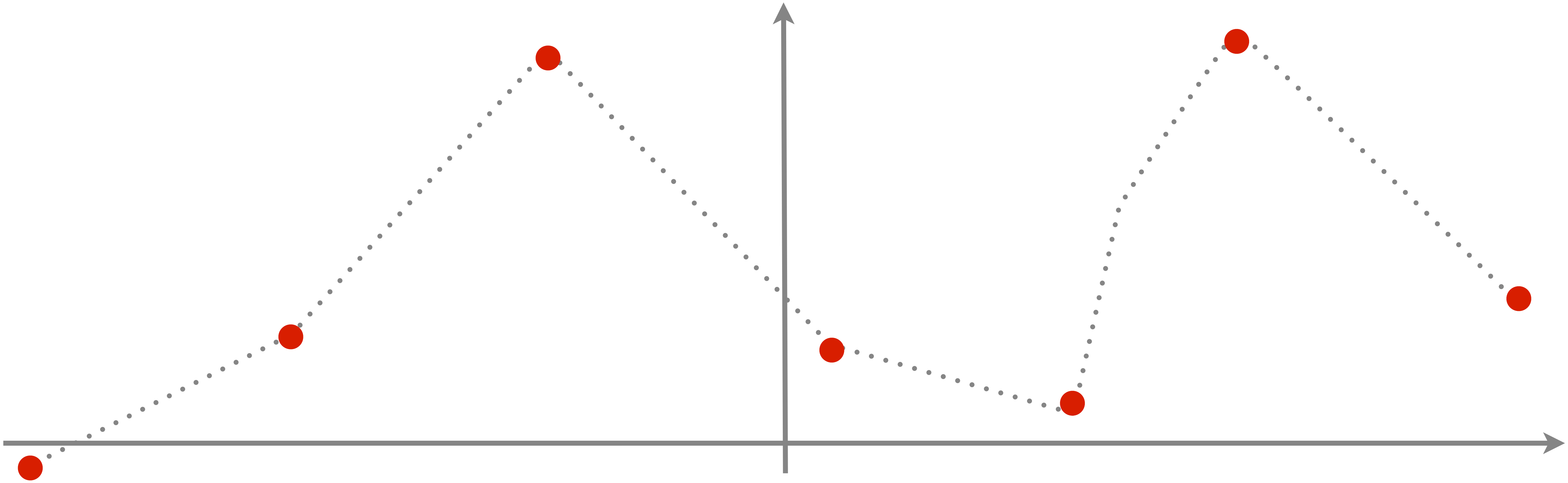
Covariance Function $k(x, x') := \mathrm{Cov}(t(x), t(x'))$

Simplifying Assumption

We assume knowledge of $k(x, x')$ and set $\mu = 0$.

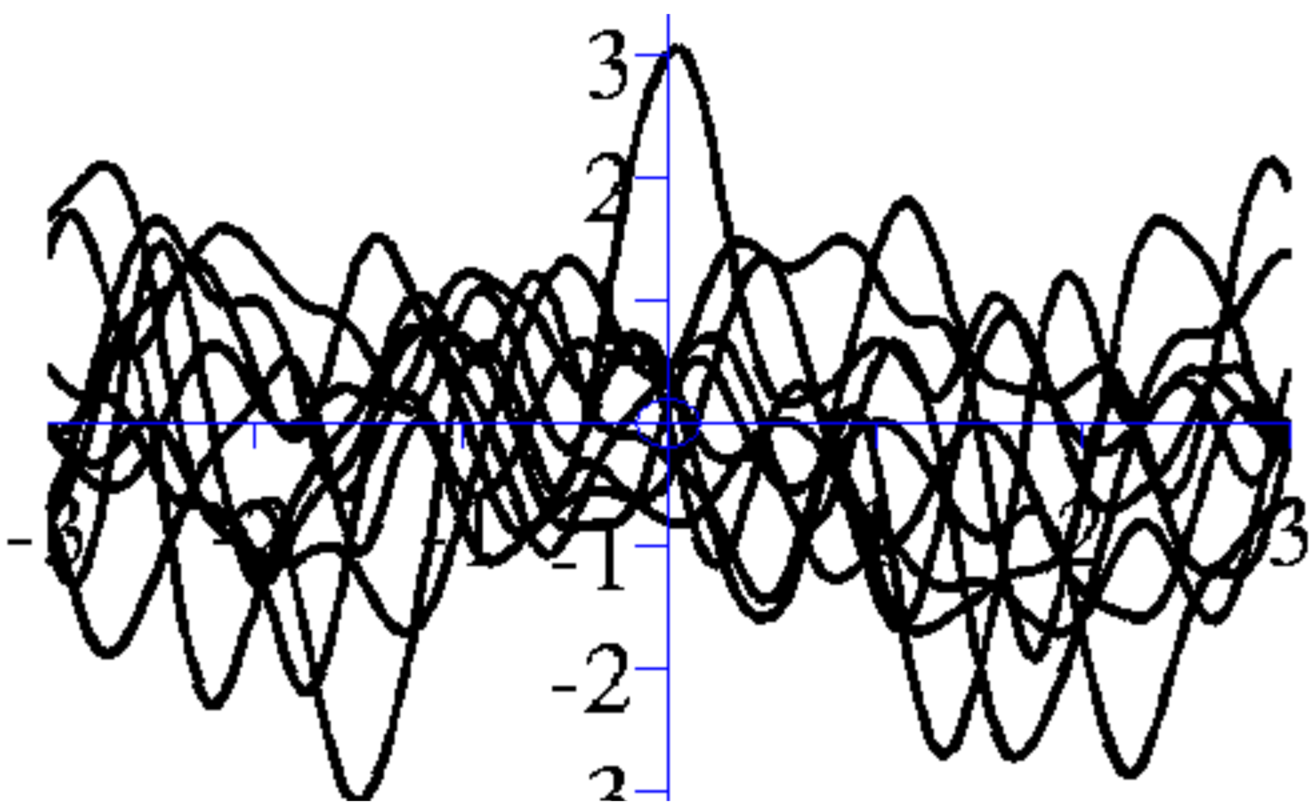
Gaussian Process

- Sampling from a Gaussian Process
  - Points x where we want to sample
  - Compute covariance matrix X
  - Can only obtain values at those points!
  - In general entire function f(x) is NOT available



Gaussian Process

- Sampling from a Gaussian Process
  - Points x where we want to sample
  - Compute covariance matrix X
  - Can only obtain values at those points!
  - In general entire function f(x) is NOT available



only looks smooth
(evaluated at many points)

Gaussian Process

- Sampling from a Gaussian Process
  - Points x where we want to sample
  - Compute covariance matrix X
  - Can only obtain values at those points!
  - In general entire function f(x) is NOT available

$$p(t|X) = (2\pi)^{-\frac{m}{2}} |K|^{-1} \exp\left(-\frac{1}{2}(t-\mu)^\top K^{-1}(t-\mu)\right)$$

$$\text{where } K_{ij} = k(x_i, x_j) \text{ and } \mu_i = \mu(x_i)$$

Kernels ...

Covariance Function

- Function of two arguments
- Leads to matrix with nonnegative eigenvalues
- Describes correlation between pairs of observations

Kernel

- Function of two arguments
- Leads to matrix with nonnegative eigenvalues
- Similarity measure between pairs of observations

Lucky Guess

- We suspect that kernels and covariance functions are the same . . .

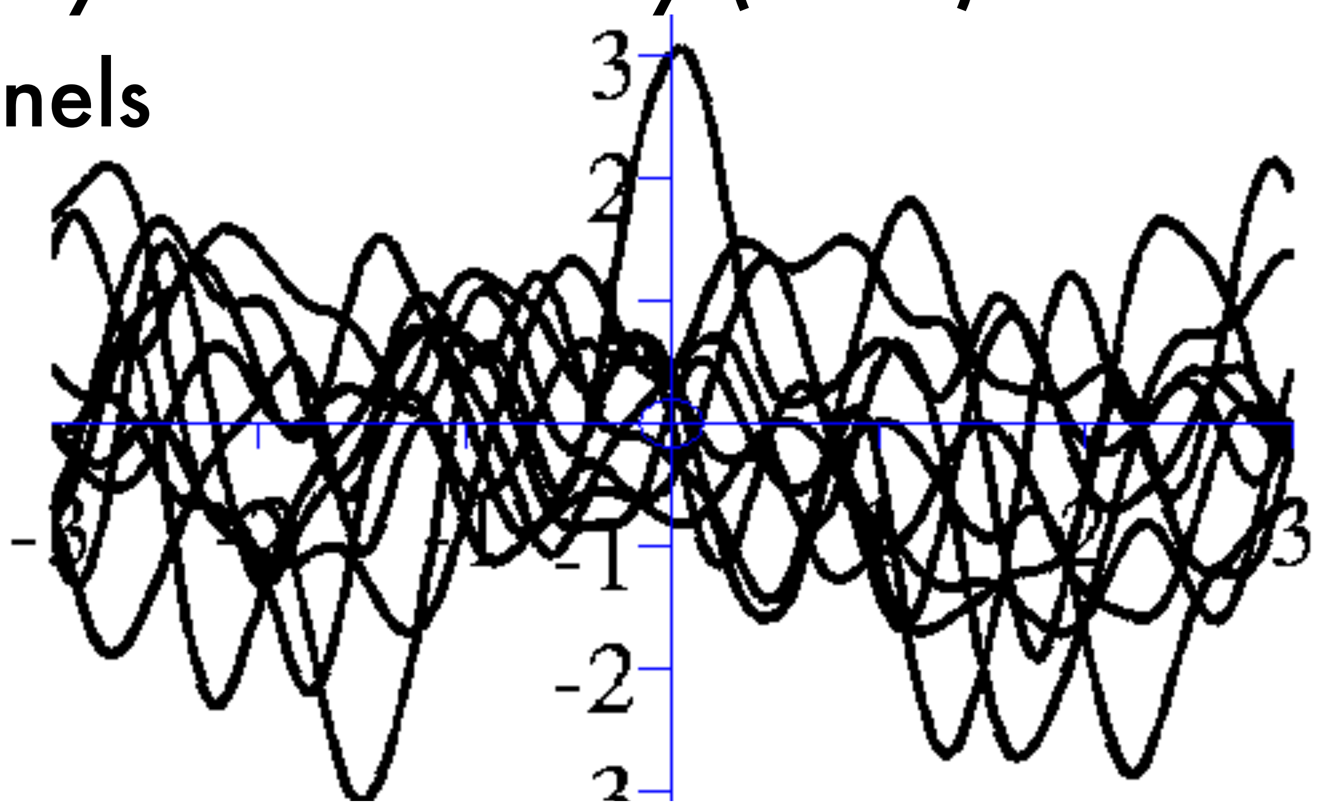
yes!

Mini Summary

- Gaussian Process
  - Think distribution over function values (not functions)
  - Defined by mean and covariance function

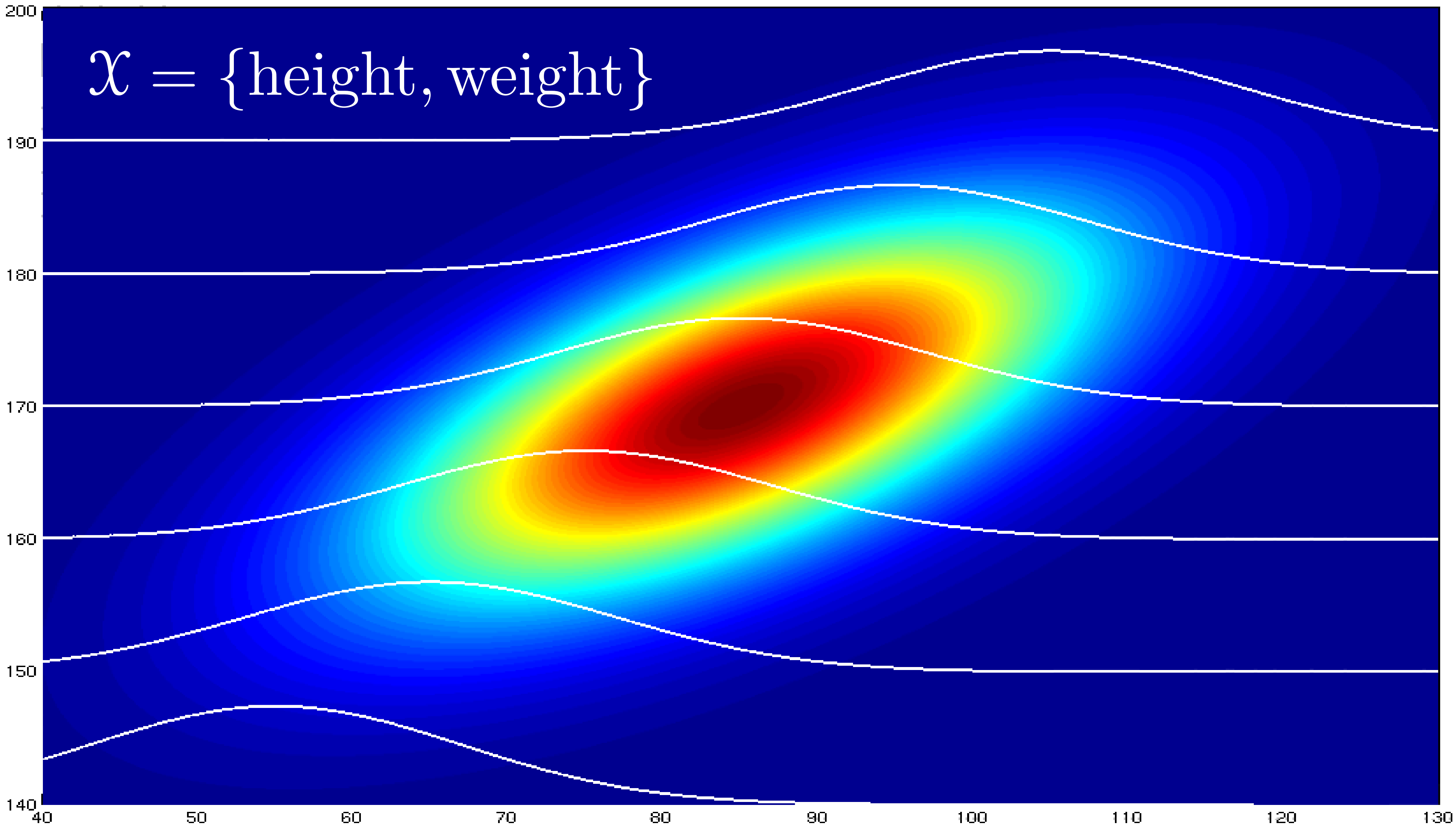
$$p(t|X) = (2\pi)^{-\frac{m}{2}} |K|^{-1} \exp\left(-\frac{1}{2}(t-\mu)^\top K^{-1}(t-\mu)\right)$$

  - Generates vectors of arbitrary dimensionality (via X)
  - Covariance function via kernels



Gaussian Process Regression

Gaussian Processes for Inference



$$p(\text{weight}|\text{height}) = \frac{p(\text{height}, \text{weight})}{p(\text{height})} \propto p(\text{height}, \text{weight})$$

Carnegie Mellon University

Joint Gaussian Model

- Random variables (t,t′) are drawn from GP

$$p(t,t') \propto \exp\left(-\frac{1}{2}\begin{bmatrix} t-\mu \\ t'-\mu' \end{bmatrix}^\top \begin{bmatrix} K_{tt} & K_{tt'} \\ K_{tt'}^\top & K_{t't'} \end{bmatrix}^{-1} \begin{bmatrix} t-\mu \\ t'-\mu' \end{bmatrix}\right)$$

- Observe subset t
- Predict t′ using

$$\tilde{K} = K_{t't'} - K_{tt'}^\top K_{tt}^{-1} K_{tt'} \text{ and } \tilde{\mu} = \mu' + K_{tt'}^\top \boxed{\left[K_{tt}^{-1}(t-\mu)\right]}$$

- Linear expansion (precompute things)
- Predictive uncertainty is data independent
  Good for experimental design
- Predictive uncertainty is data independent

Linear Gaussian Process Regression

Linear kernel: $k(x, x') = \langle x, x' \rangle$

- Kernel matrix $X^\top X$
- Mean and covariance

$$\tilde{K} = X'^\top X' - X'^\top X (X^\top X)^{-1} X^\top X' = X'^\top (\mathbf{1} - P_X) X'.$$

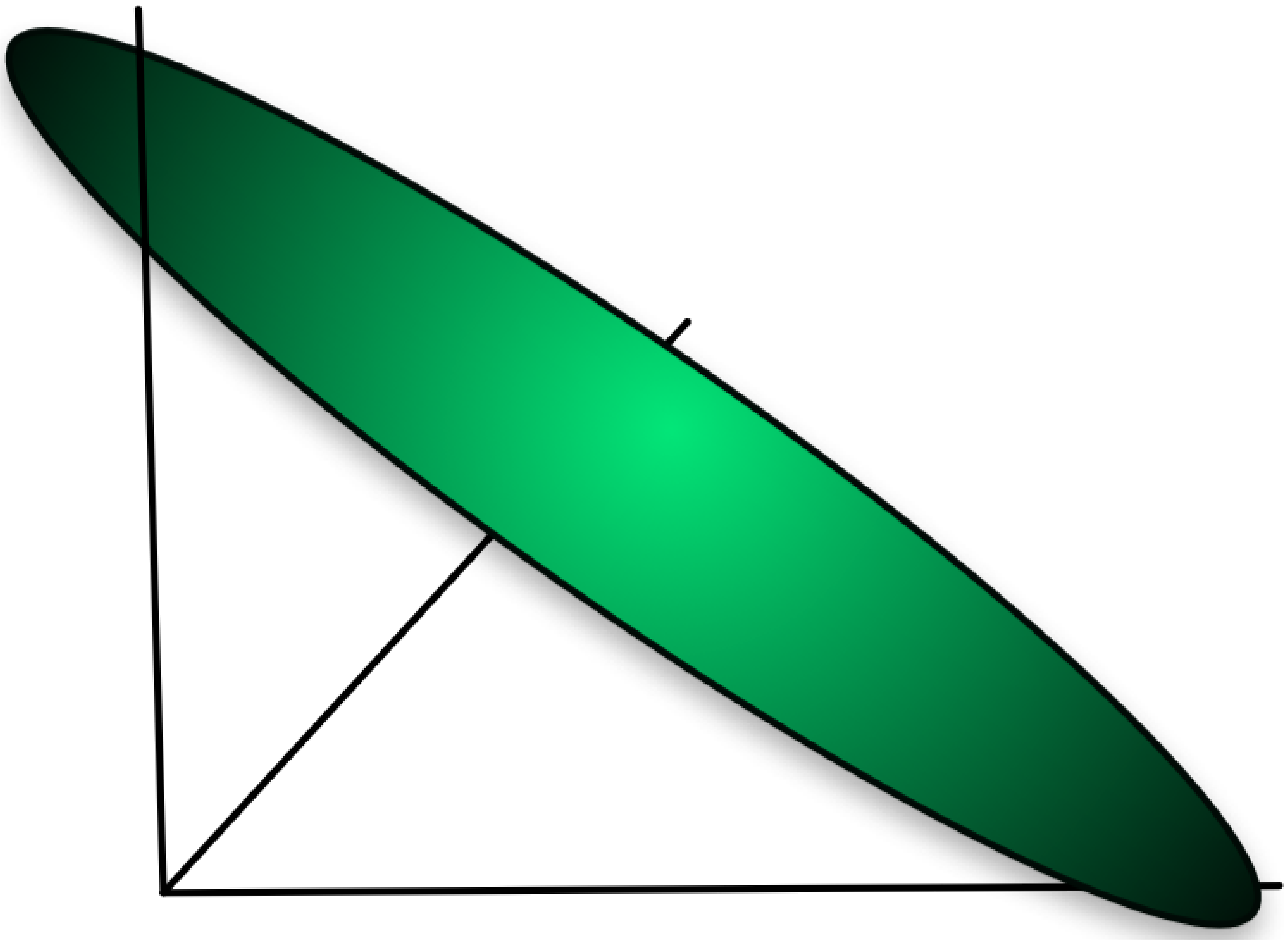
$$\tilde{\mu} = X'^\top \left[X (X^\top X)^{-1} t \right]$$

- $\tilde{\mu}$ is a **linear function of** X'.

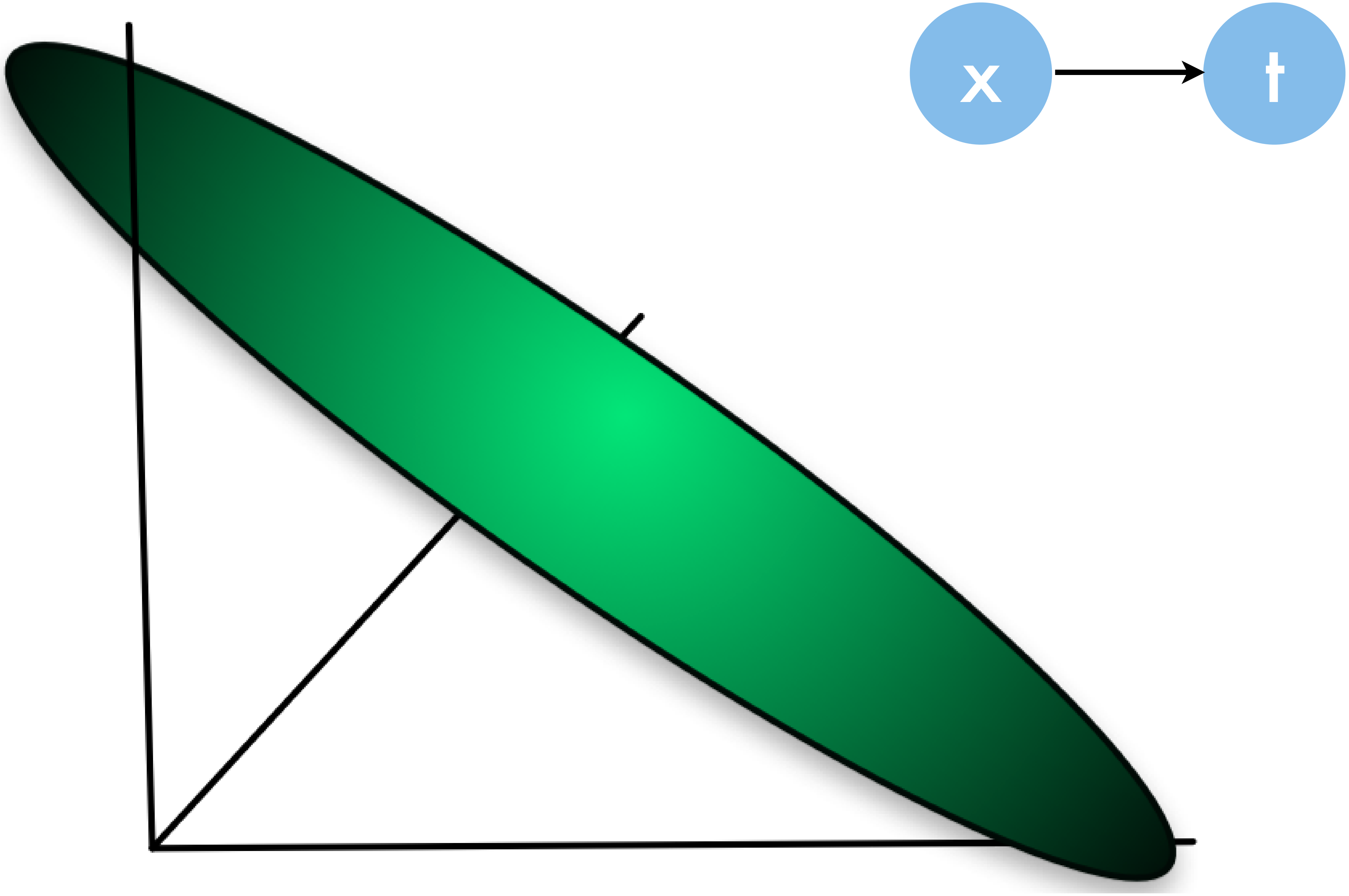
Problem

- The covariance matrix $X^\top X$ has at most rank n.
- After n observations ($x \in \mathbb{R}^n$) the **variance vanishes**. This is **not realistic**.
- "Flat pancake" or "cigar" distribution.

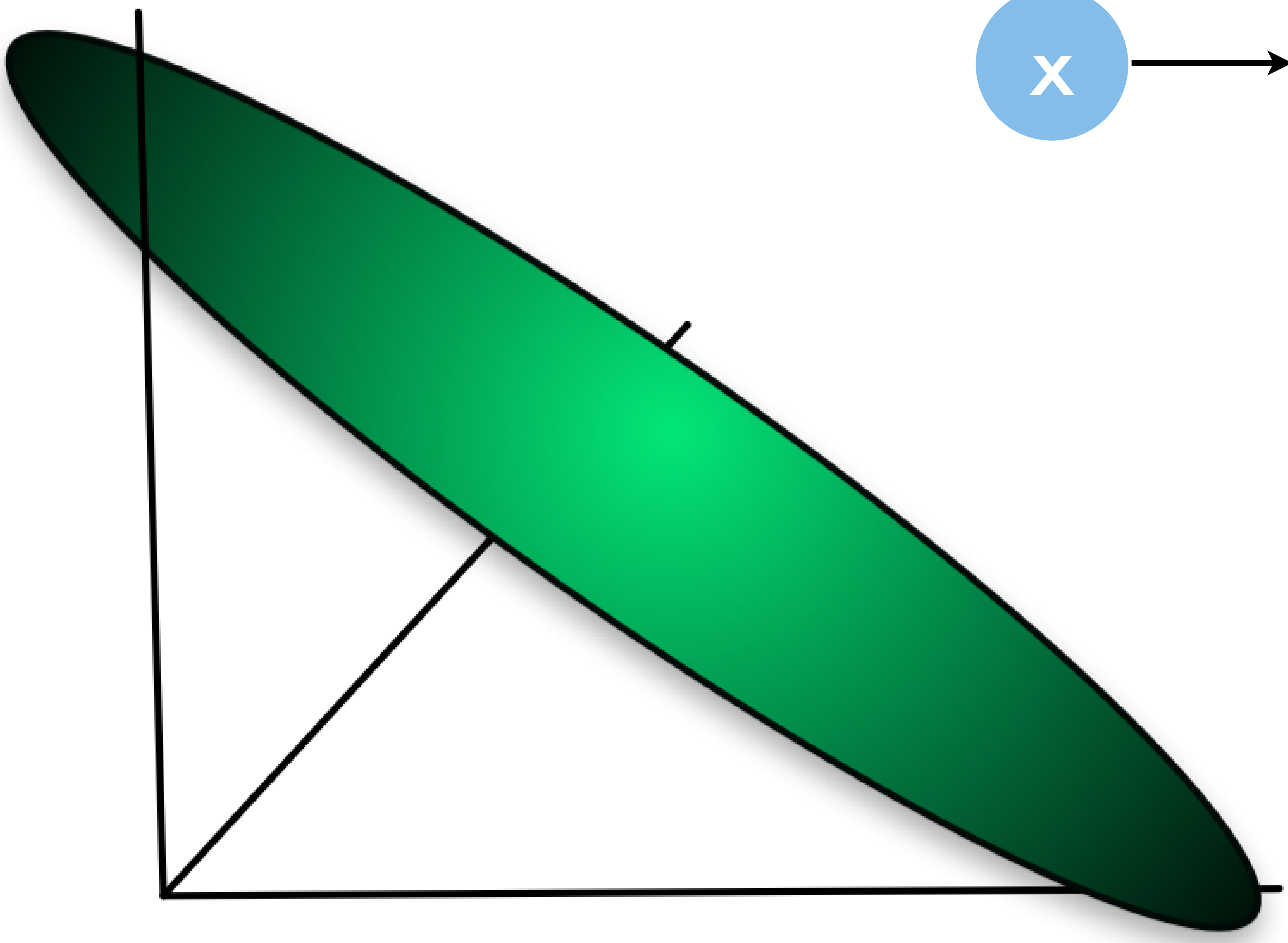
Degenerate Covariance

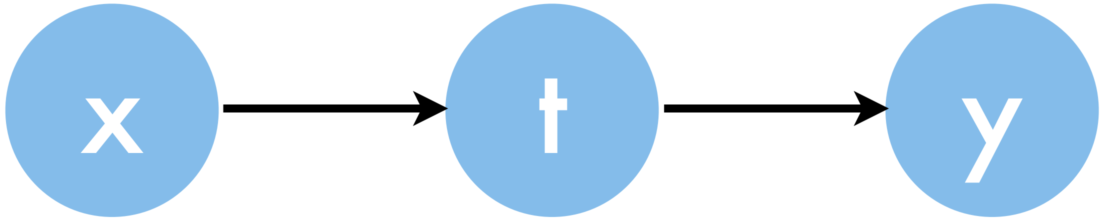


Degenerate Covariance



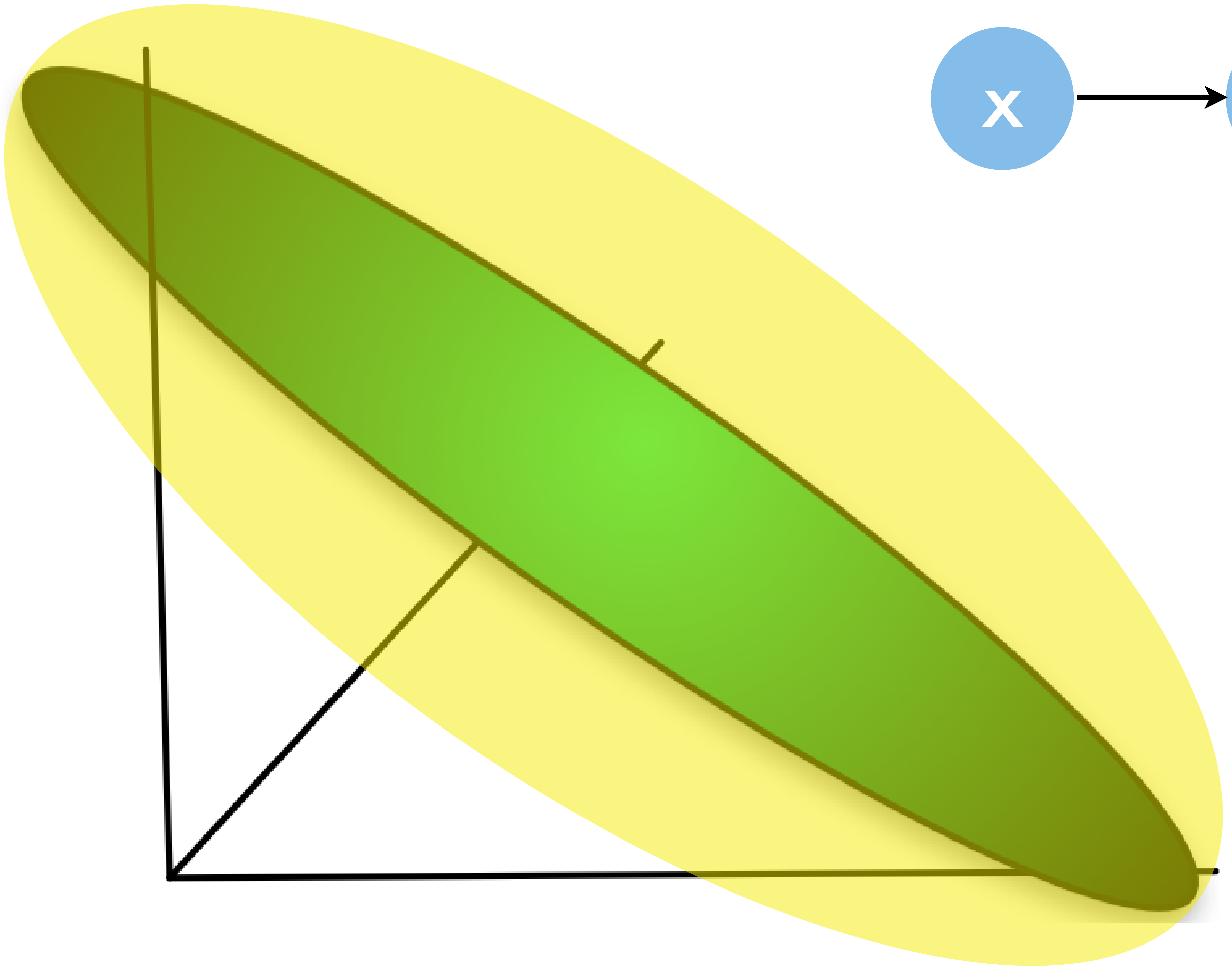
Degenerate Covariance

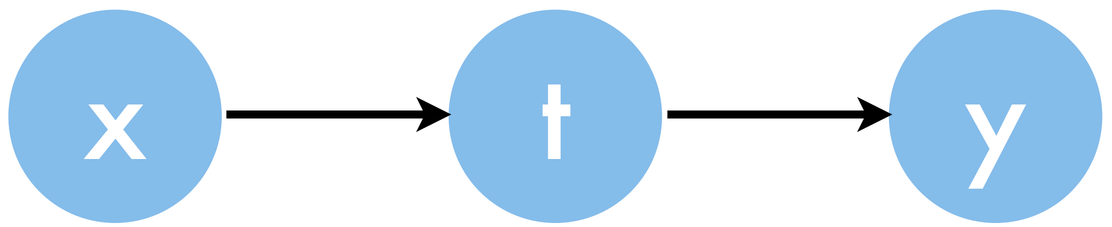




'fatten up' covariance

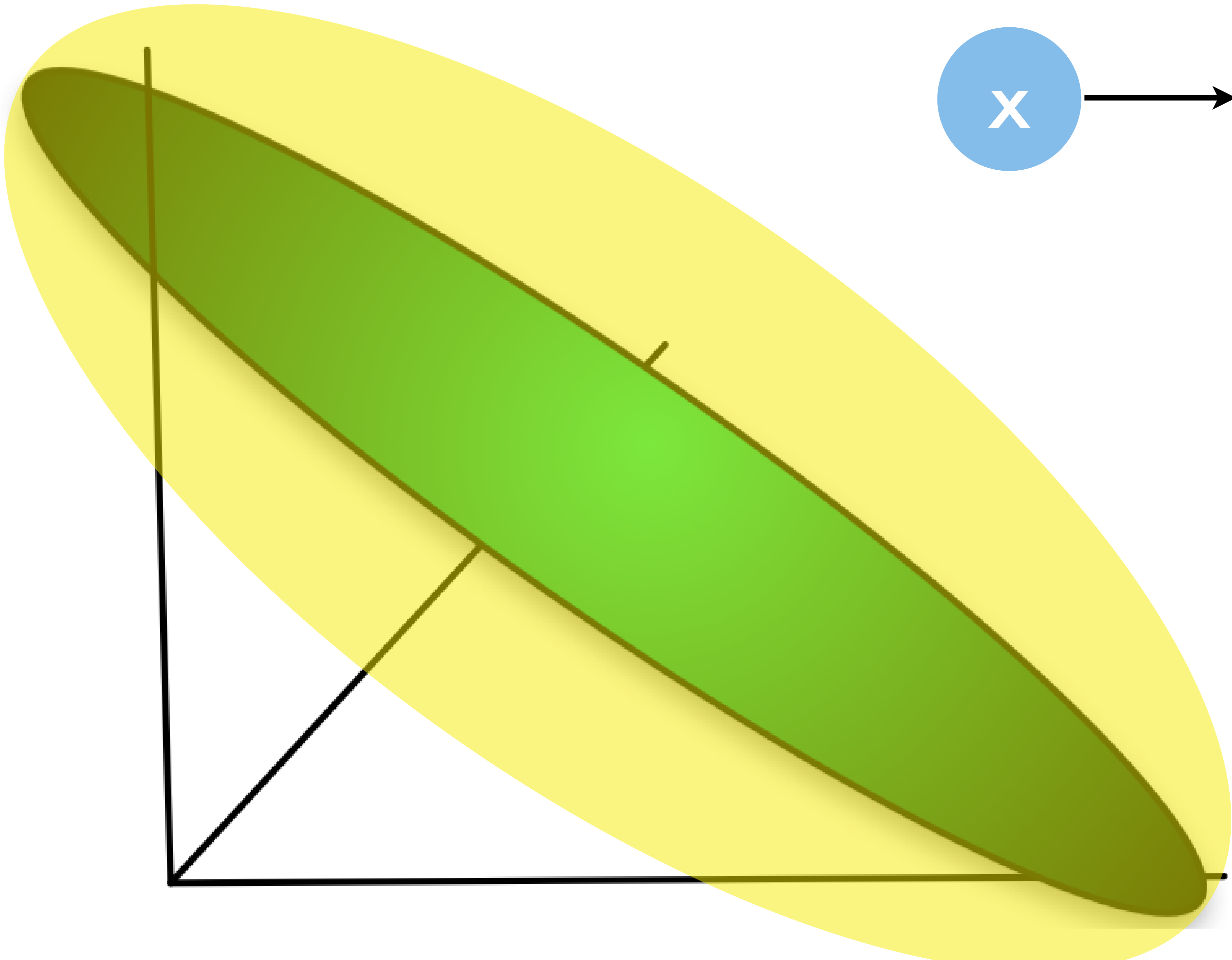
Degenerate Covariance

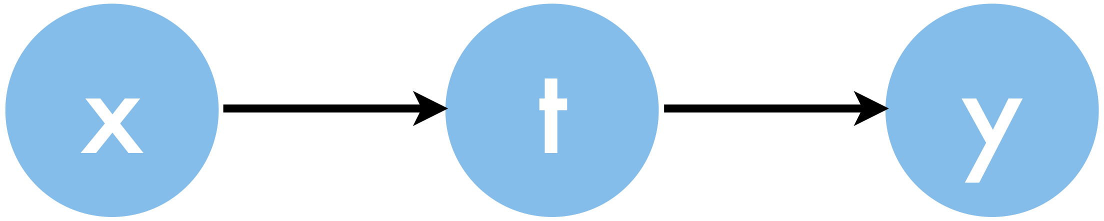




'fatten up' covariance

Degenerate Covariance





'fatten up' covariance

$$t \sim \mathcal{N}(\mu, K)$$

$$y_i \sim \mathcal{N}(t_i, \sigma^2)$$

Additive Noise

Indirect Model

Instead of observing $t(x)$ we observe $y = t(x) + \xi$, where ξ is a nuisance term. This yields

$$p(Y|X) = \int \prod_{i=1}^{m} p(y_i|t_i) p(t|X) dt$$

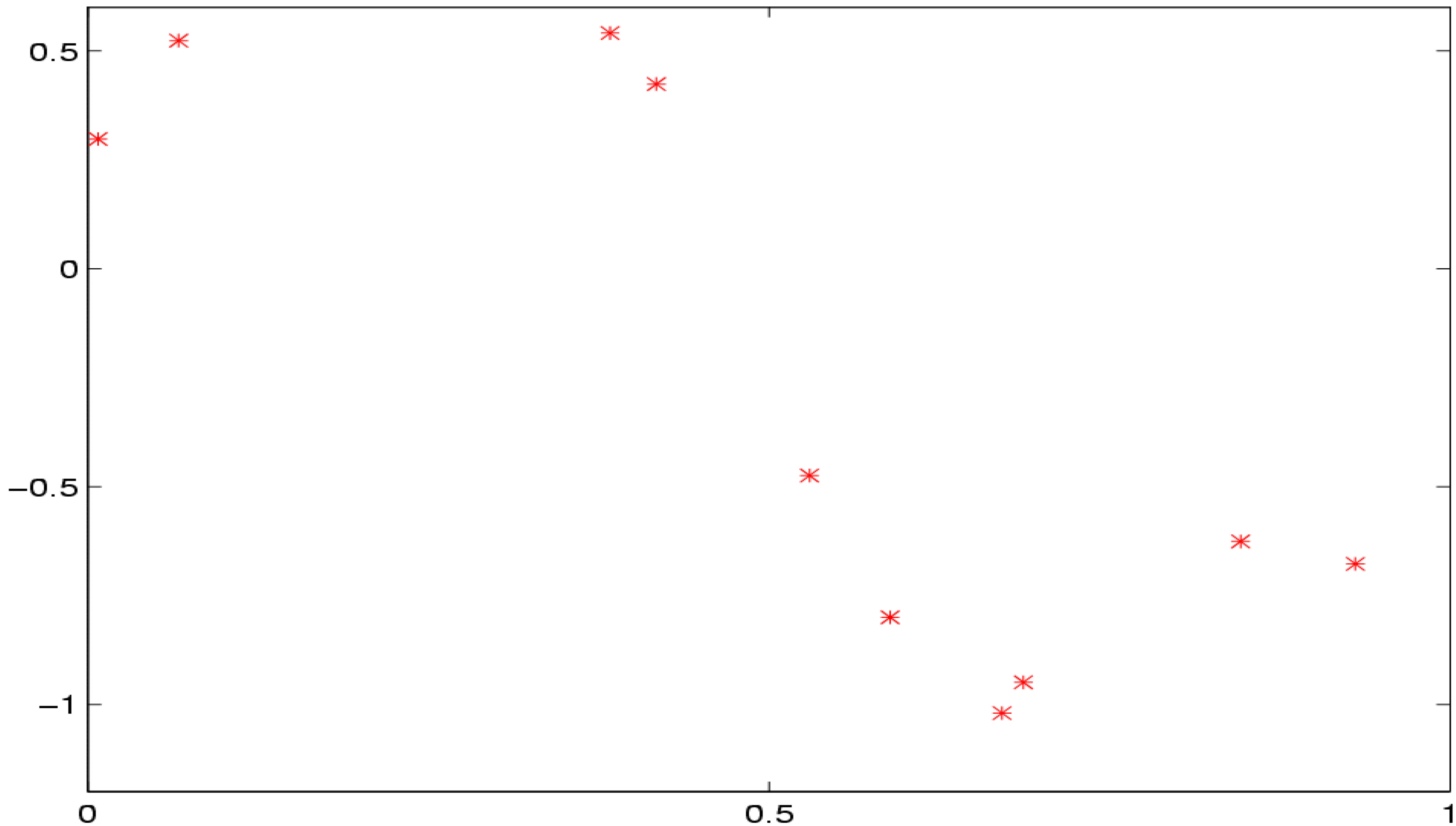
where we can now find a maximum a posteriori solution for t by maximizing the integrand (we will use this later).

Additive Normal Noise

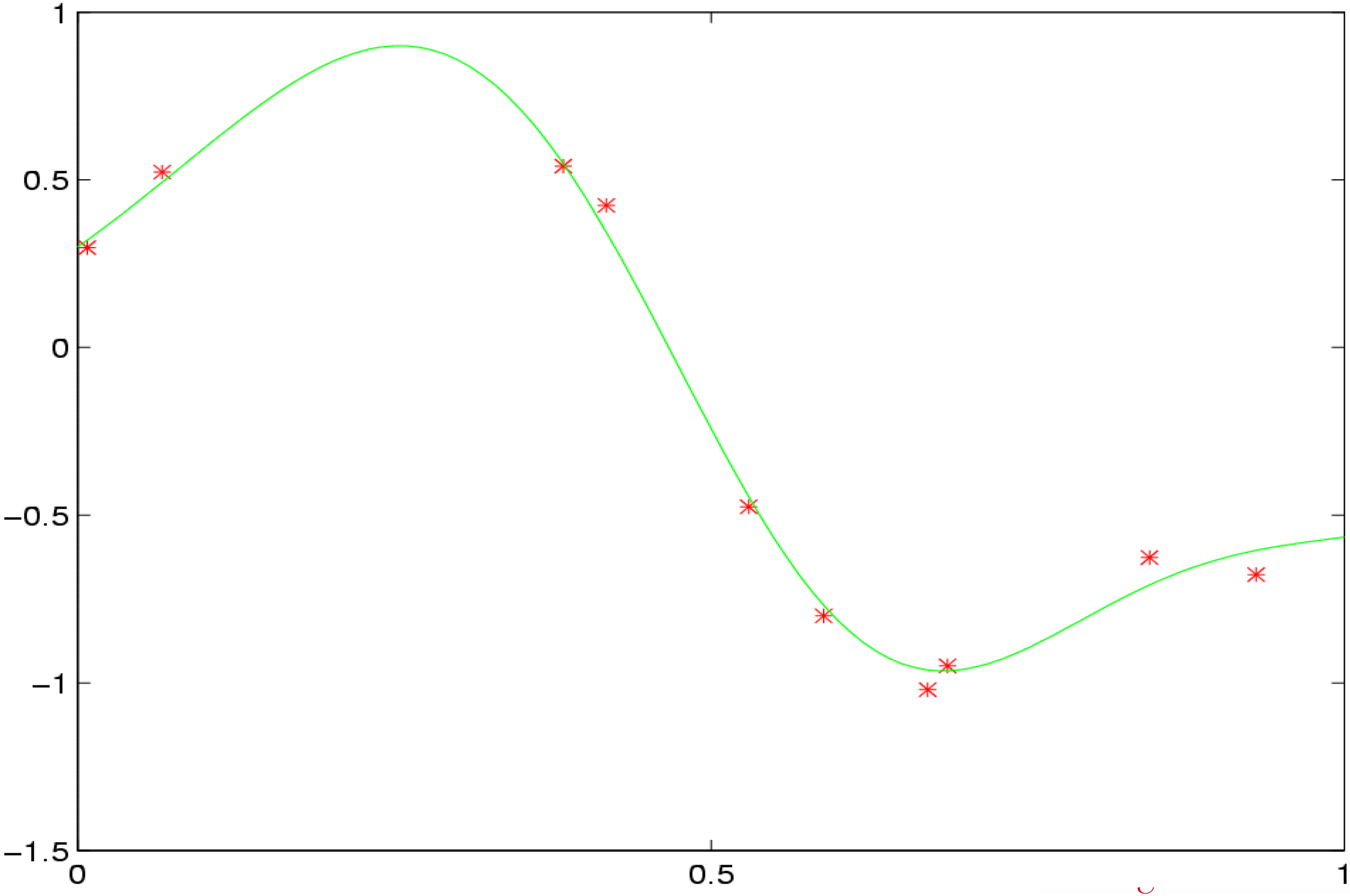
- If $\xi \sim \mathcal{N}(0, \sigma^2)$ then y is the sum of two Gaussian random variables.
- Means and variances **add up**.

$$y \sim \mathcal{N}(\mu, K + \sigma^2 \mathbf{1}).$$

Data

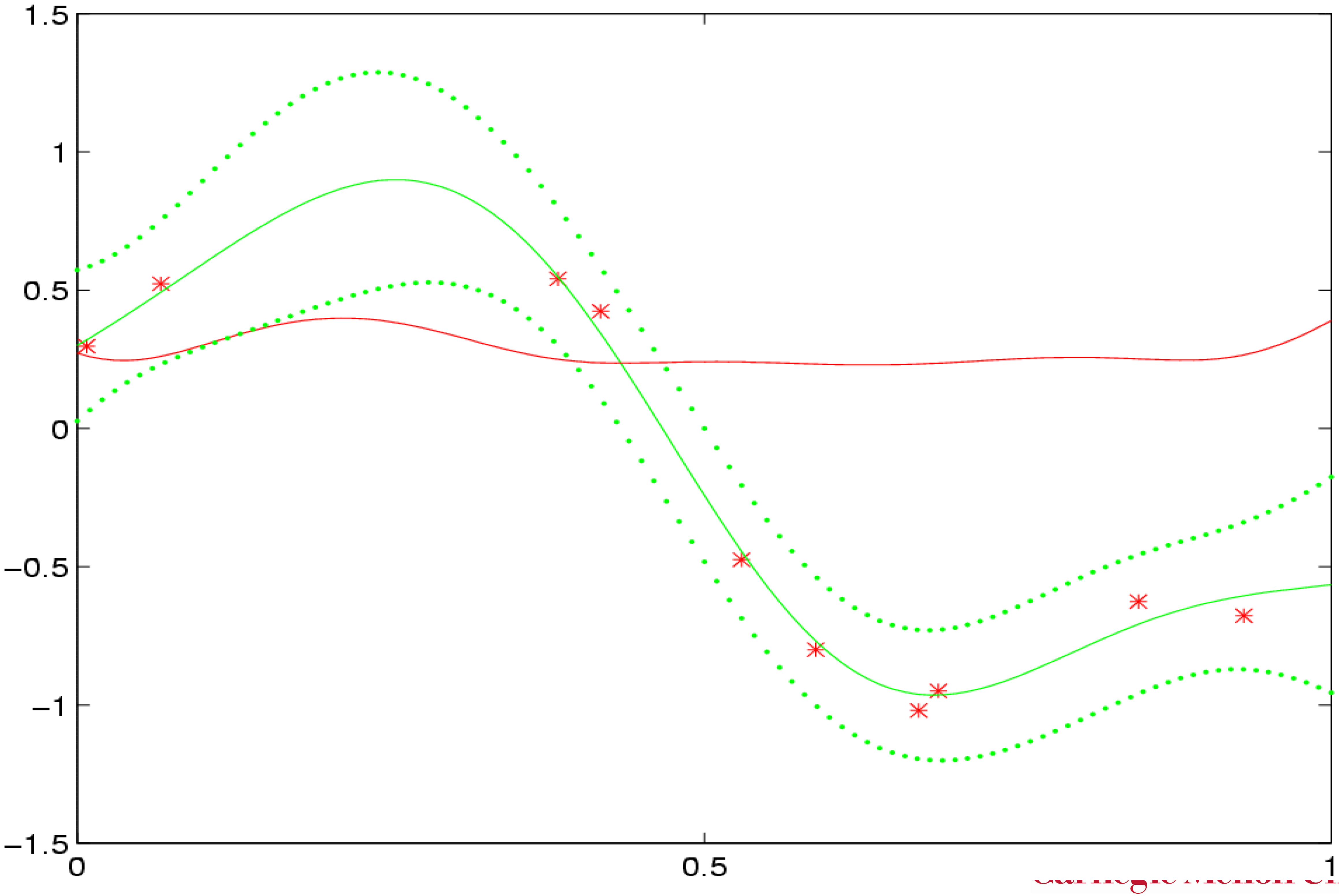


Predictive mean $k(x, X)^\top (K(X, X) + \sigma^2 \mathbf{1})^{-1} y$

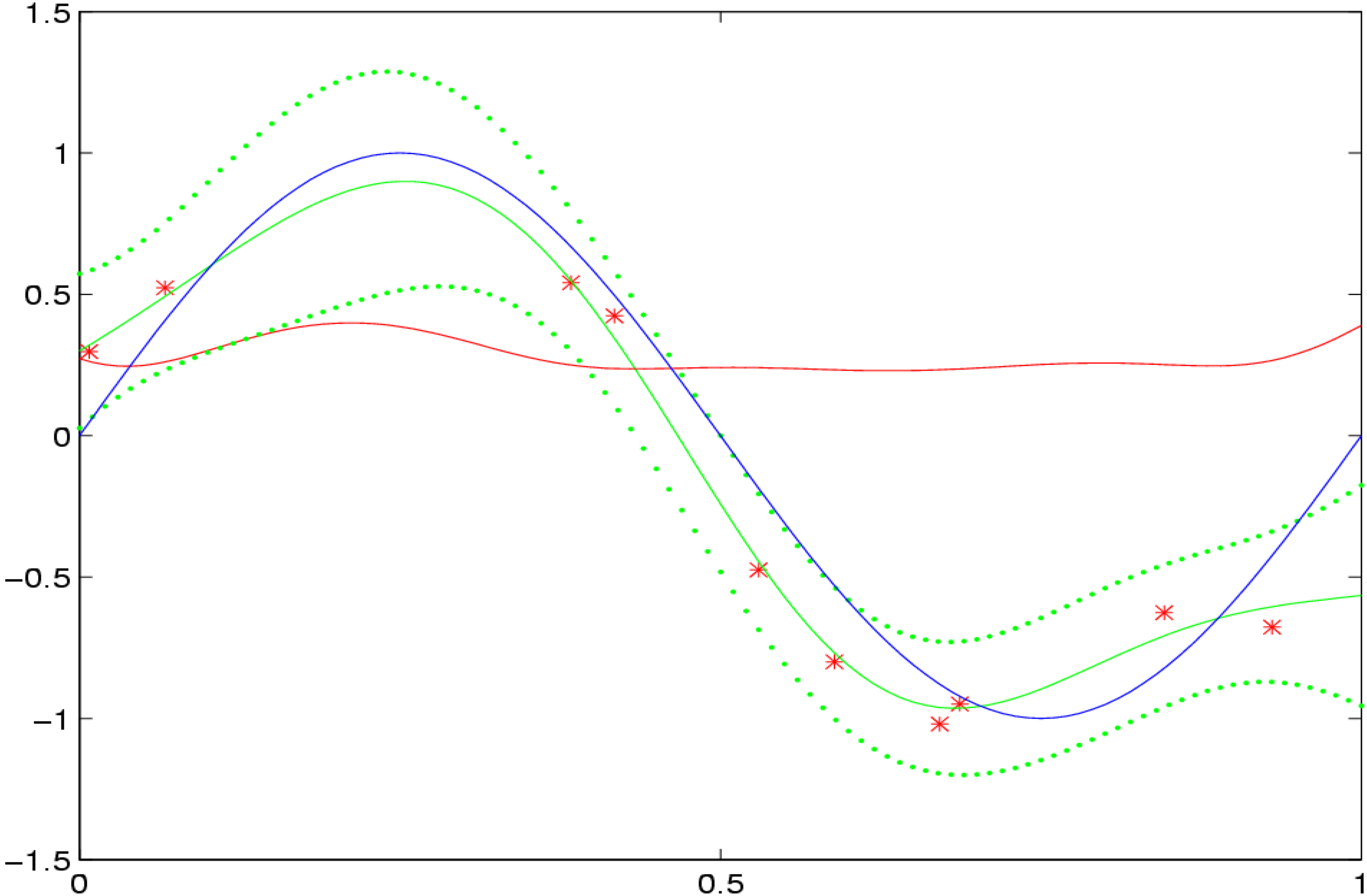


versity

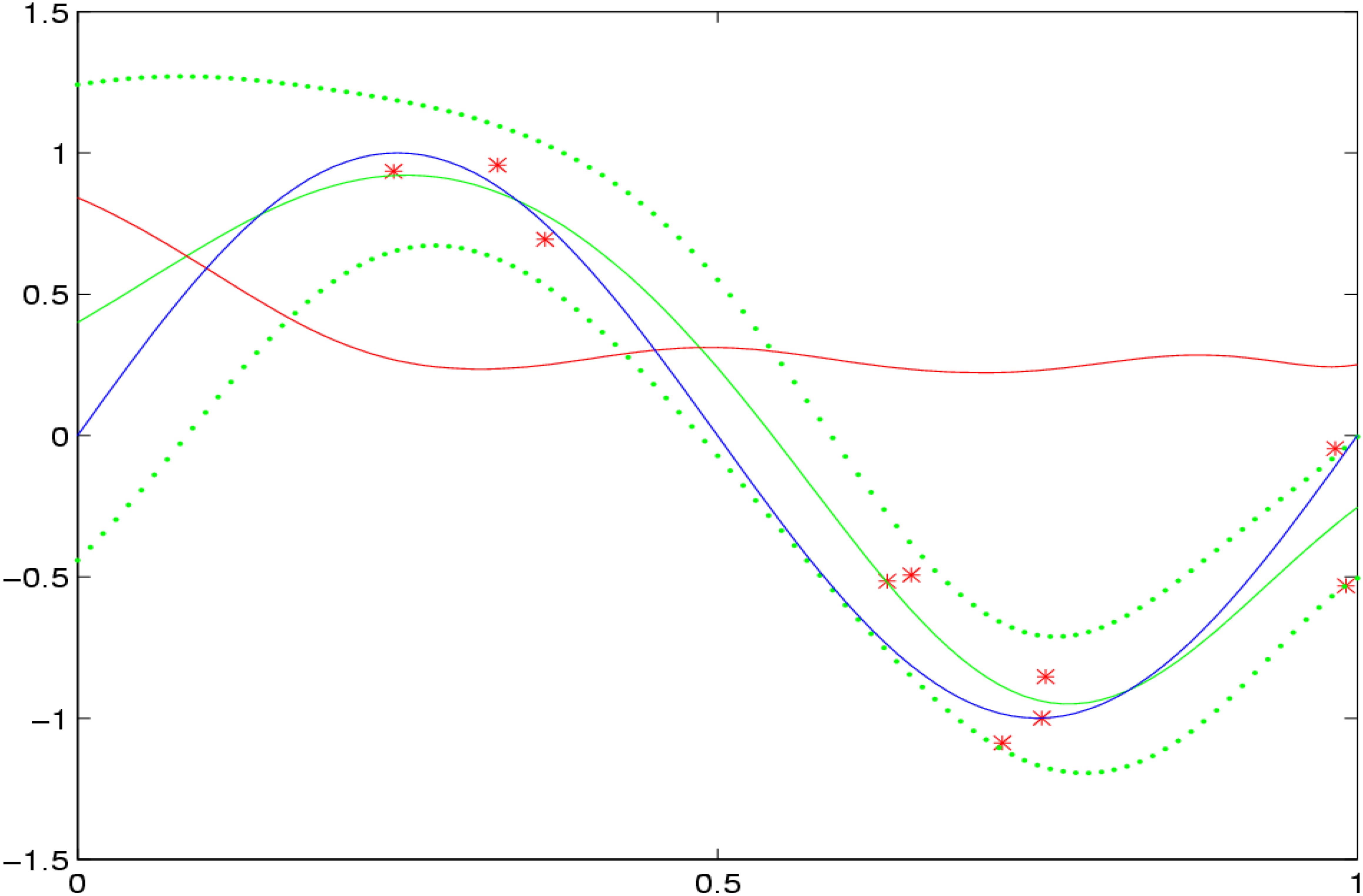
Variance



Putting it all together



Putting it all together



Ugly details

Covariance Matrices

- Additive noise

$$K = K_{\text{kernel}} + \sigma^2 \mathbf{1}$$

- Predictive mean and variance

$$\tilde{K} = K_{t't'} - K_{tt'}^{\top} K_{tt}^{-1} K_{tt'} \text{ and } \tilde{\mu} = K_{tt'}^{\top} K_{tt}^{-1} t$$

With Noise

$$\tilde{K} = K_{t't'} + \sigma^2 \mathbf{1} - K_{tt'}^{\top} \left(K_{tt} + \sigma^2 \mathbf{1} \right)^{-1} K_{tt'}$$

$$\text{and } \tilde{\mu} = \mu' + K_{tt'}^{\top} \left[\left(K_{tt} + \sigma^2 \mathbf{1} \right)^{-1} (y - \mu) \right]$$

Pseudocode

$$\tilde{K} = K_{t't'} + \sigma^2 \mathbf{1} - K_{tt'}^\top \left(K_{tt} + \sigma^2 \mathbf{1}\right)^{-1} K_{tt'}$$

$$\text{and } \tilde{\mu} = \mu' + K_{tt'}^\top \left[\left(K_{tt} + \sigma^2 \mathbf{1}\right)^{-1} (y - \mu)\right]$$

```
ktrtr   = k(xtrain,xtrain) + sigma2 * eye(mtr)
ktetr   = k(xtest,xtrain)
ktete   = k(xtest,xtest)

alpha   = ytr/ktrtr %better if you use cholesky
yte     = ktetr * alpha
sigmate = ktete + sigma2 * eye(mte) + ...
          ktetr * (ktetr/ktrtr)'
```

The connection between SVM and GP

Gaussian Process on Parameters

$$t \sim \mathcal{N}(\mu, K) \text{ where } K_{ij} = k(x_i, x_j)$$

Linear Model in Feature Space

$$t(x) = \langle \Phi(x), w \rangle + \mu(x) \text{ where } w \sim \mathcal{N}(0, \mathbf{1})$$

The covariance between $t(x)$ and $t(x')$ is then given by

$$\mathbf{E}_w \left[\langle \Phi(x), w \rangle \langle w, \Phi(x') \rangle \right] = \langle \Phi(x), \Phi(x') \rangle = k(x, x')$$

Linear model in feature space induces a Gaussian Process

Mini Summary

- Latent variables t drawn from a Gaussian Process
- Observations y are t corrupted with noise
- Observations y are drawn from Gaussian Process

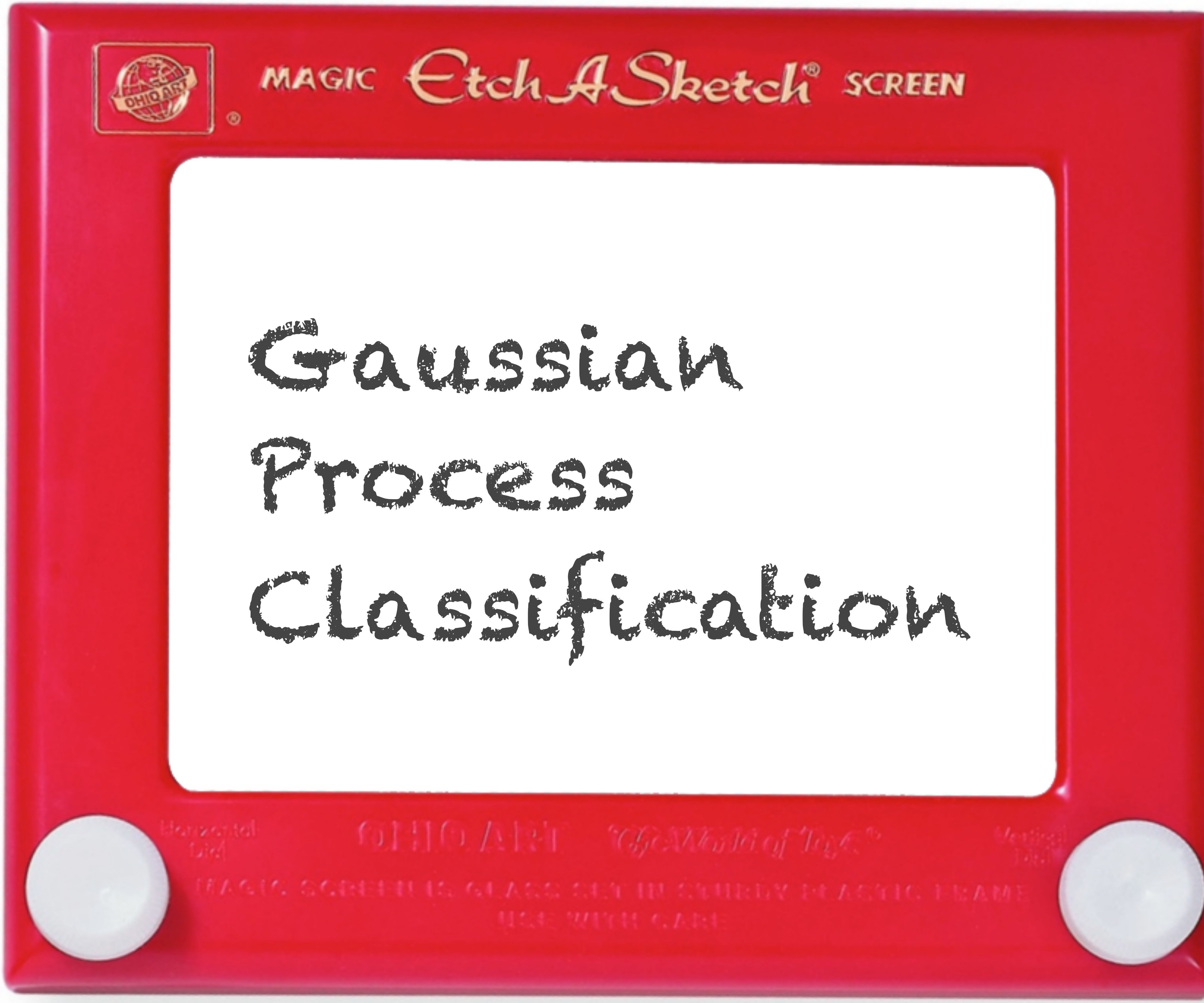
$$\mu \to \mu \text{ and } K \to K + \sigma^2 \mathbf{1}$$

- Estimate y'|y,x,x' (matrix inversion)

$$\tilde{K} = K_{t't'} + \sigma^2 \mathbf{1} - K_{tt'}^\top \left(K_{tt} + \sigma^2 \mathbf{1}\right)^{-1} K_{tt'}$$

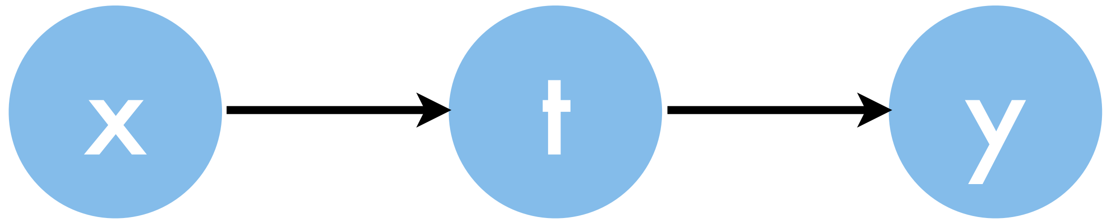
$$\text{and } \tilde{\mu} = \mu' + K_{tt'}^\top \left[\left(K_{tt} + \sigma^2 \mathbf{1}\right)^{-1} (y - \mu)\right]$$

- SVM kernel is GP kernel



Gaussian Process Classification

Gaussian Process Classification



- Regression
  - Data y is scalar
  - Connection to t is by additive noise

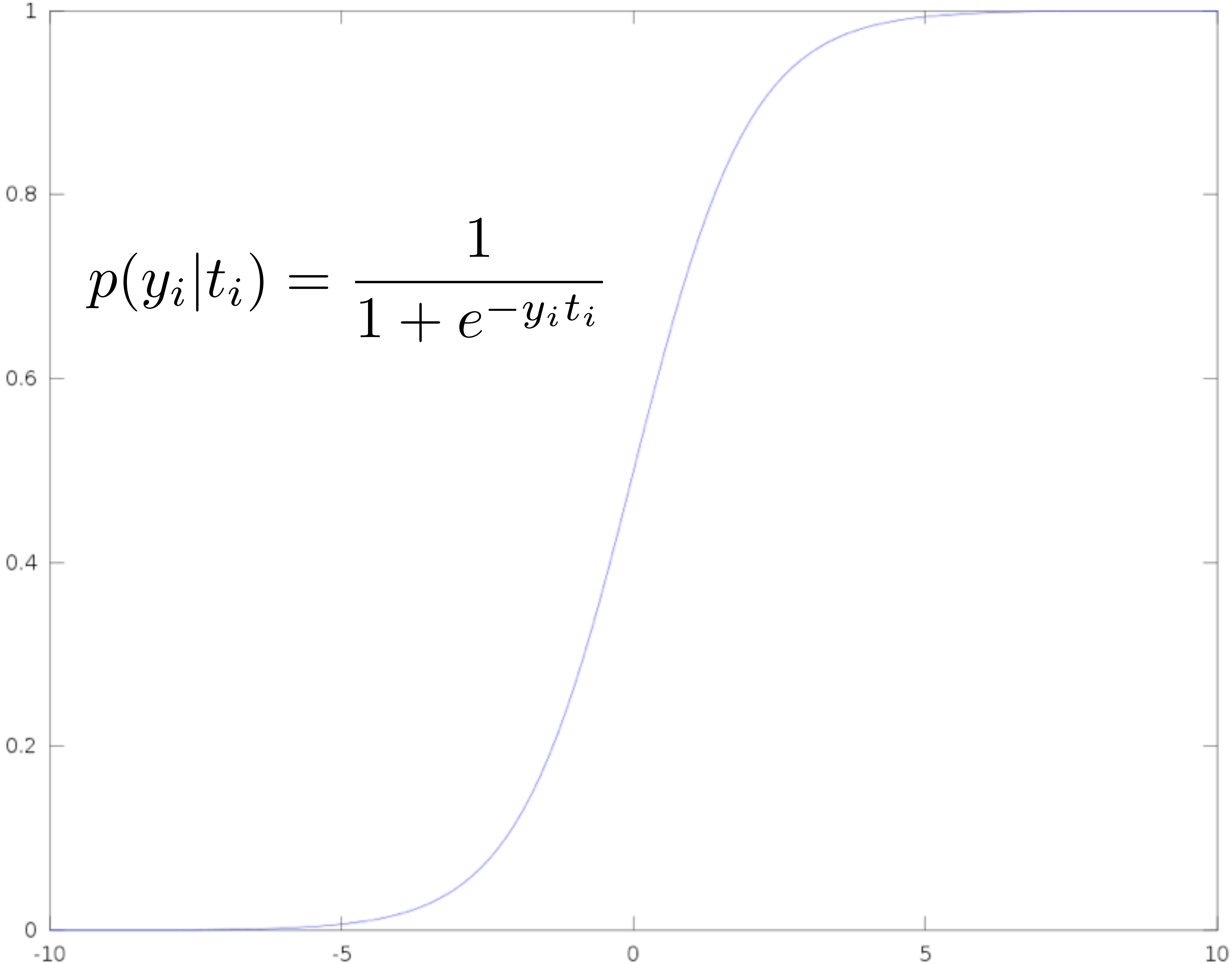
$$t \sim \mathcal{N}(\mu, K) \text{ and } y_i \sim \mathcal{N}(t_i, \sigma^2)$$

$$\text{i.e. } p(y_i|t_i) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} e^{-\frac{1}{2\sigma^2}(y_i - t_i)^2}$$

- (Binary) Classification
  - Data y in {-1, 1}
  - Connection to t is by logistic model

$$t \sim \mathcal{N}(\mu, K) \text{ and } p(y_i|t_i) = \frac{1}{1 + e^{-y_i t_i}}$$

Logistic function



$$p(y_i|t_i) = \frac{1}{1 + e^{-y_i t_i}}$$

Gaussian Process Classification

- Regression

$$t \sim \mathcal{N}(\mu, K) \text{ and } y_i \sim \mathcal{N}(t_i, \sigma^2) \text{ hence } y \sim \mathcal{N}(\mu, K + \sigma^2 \mathbf{1})$$

We can integrate out the latent variable t.

- Classification

Closed form solution is not possible

$$t \sim \mathcal{N}(\mu, K) \text{ and } y_i \sim \text{Logistic}(t_i)$$

(we cannot solve the integral in t).

$$p(t|y, x) \propto p(t|x) \prod_{i=1}^{m} p(y_i|t_i)$$

$$\propto \exp\left(-\frac{1}{2} t^\top K^{-1} t\right) \prod_{i=1}^{m} \frac{1}{1 + e^{-y_i t_i}}$$

Gaussian Process Classification

- What we should do: integrate out t,t′

$$p(y'|y, x, x') = \int d(t, t')p(y'|t')p(y|t)p(t, t'|x, x')$$

  But this is very very expensive (e.g. MCMC)

- Maximum a Posteriori approximation
  - Find $\hat{t} := \operatorname*{argmax}_{t} p(y|t)p(t|x)$
  - Ignore correlation in test data (horrible)
  - Find $\hat{t}'(x') := \operatorname*{argmax}_{t'} p(\hat{t}, t'|x, x')$
  - Estimate $y'|y, x, x' \sim \text{Logistic}(\hat{t}'(x'))$

Maximum a Posteriori Approximation

- Step 1 - maximize p(t|y,x)

$$\underset{t}{\text{minimize}}\ \frac{1}{2} t^\top K^{-1} t + \sum_{i=1}^{m} \log\left(1 + e^{-y_i t_i}\right)$$

- Step 2 - find t'|t for MAP estimate of t

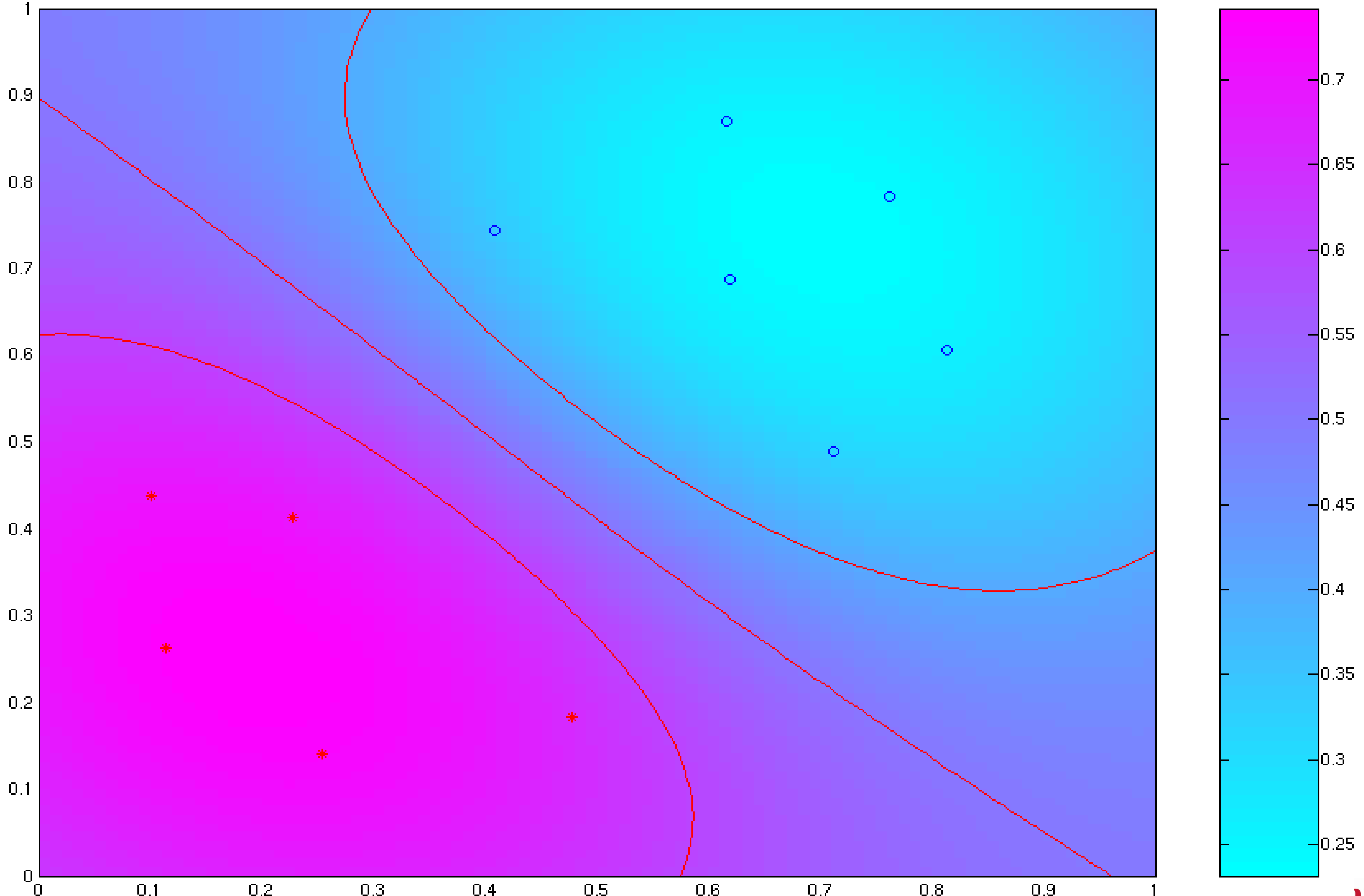
$$t' = K_{tt'}^\top K_{tt}^{-1} t$$

precompute

- Step 3 - estimate p(y'|t')

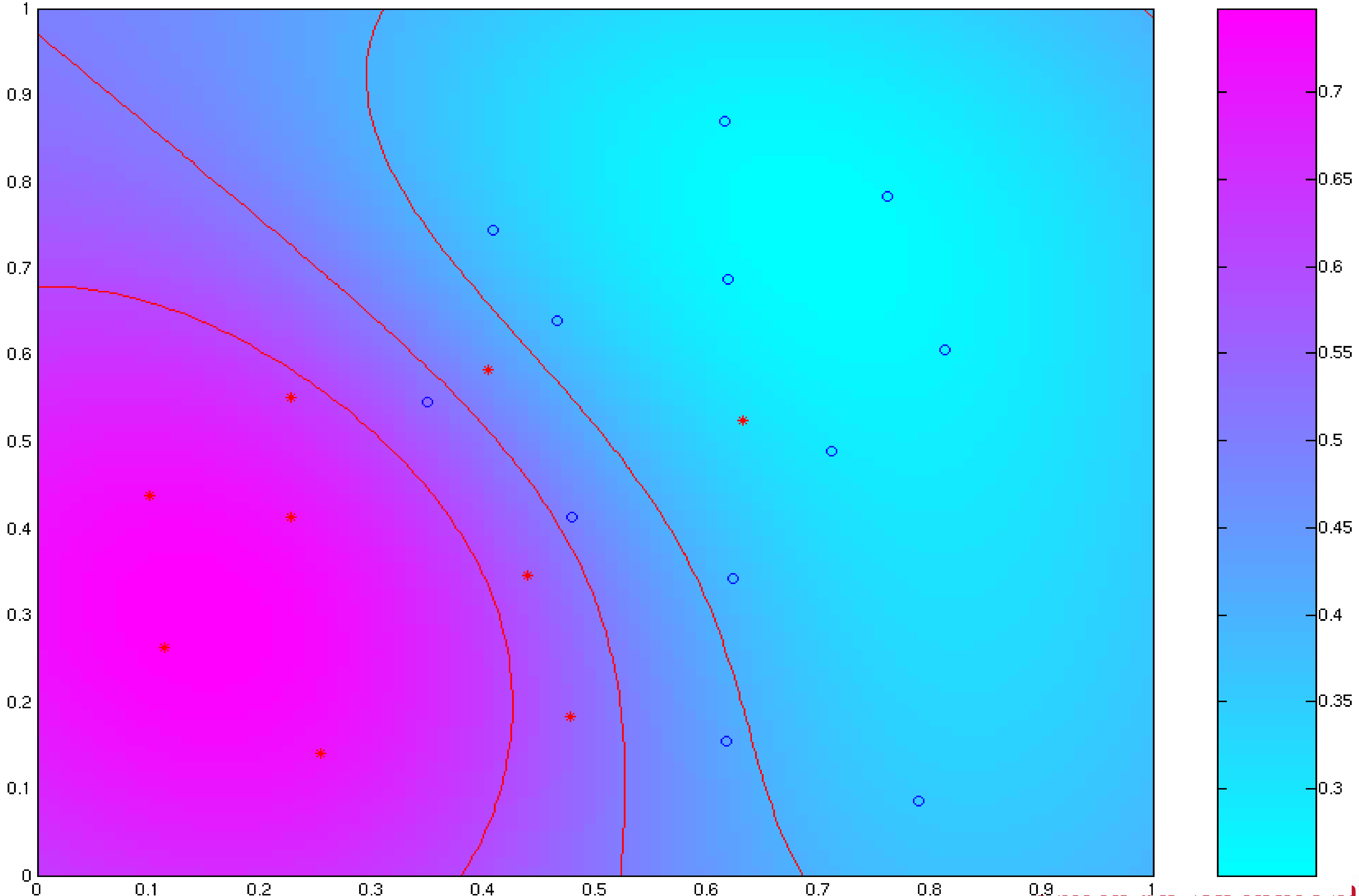
$$p(y'|t') = \frac{1}{1 + e^{-y' t'}}$$

Clean Data



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25

Noisy Data



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3

Connection to SVMs revisited

- **SVM objective**

$$\underset{\alpha}{\text{minimize}} \; \frac{1}{2}\alpha^\top K\alpha + \sum_{i=1}^{m} \max\left(0, 1 - y_i[K\alpha]_i\right)$$

- **Logistic regression objective (MAP estimation)**

$$\underset{t}{\text{minimize}} \; \frac{1}{2}t^\top K^{-1}t + \sum_{i=1}^{m} \log\left(1 + e^{-y_i t_i}\right)$$

- **Reparametrize** $\alpha = K^{-1}t$

$$\underset{\alpha}{\text{minimize}} \; \frac{1}{2}\alpha^\top K\alpha + \sum_{i=1}^{m} \log\left(1 + \exp y_i[K\alpha]_i\right)$$

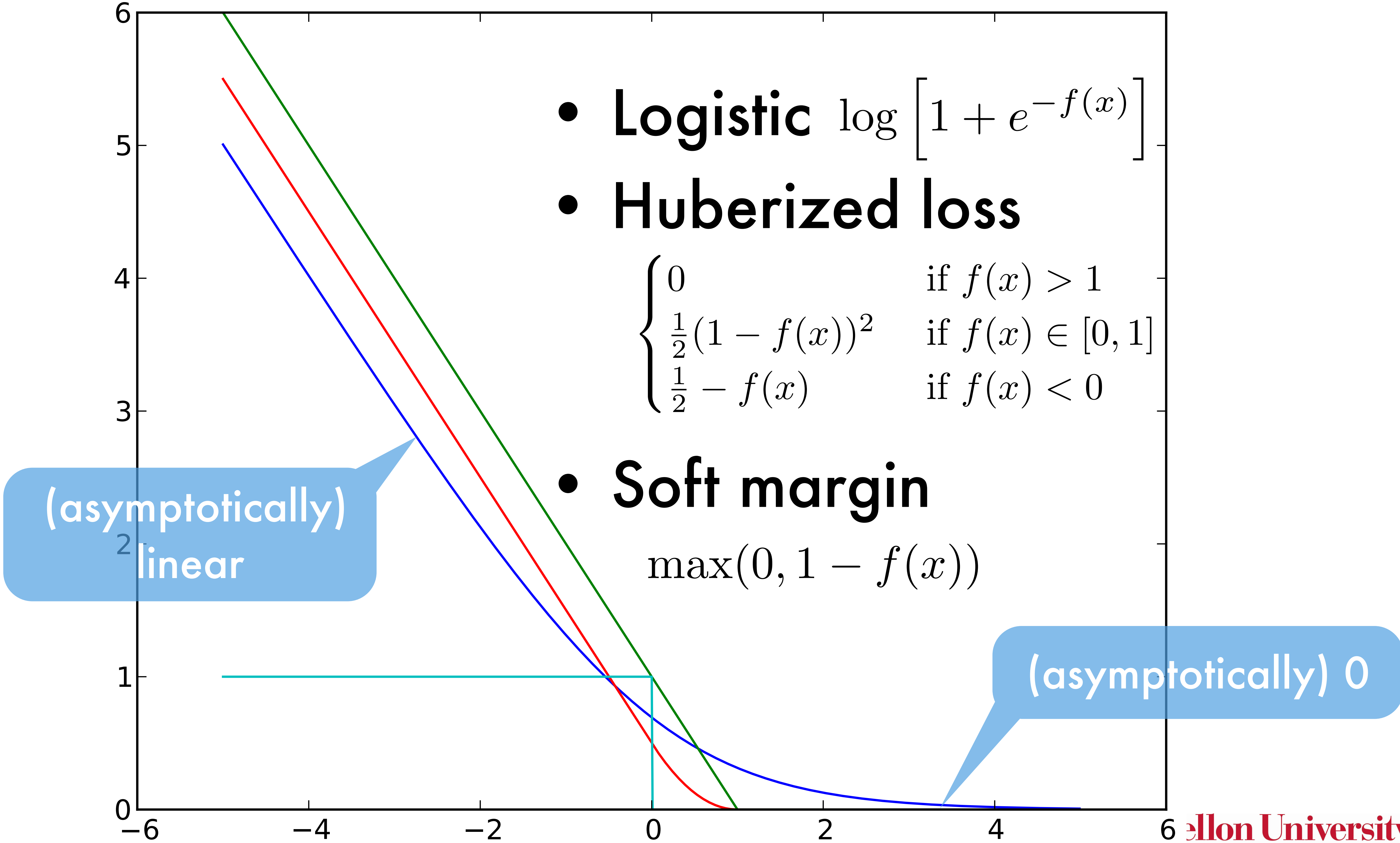
More loss functions

- **Logistic** $\log\left[1 + e^{-f(x)}\right]$
- **Huberized loss**

$$\begin{cases} 0 & \text{if } f(x) > 1 \\ \frac{1}{2}(1 - f(x))^2 & \text{if } f(x) \in [0, 1] \\ \frac{1}{2} - f(x) & \text{if } f(x) < 0 \end{cases}$$

- **Soft margin**

$$\max(0, 1 - f(x))$$



Mini Summary

- Latent variables drawn from Gaussian Process
- Observation drawn from logistic model
  - Impossible to integrate out latent variables
  - Maximum a posteriori inference
    (with many hacks to make it scale)
- Optimization problem is similar to SVM
  (different loss and parametrization $\alpha = K^{-1}t$)
- Advanced topic - adjusting K via prior on k