MACHINE LEARNING
DEPARTMENT

6.1 Gaussians

6 Bayesian Kernel Methods

Alexander Smola

Introduction to Machine Learning 10-701

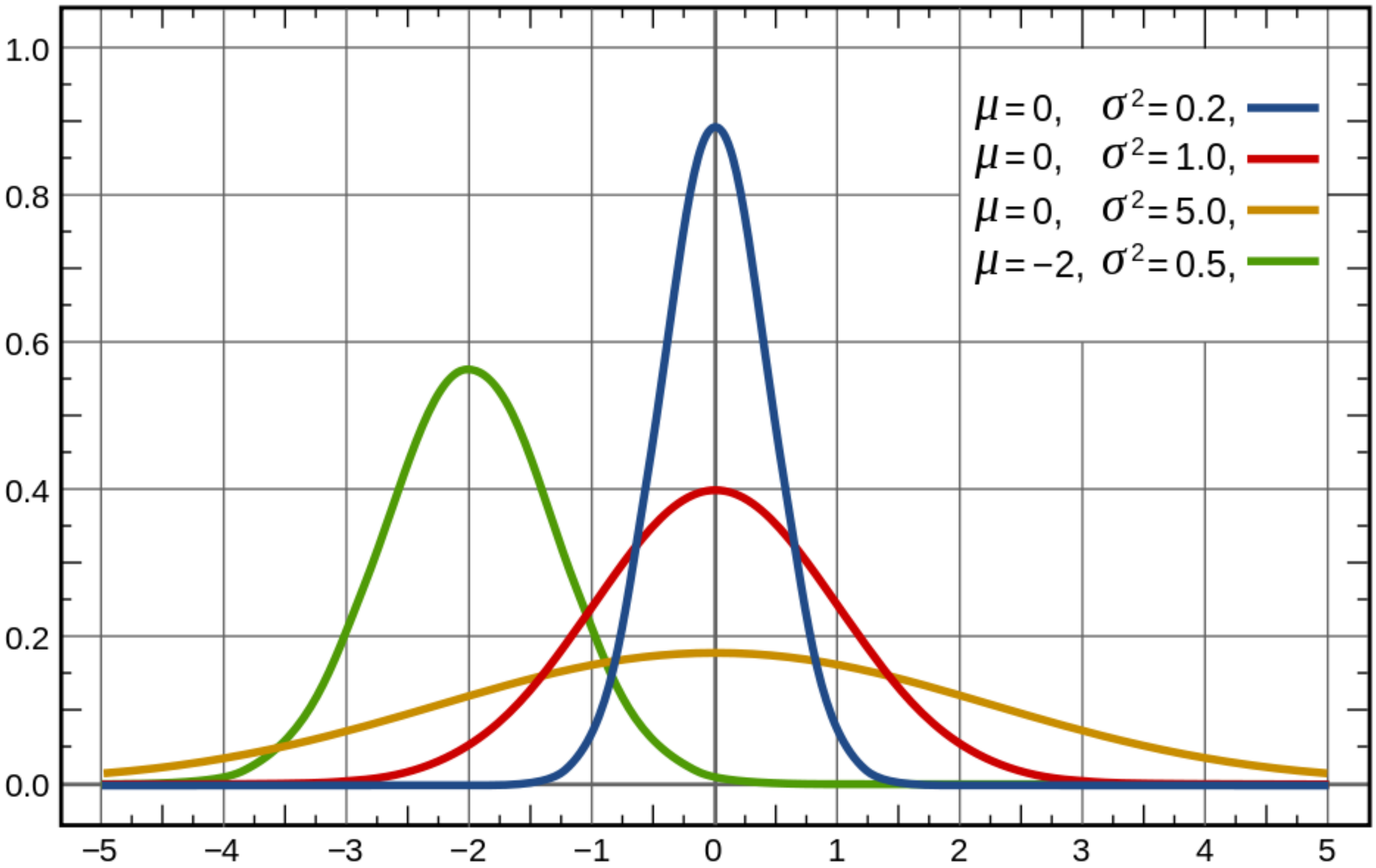
http://alex.smola.org/teaching/10-701-15

Carnegie Mellon University

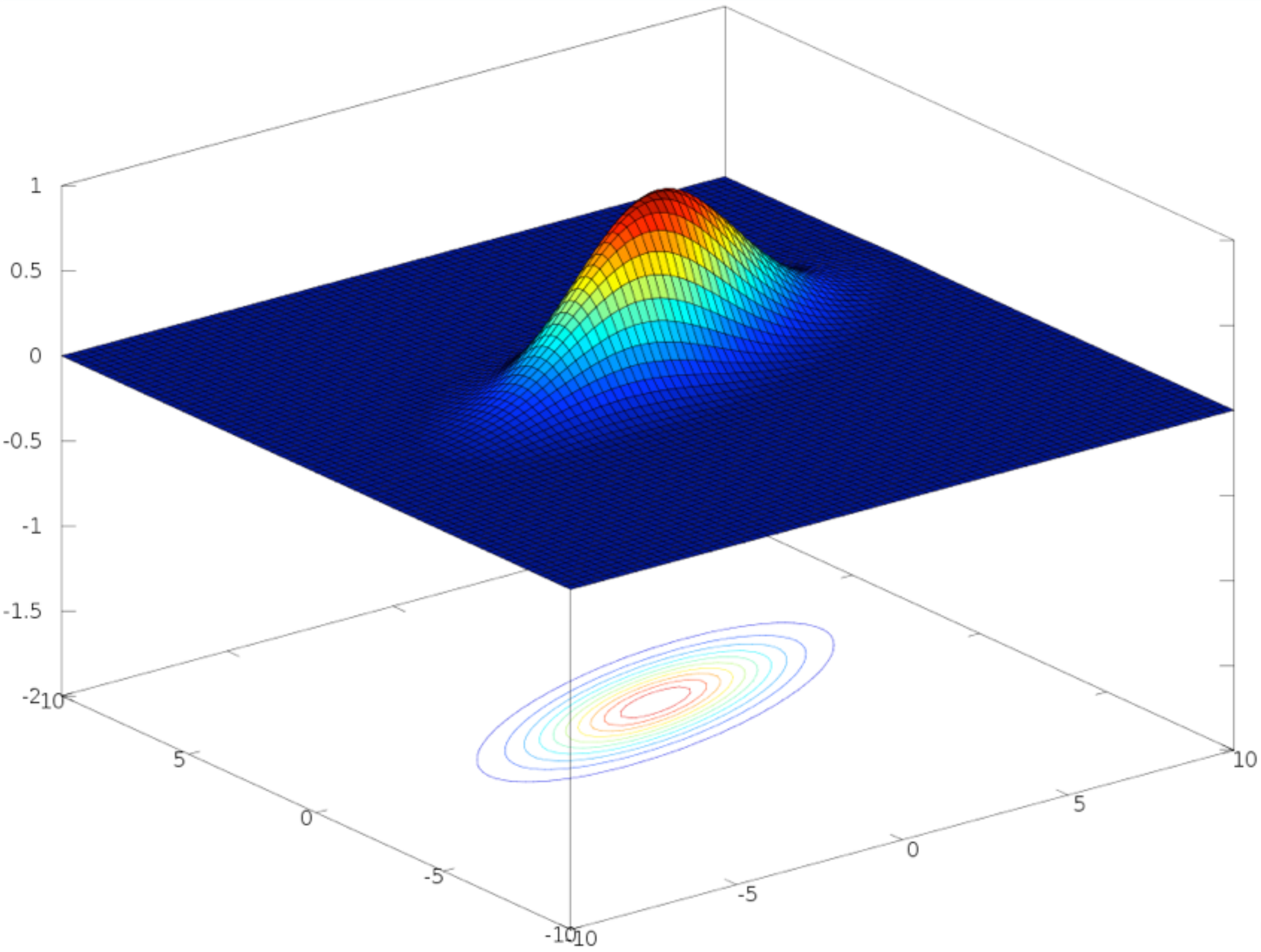
Normal Distribution

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

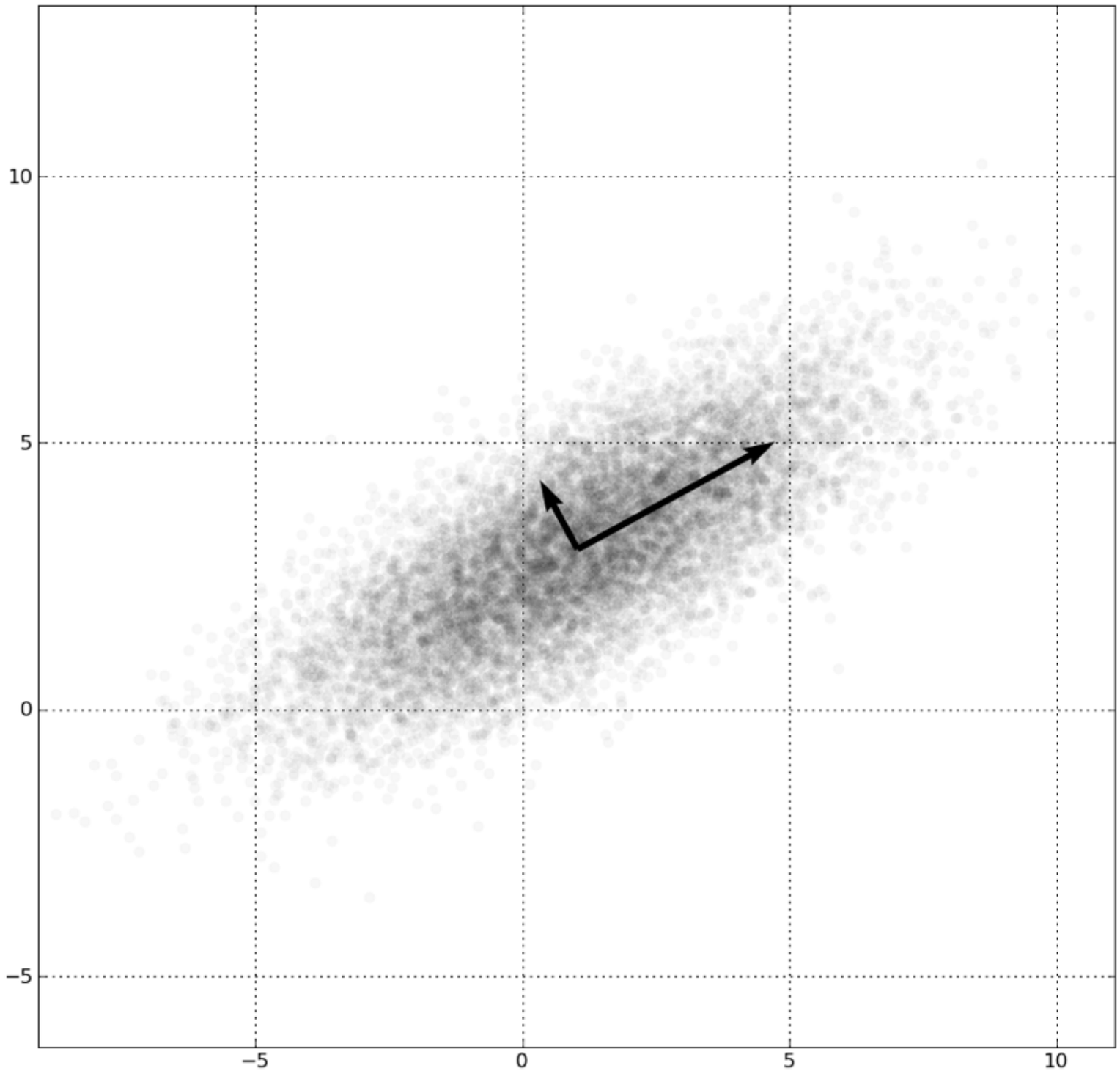
The Normal Distribution



Gaussians in Space



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

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

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

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

Recall - Gaussian is in the Exponential Family

- Binomial Distribution
- Discrete Distribution
  (e_x is unit vector for x)
- Gaussian
- Poisson (counting measure 1/x!)
- Dirichlet, Beta, Gamma, Wishart, ...

$$\phi(x) = x$$

$$\phi(x) = e_x$$

$$\phi(x) = \begin{pmatrix} x, \frac{1}{2} x x^\top \end{pmatrix}$$

$$\phi(x) = x$$

Recall - Gaussian is in the Exponential Family

- Binomial Distribution
- Discrete Distribution
  (e_x is unit vector for x)
- Gaussian
- Poisson (counting measure 1/x!)
- Dirichlet, Beta, Gamma, Wishart, ...

$$\phi(x) = x$$

$$\phi(x) = e_x$$

$$\phi(x) = \left(x, \frac{1}{2} x x^\top\right)$$

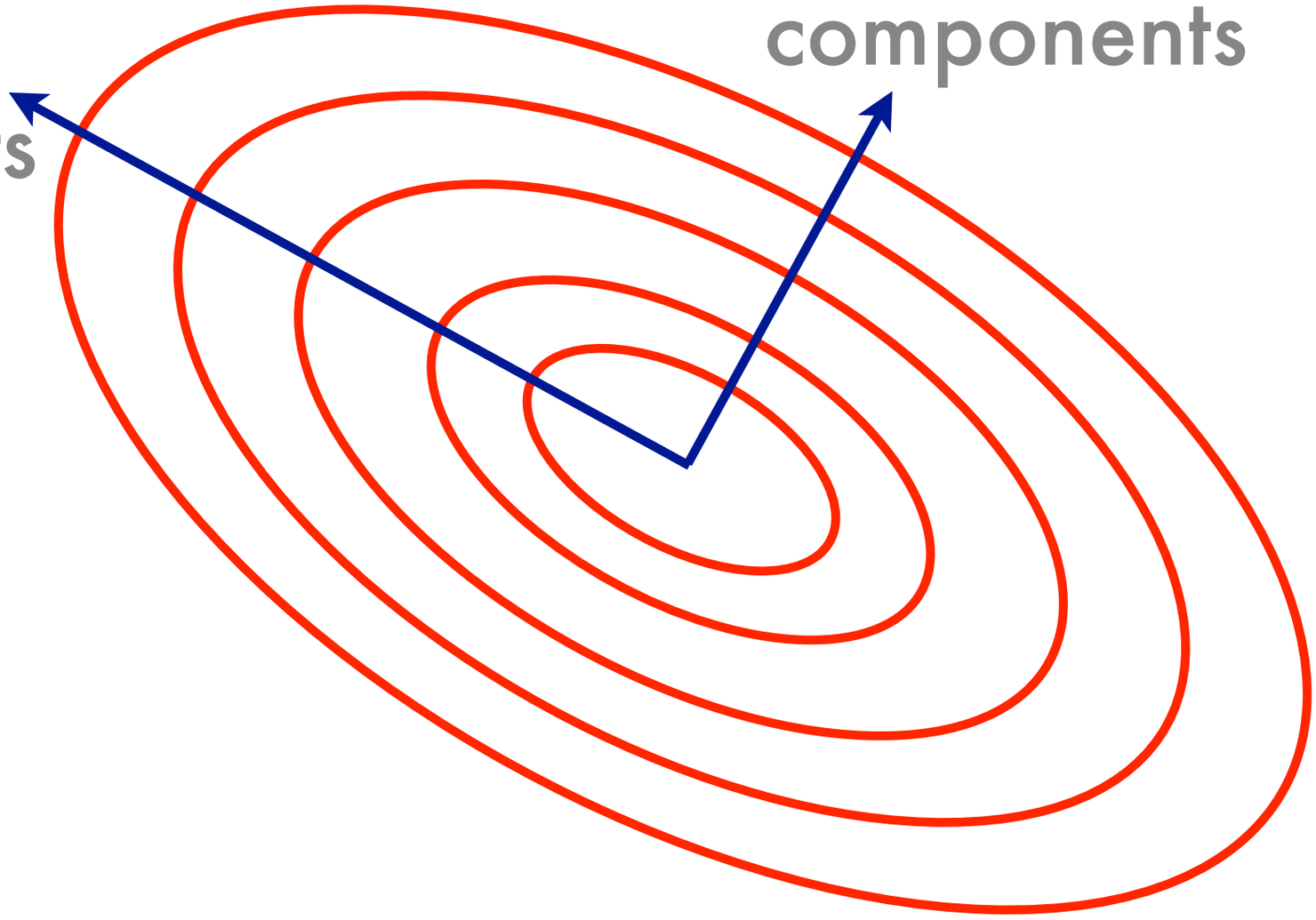
$$\phi(x) = x$$

$$-\partial_\theta \log p(X;\theta) = m \left[\mathbf{E}[\phi(x)] - \frac{1}{m}\sum_{i=1}^{n} \phi(x_i)\right]$$

The Normal Distribution

principal components

principal components



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

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

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: $\mathbf{E}\left[(x-\mu)(x-\mu)^\top\right] = \mathbf{E}\left[Lzz^\top L^\top\right]$

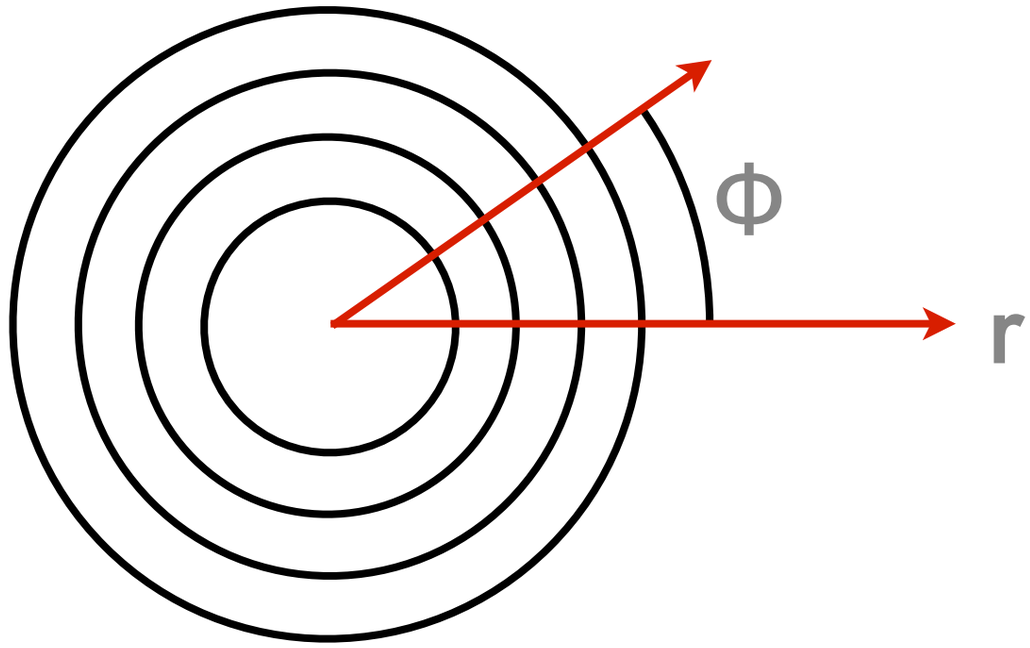
$$= L\mathbf{E}\left[zz^\top\right]L^\top = LL^\top = \Sigma$$

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

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

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

Sampling from a Gaussian



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

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

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

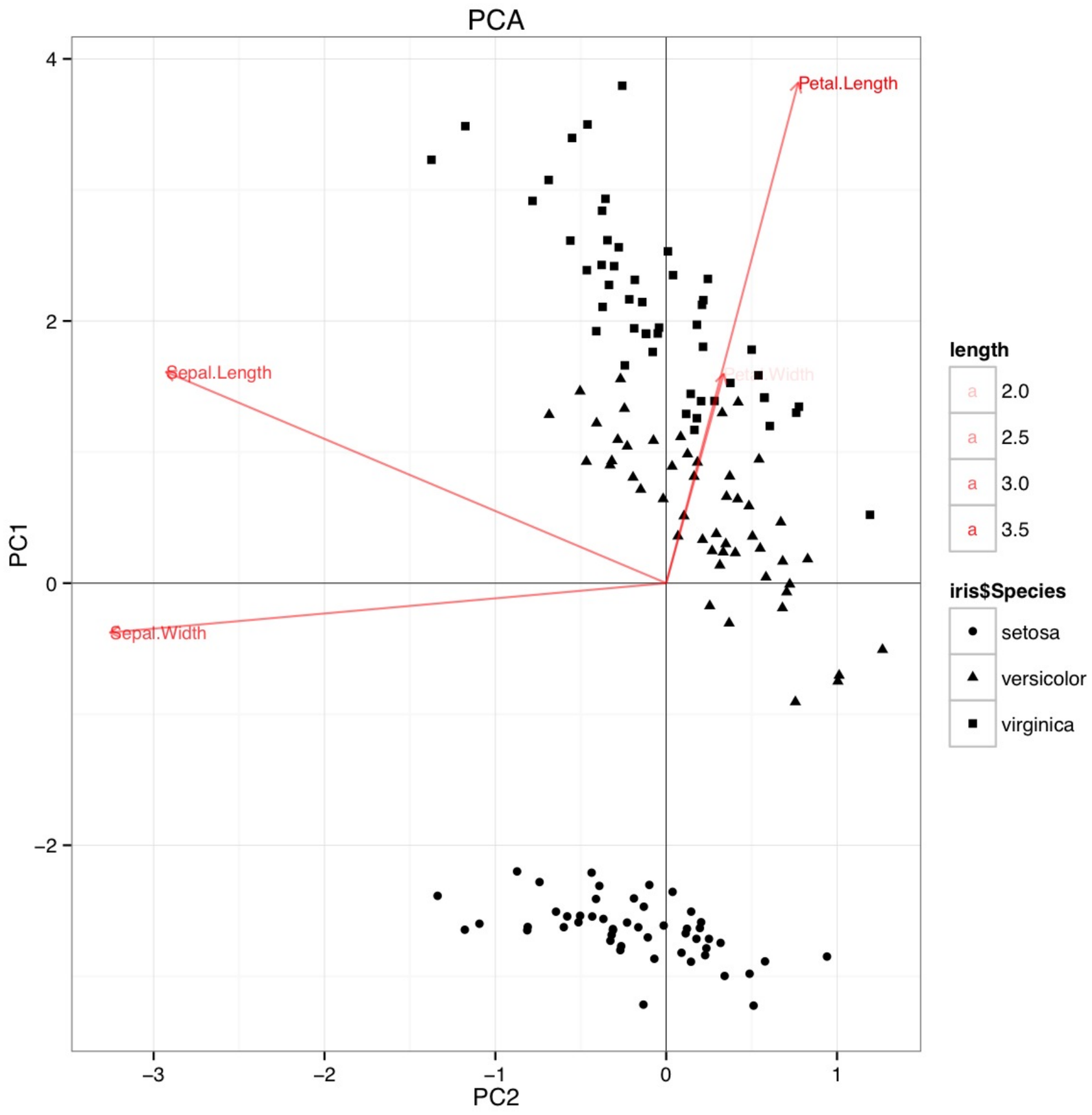
- Cumulative distribution function

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

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?



Principal Component Analysis

Data Visualization

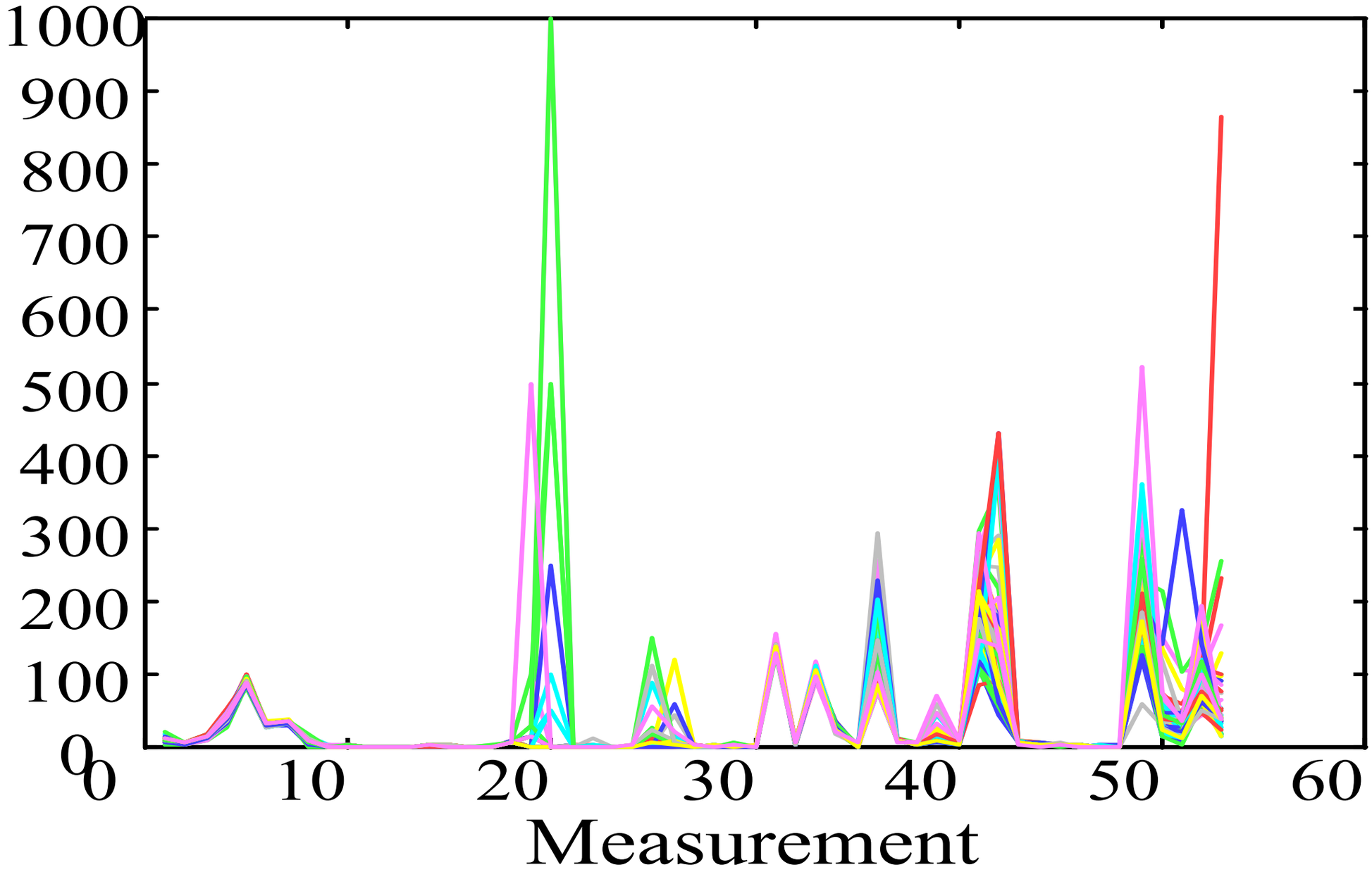
Instances

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

Features

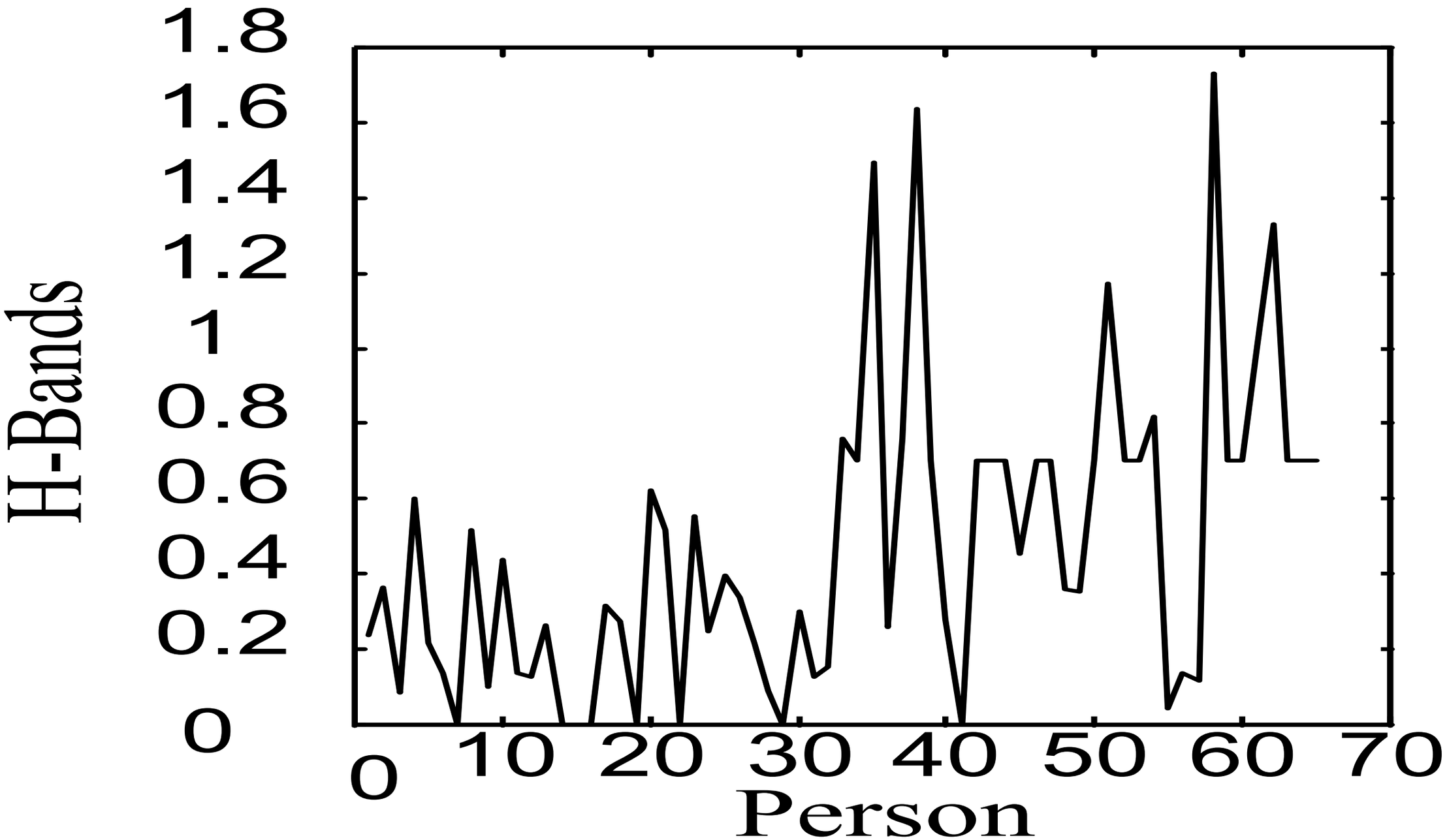
- 53 Blood and urine samples from 65 people
- Difficult to see the correlations between features

Data Visualization



- Spectral format (65 curves, one for each person)
- Difficult to compare different patients

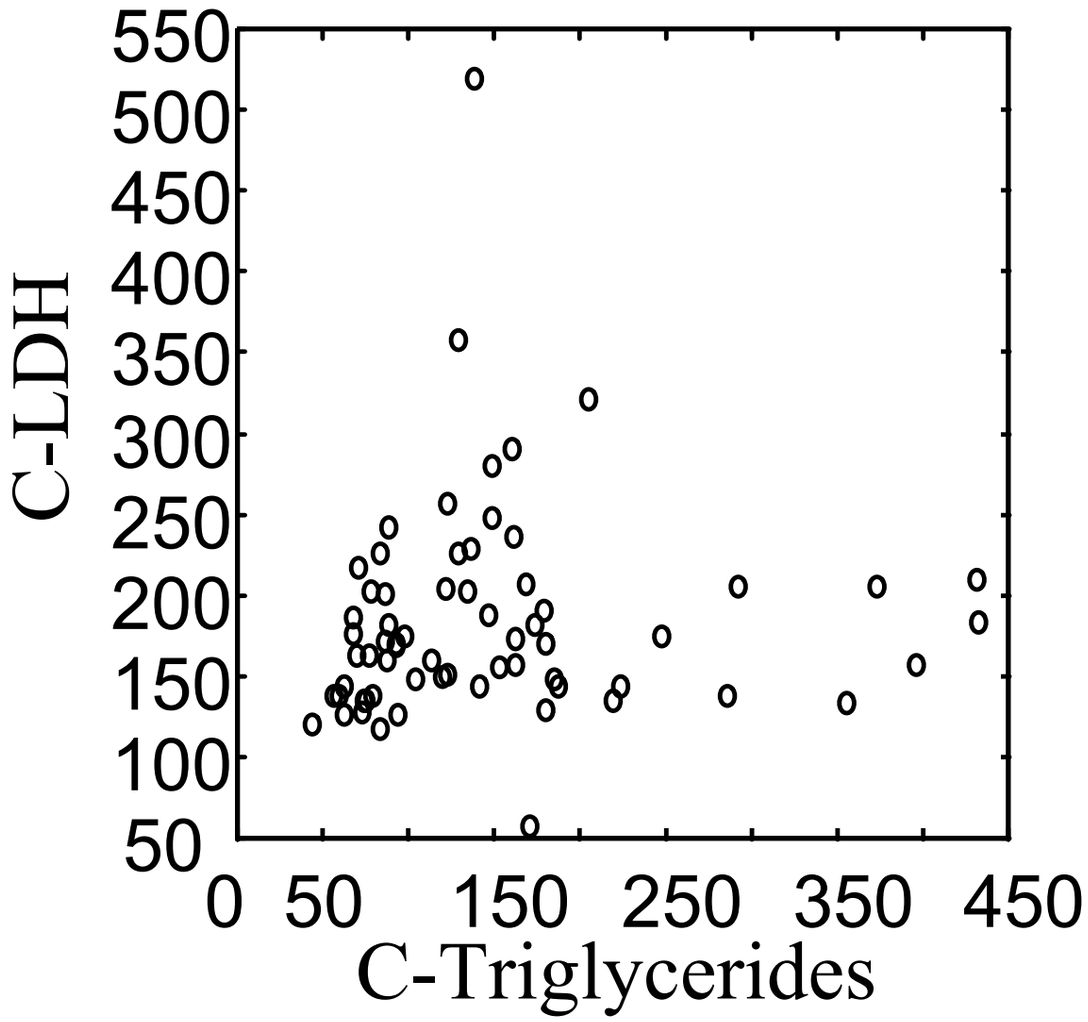
Data Visualization



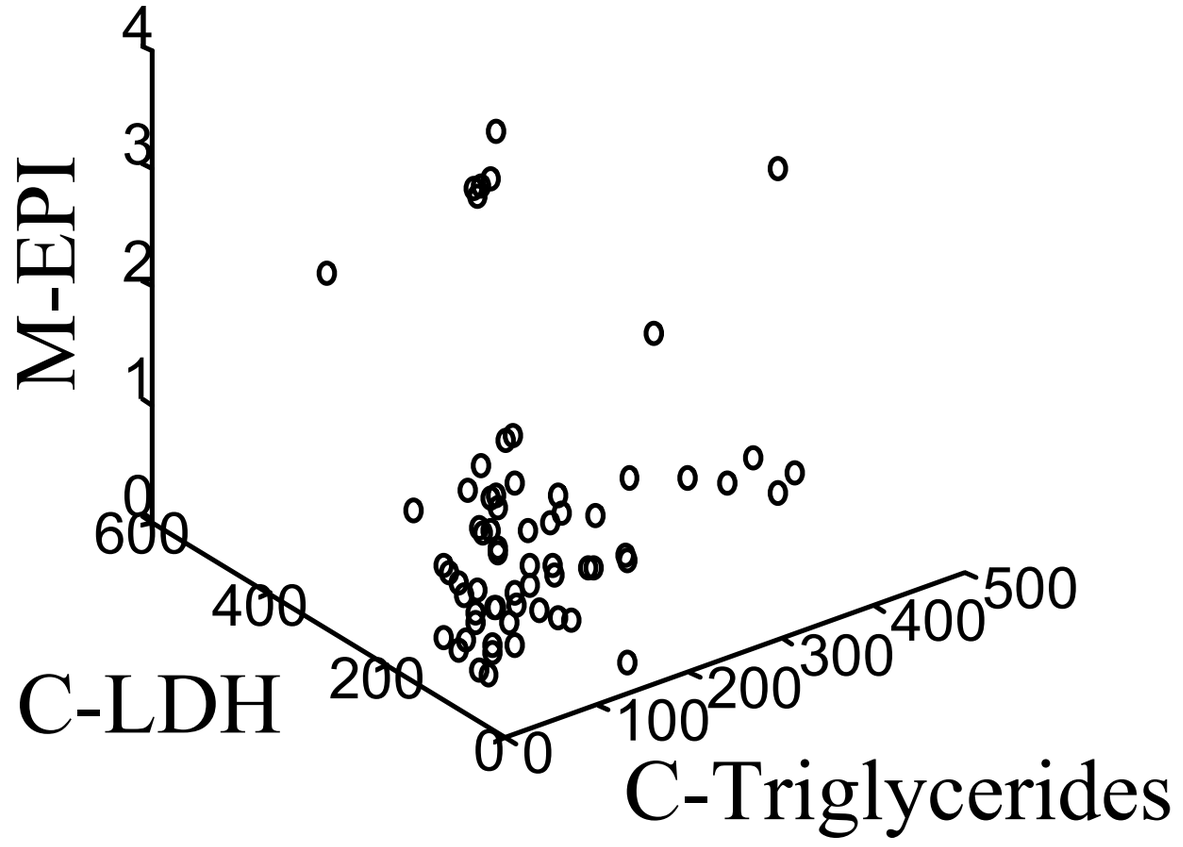
One plot per person …

Data Visualization

Bi-variate



Tri-variate



Even 3 dimensions are already difficult. How to extend this?

Compact Summaries via PCA

$$\underset{\text{rank}P=k}{\text{minimize}} \frac{1}{m}\sum_{i=1}^{m}\|x_i - Px_i\|^2 \quad \text{where} \quad \frac{1}{m}\sum_{i=1}^{m} x_i = \mu$$

assume centering

$$\text{tr}\left[\frac{1}{m}\sum_{i=1}^{m} x_i x_i^\top - Px_i x_i^\top P^\top\right]$$

$$\text{tr}\,\Sigma - \text{tr}\,P\Sigma P^\top$$

maximize this

Compact Summaries via PCA

- Is there a representation better than the coordinate axes?
- Is it really necessary to show all the 53 dimensions?
  - What if there are strong correlations between features?
  - What if there's some additive noise?

$$\underset{\operatorname{rank} P=k}{\text{minimize}} \frac{1}{m}\sum_{i=1}^{m}\|x_i - Px_i\|^2 \text{ where } \frac{1}{m}\sum_{i=1}^{m} x_i = \mu$$

$$\operatorname{tr}\left[\frac{1}{m}\sum_{i=1}^{m} x_i x_i^\top - P x_i x_i^\top P^\top\right]$$

$$\operatorname{tr}\Sigma - \operatorname{tr} P\Sigma P^\top$$

assume centering

maximize this

Compact Summaries via PCA

- Is there a representation better than the coordinate axes?
- Is it really necessary to show all the 53 dimensions?
  - What if there are strong correlations between features?
  - What if there's some additive noise?
- Find the **smallest subspace** that keeps **most information**

$$\underset{\text{rank}P=k}{\text{minimize}} \frac{1}{m}\sum_{i=1}^{m}\|x_i - Px_i\|^2 \text{ where } \frac{1}{m}\sum_{i=1}^{m} x_i = \mu$$

$$\text{tr}\left[\frac{1}{m}\sum_{i=1}^{m} x_i x_i^\top - Px_i x_i^\top P^\top\right]$$

$$\text{tr}\,\Sigma - \text{tr}\,P\Sigma P^\top$$

assume centering

maximize this

Compact Summaries via PCA

maximize this

$$\text{Residual} = \operatorname{tr}\Sigma - \operatorname{tr} P\Sigma P^\top$$

$$= \sum_{i=1}^{d} \sigma_i^2 - \sum_{i=1}^{k} \sigma_i^2 = \sum_{i=k+1}^{d} \sigma_i^2$$

$$x = z + \epsilon \text{ where } z \sim \mathcal{N}(\mu, \Sigma) \text{ and } \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$$

$$\Sigma + \sigma^2 \mathbf{1}$$

$$\sigma_i^2 + \sigma^2$$

Compact Summaries via PCA

- Subspace optimization

maximize this

$$\text{Residual} = \operatorname{tr}\Sigma - \operatorname{tr} P\Sigma P^\top$$

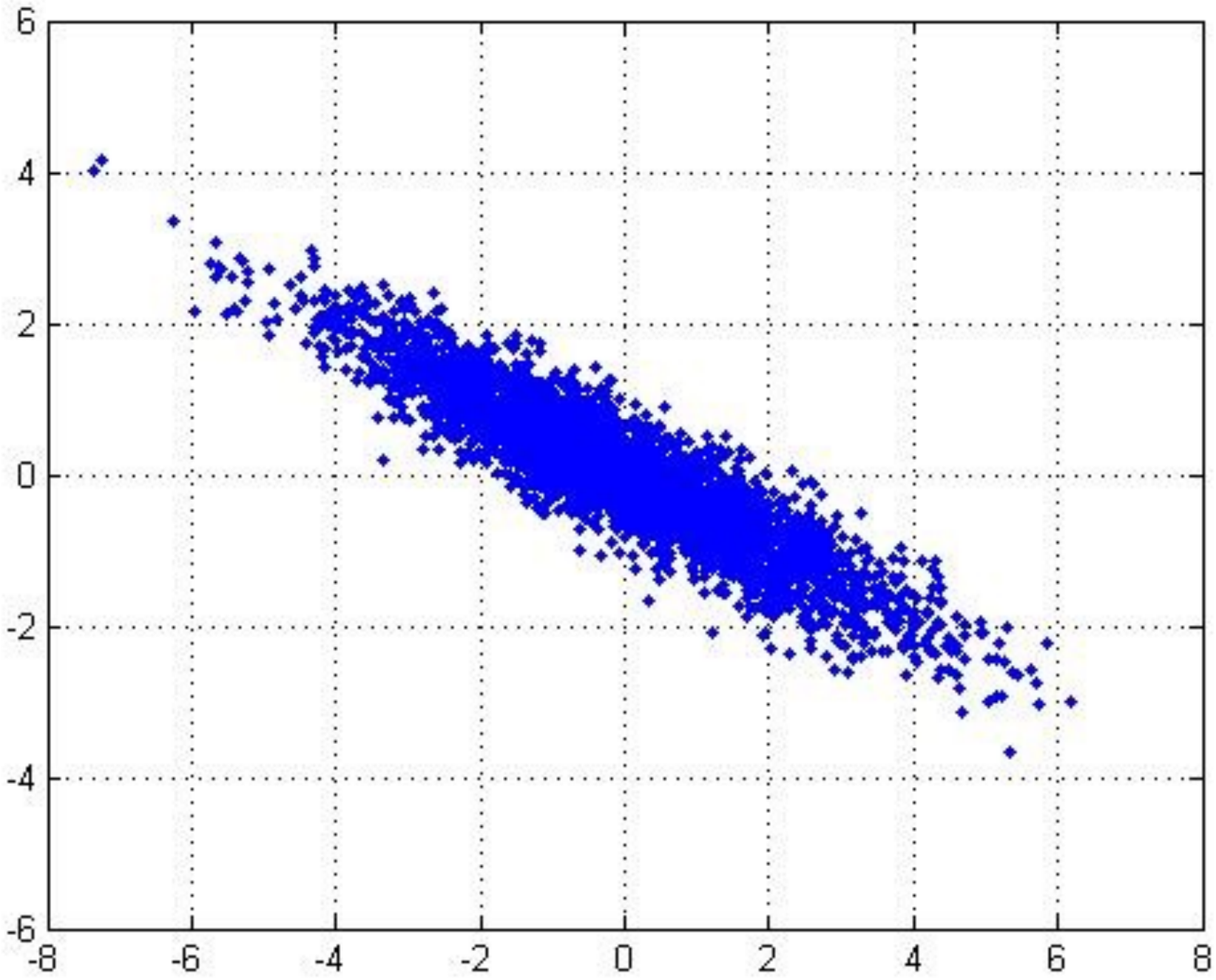
$$= \sum_{i=1}^{d} \sigma_i^2 - \sum_{i=1}^{k} \sigma_i^2 = \sum_{i=k+1}^{d} \sigma_i^2$$

- Signal to Noise ratio optimization
  - Assume data x is generated with additive noise

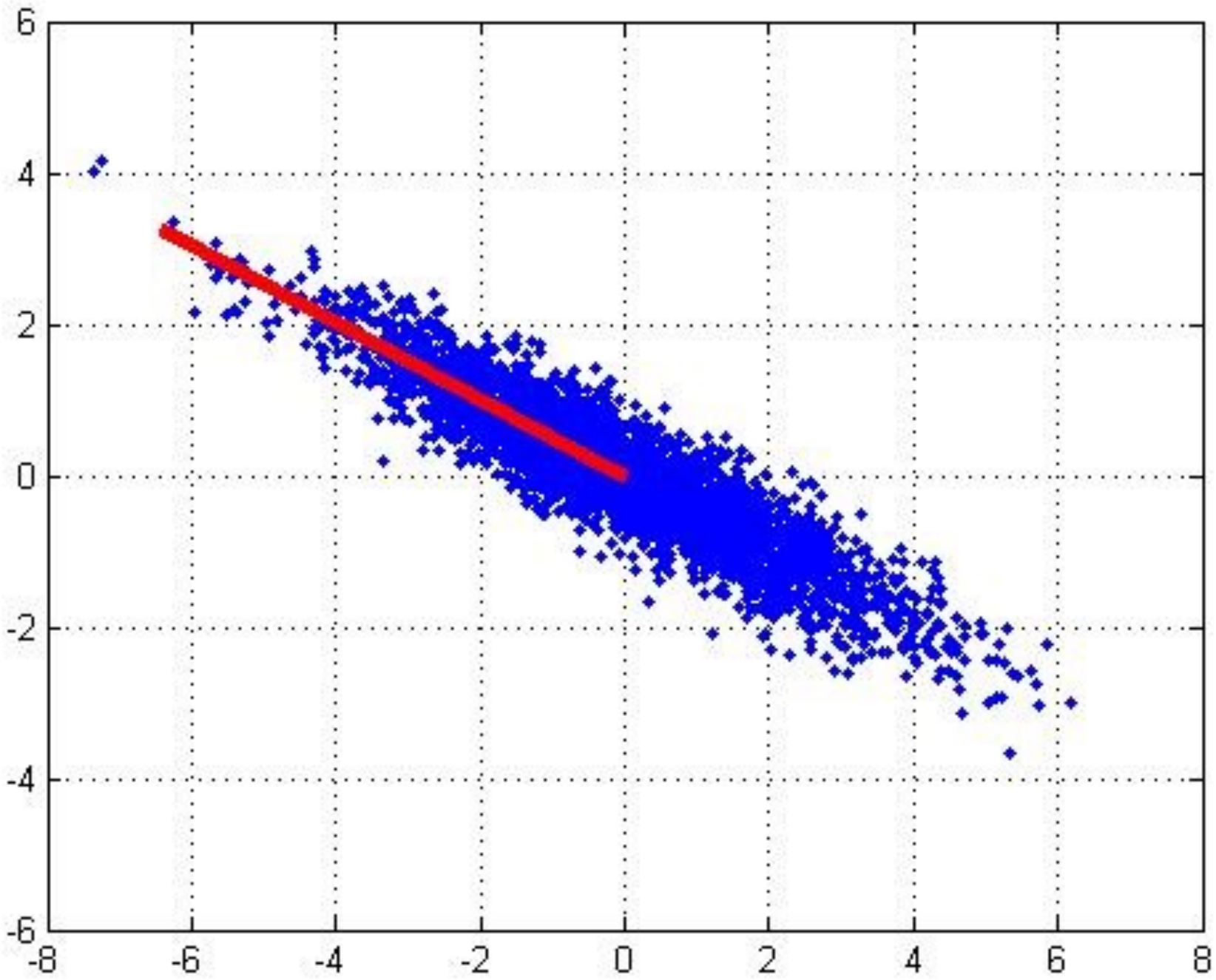
    $$x = z + \epsilon \text{ where } z \sim \mathcal{N}(\mu, \Sigma) \text{ and } \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$$

  - Joint covariance matrix is $\Sigma + \sigma^2 \mathbf{1}$
  - Joint eigenvalues are $\sigma_i^2 + \sigma^2$, so we can ignore everything below the noise threshold

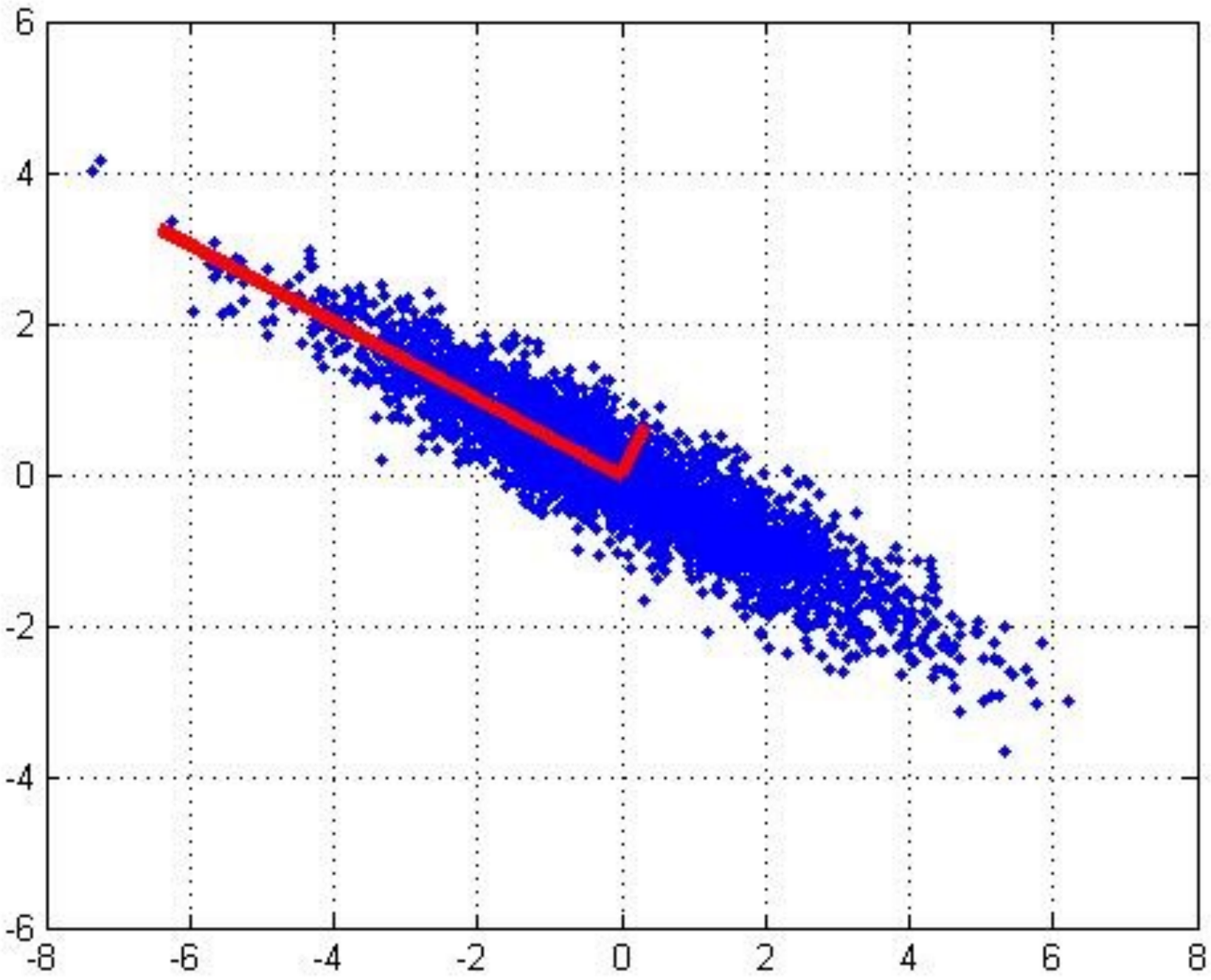
2d dataset



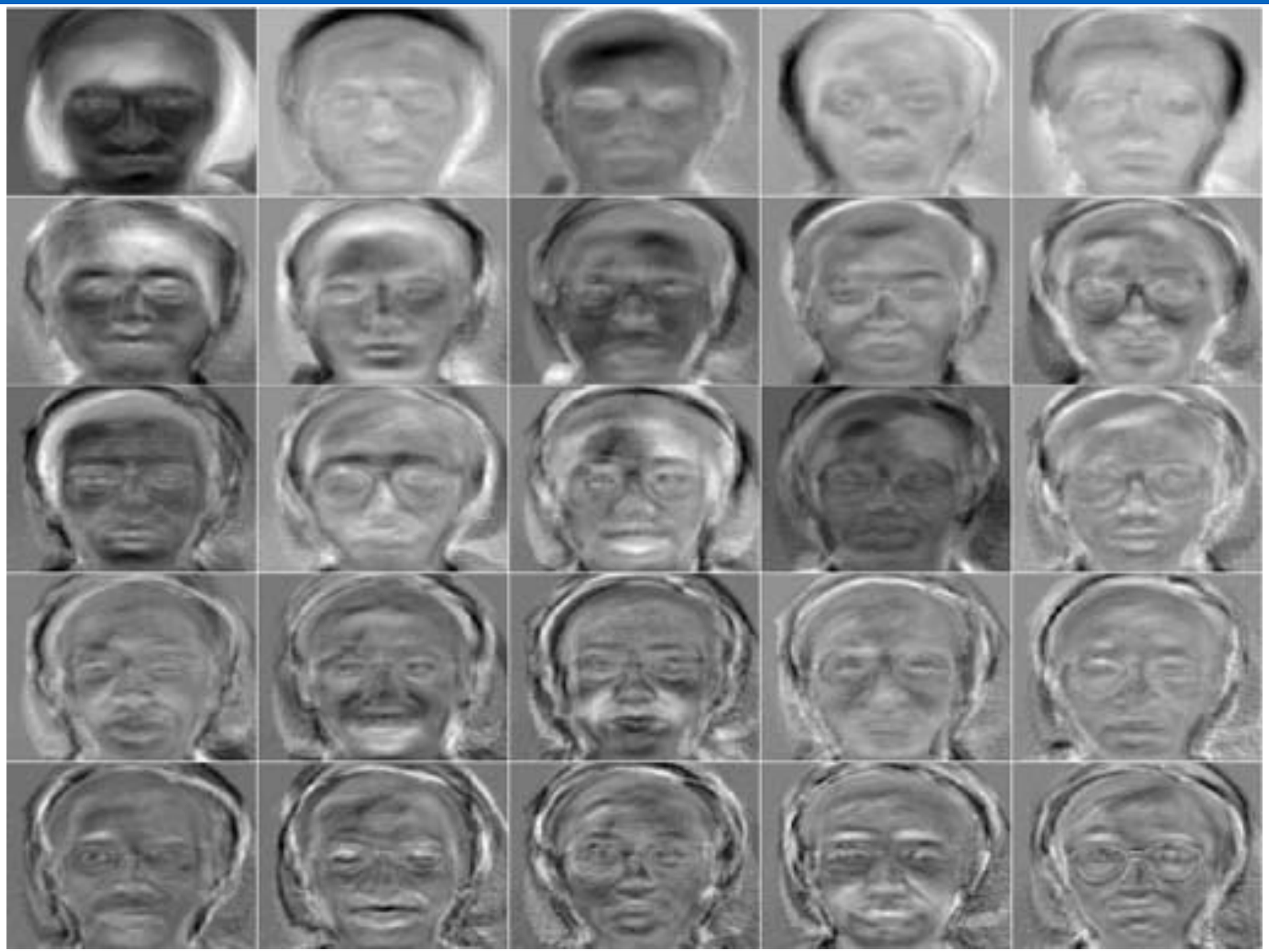
First principal axis



Second principal axis



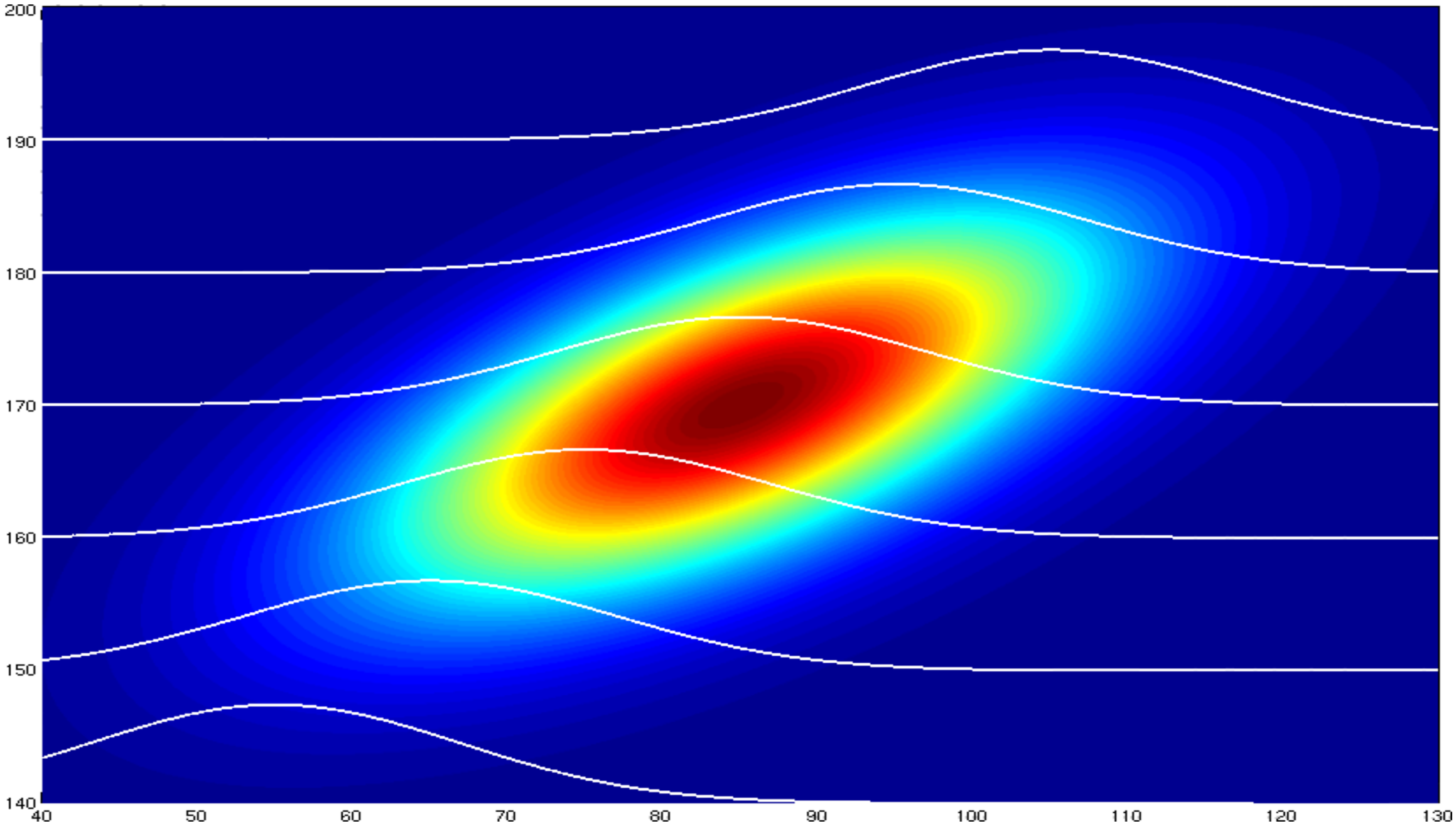
Eigenfaces (PCA on images)



Eigenfaces (PCA on images)

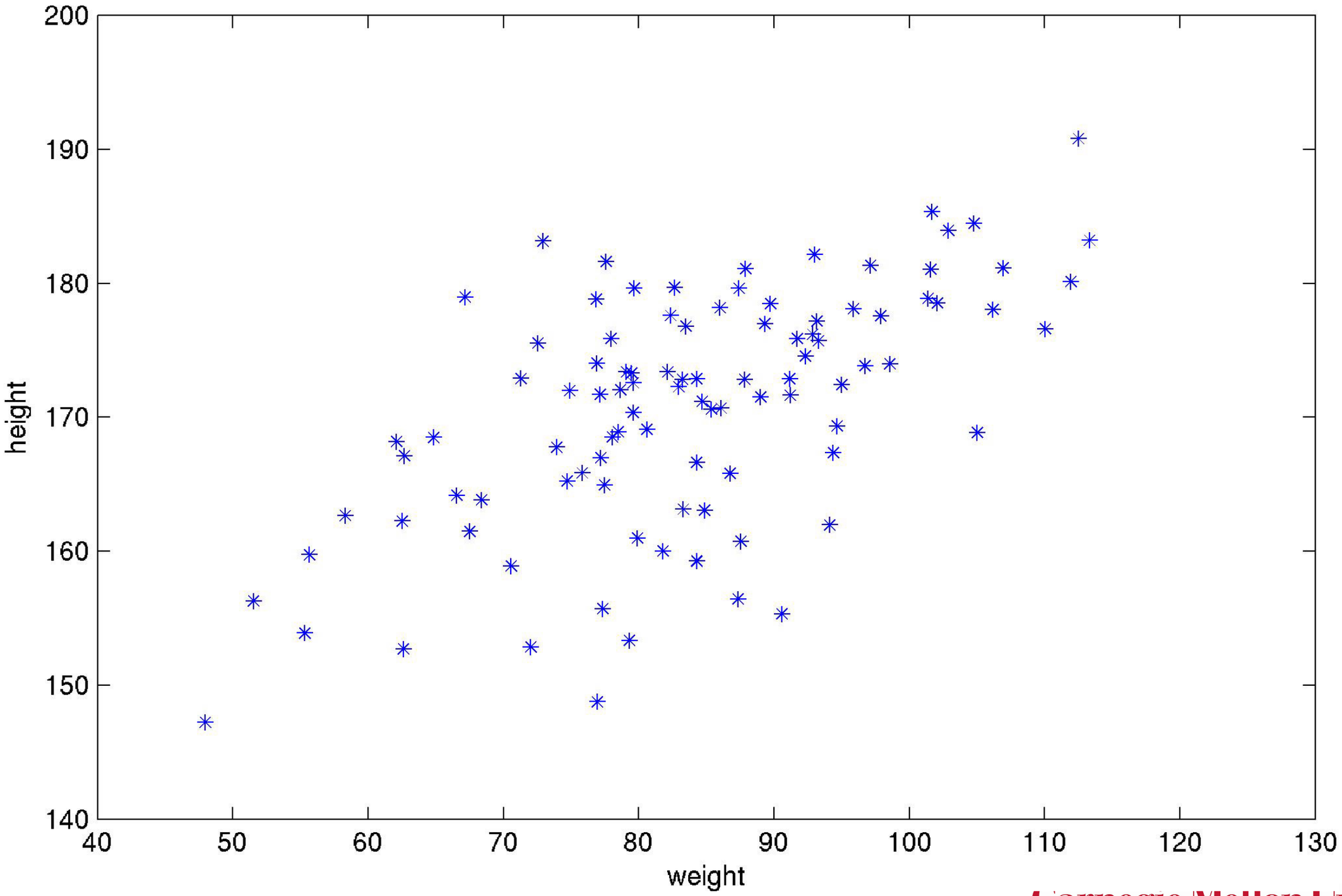
When projecting strange data

- Original images
- Reconstruction doesn't look like the original

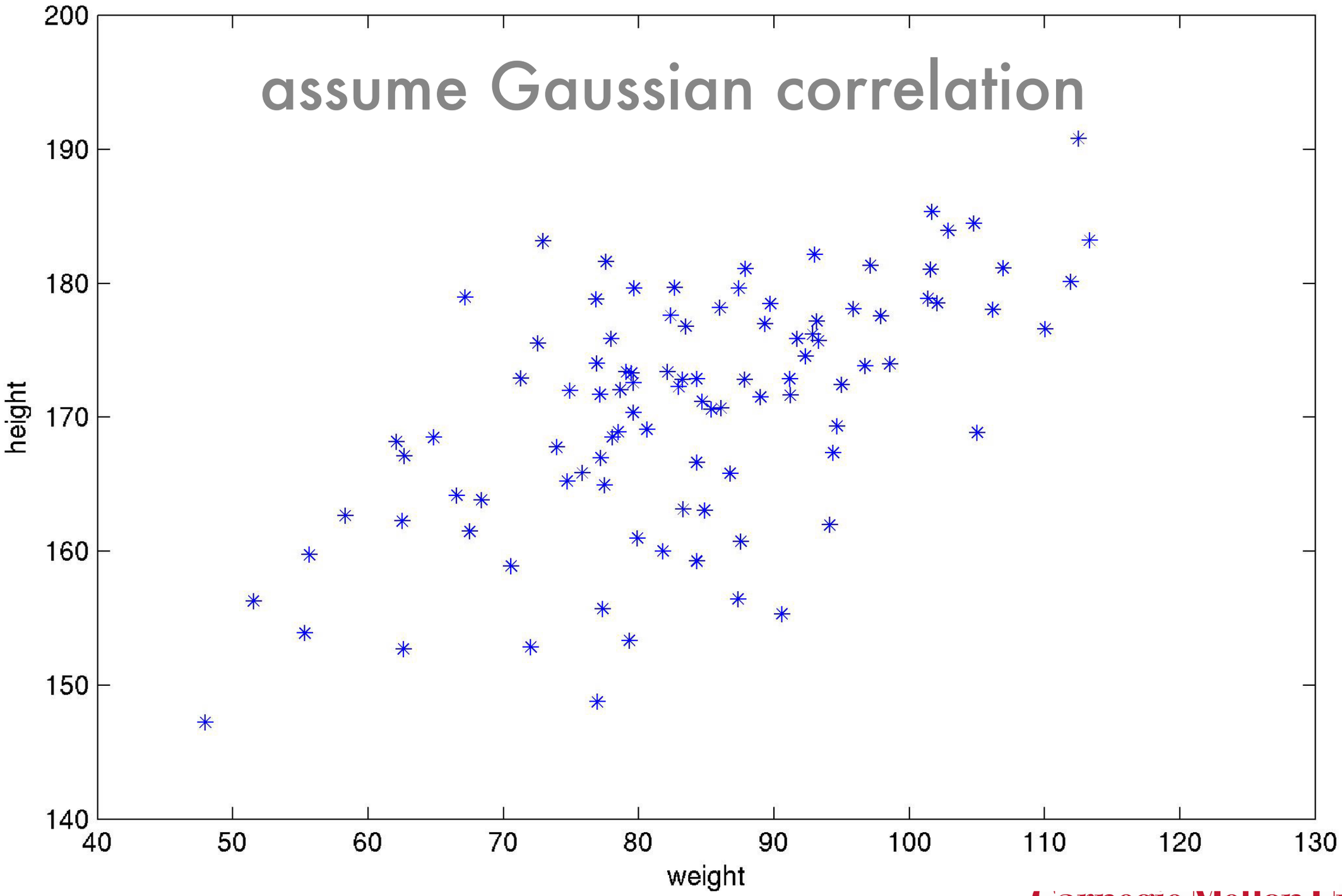


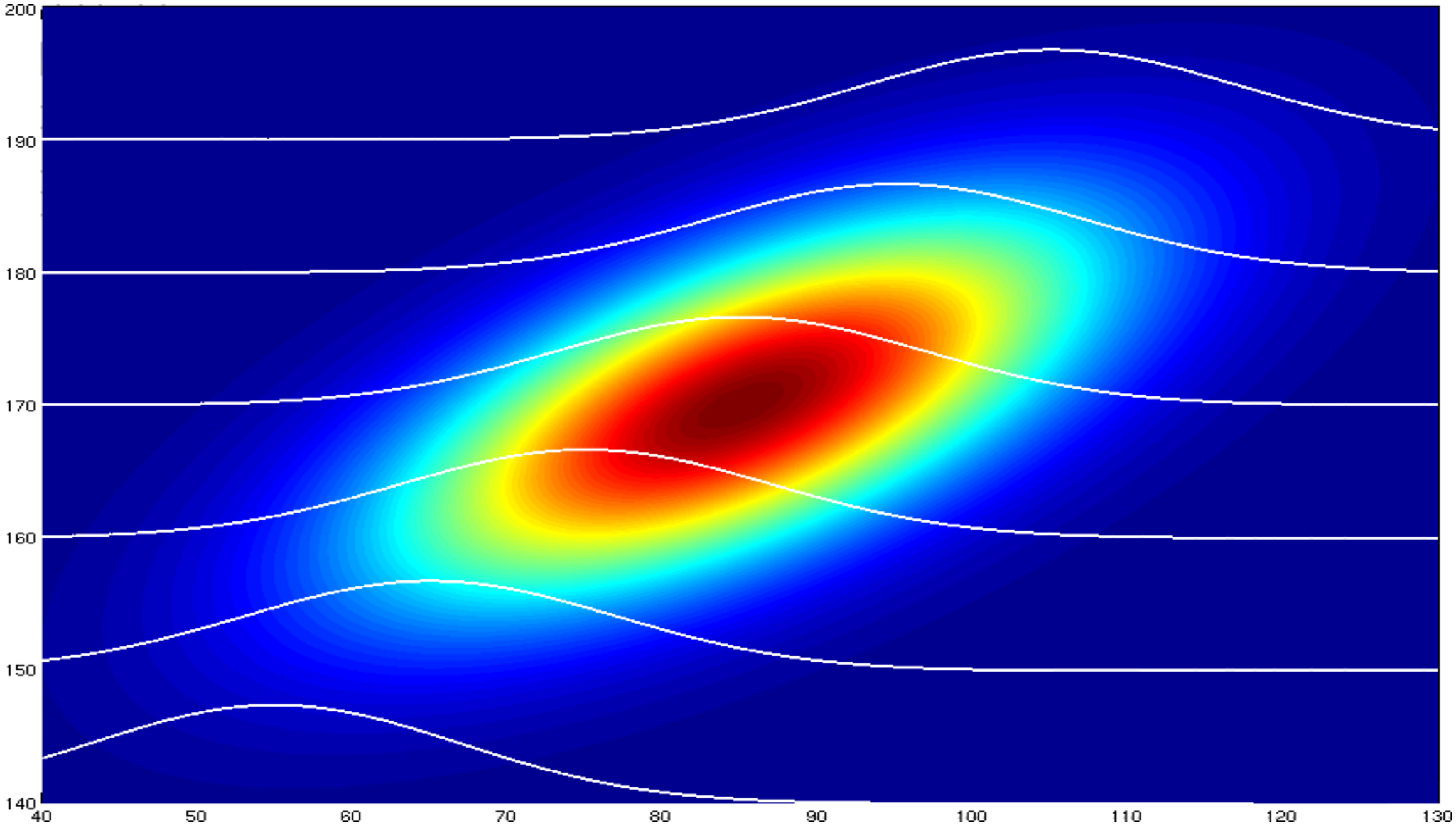
Inference

Correlating weight and height

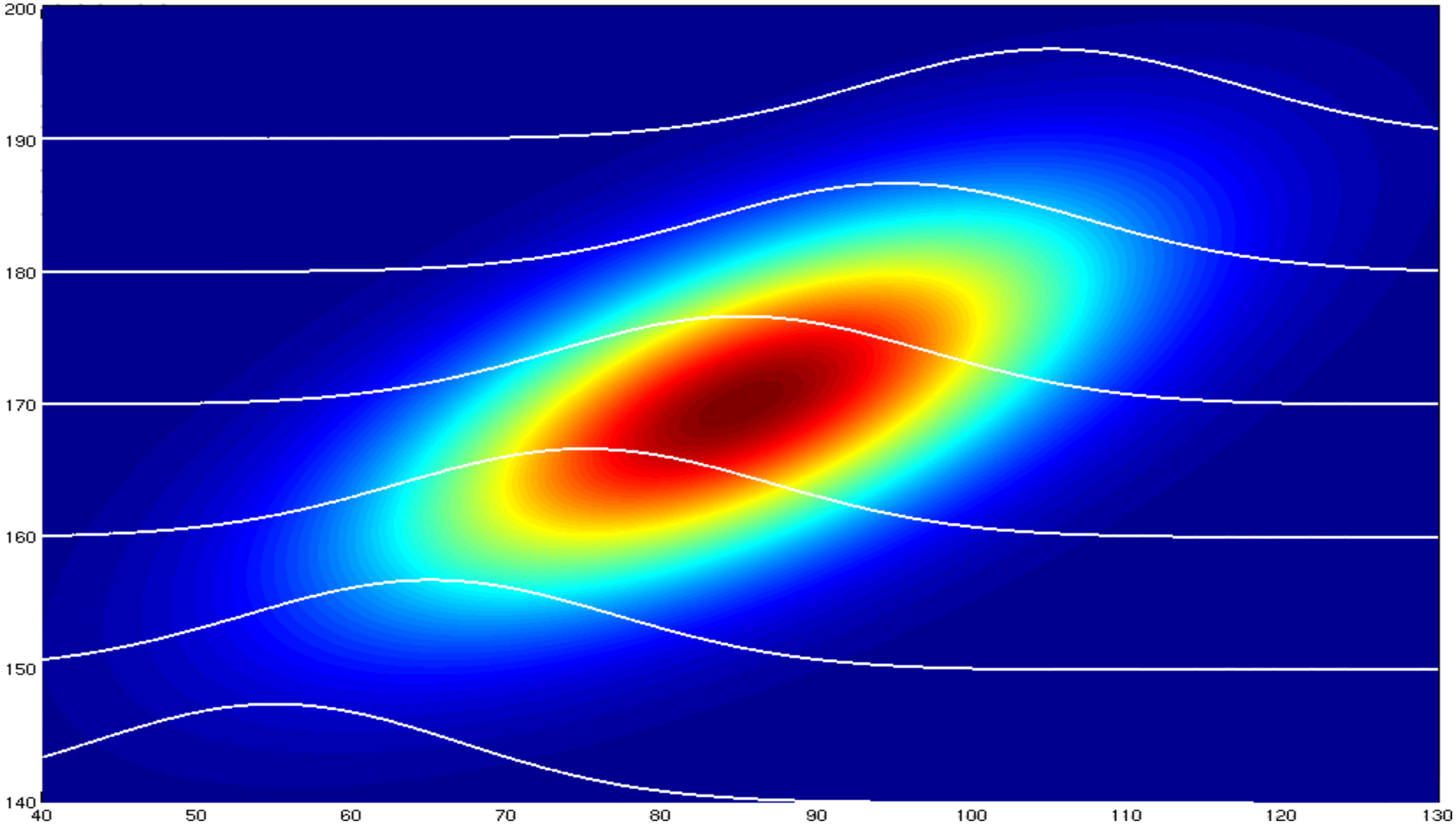


Correlating weight and height





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



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

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!

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

MACHINE LEARNING
DEPARTMENT

6.2 Gaussian Processes

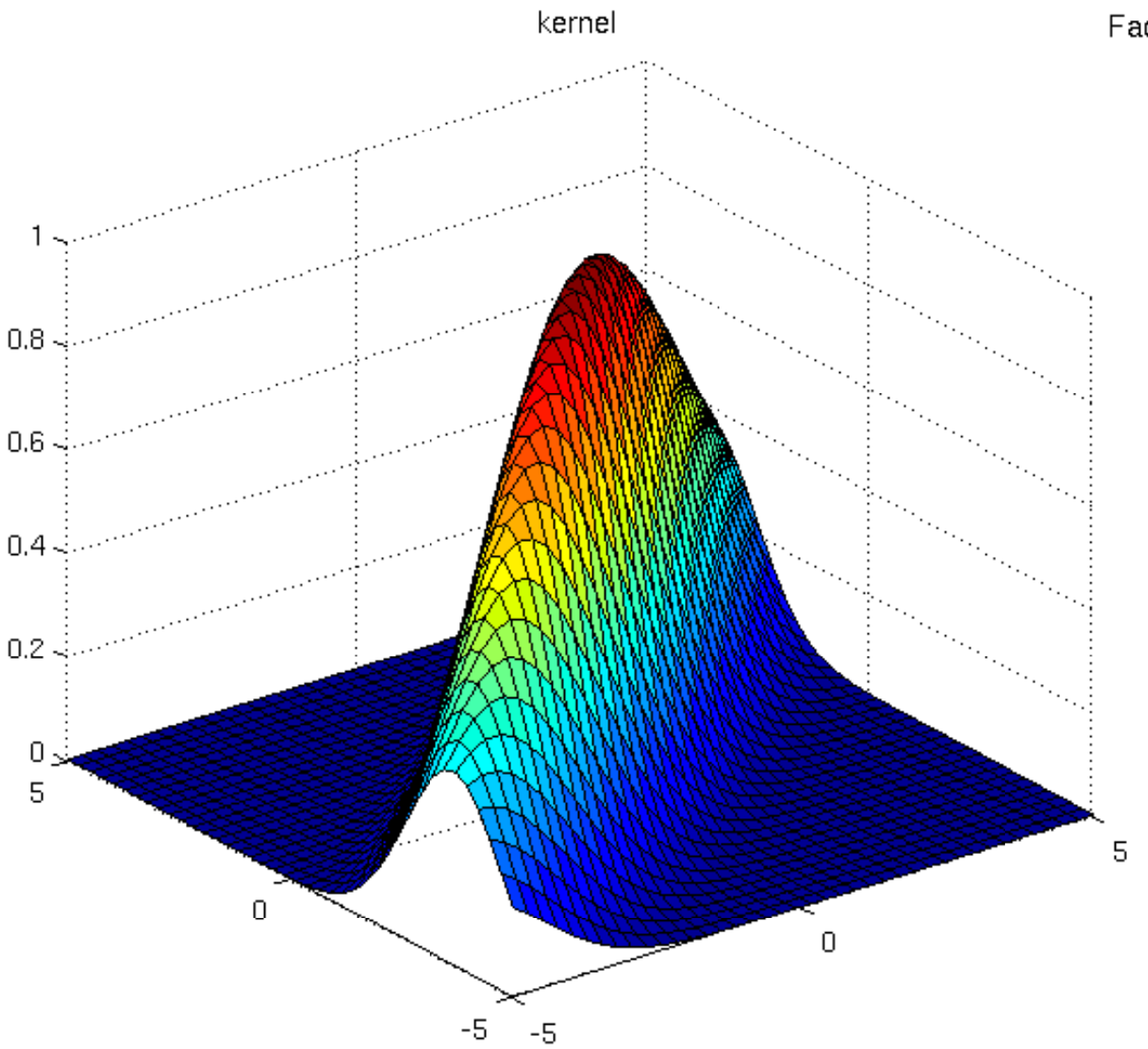
6 Bayesian Kernel Methods

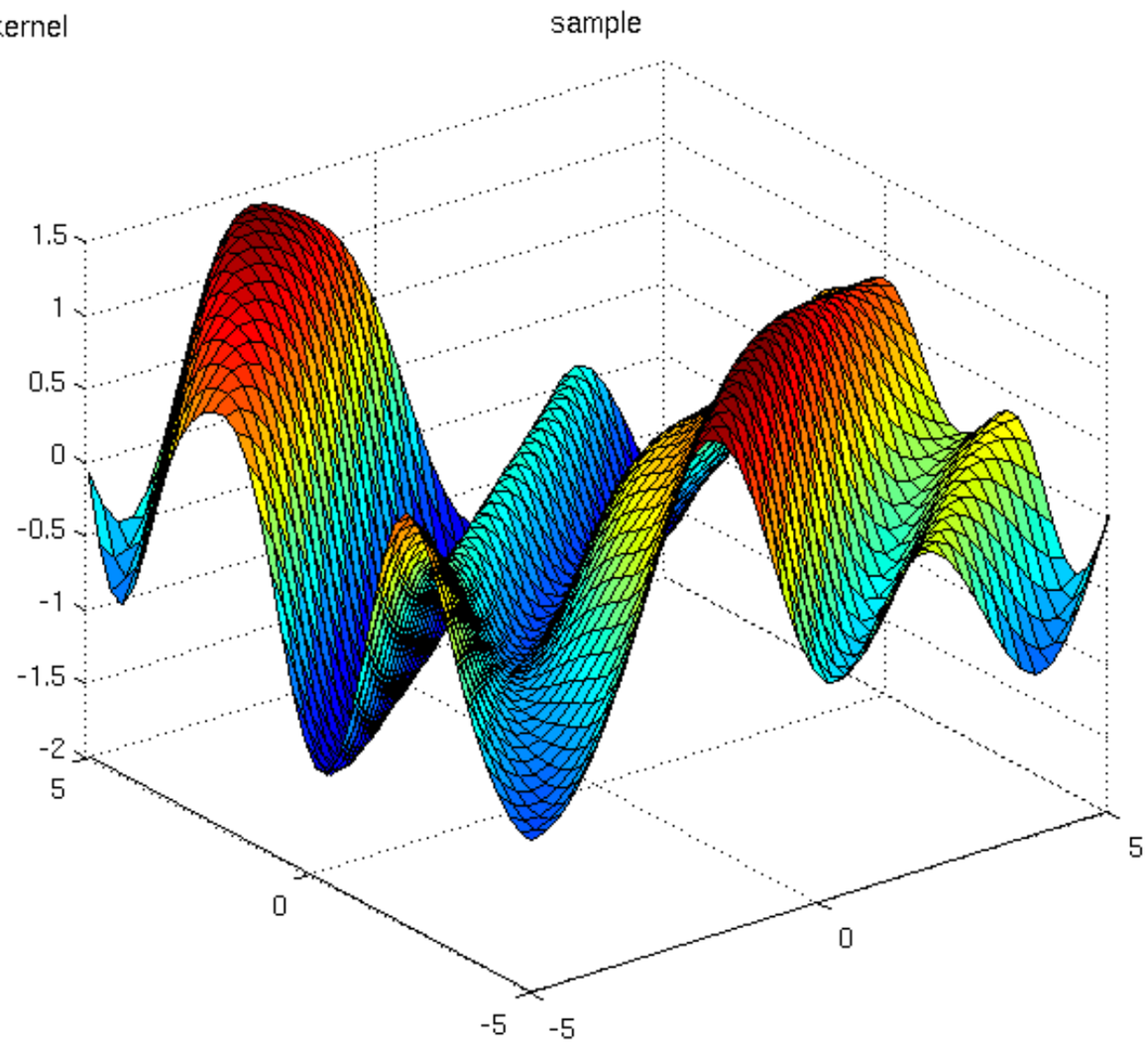
Alexander Smola

Introduction to Machine Learning 10-701

http://alex.smola.org/teaching/10-701-15

Carnegie Mellon University





Gaussian Process

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

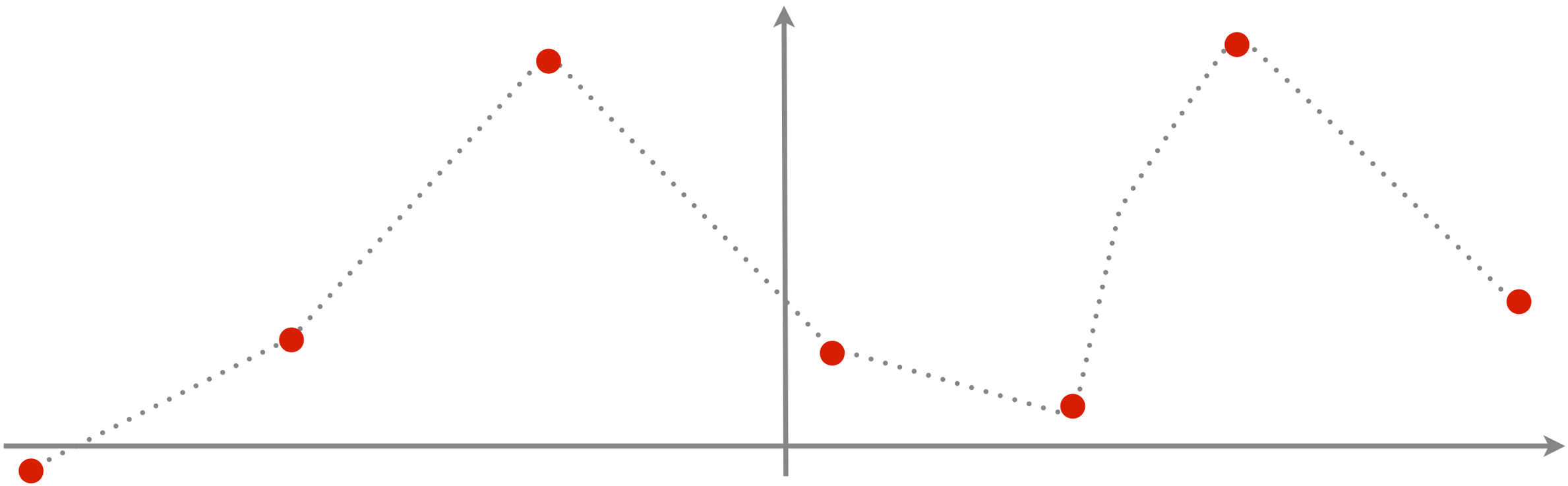
Covariance Function $k(x, x') := \text{Cov}(t(x), t(x'))$

Simplifying Assumption

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

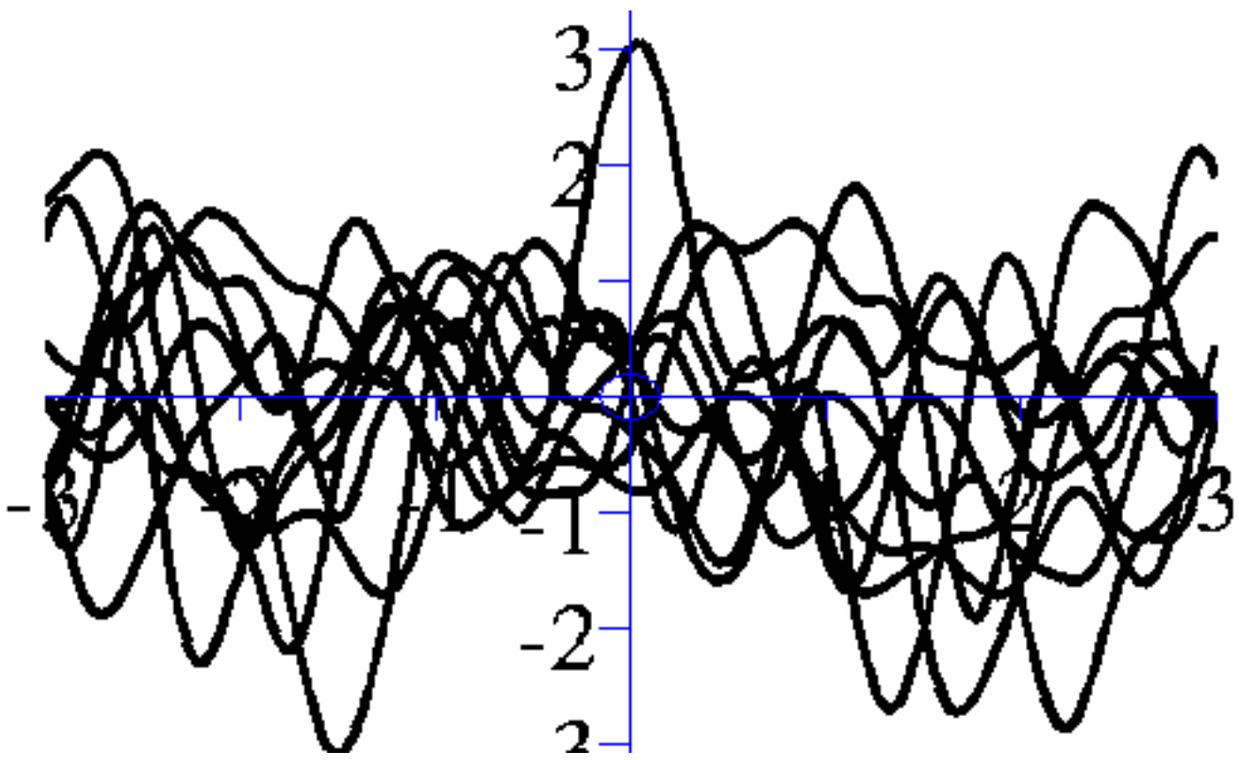
Gaussian Process

- 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 . . .

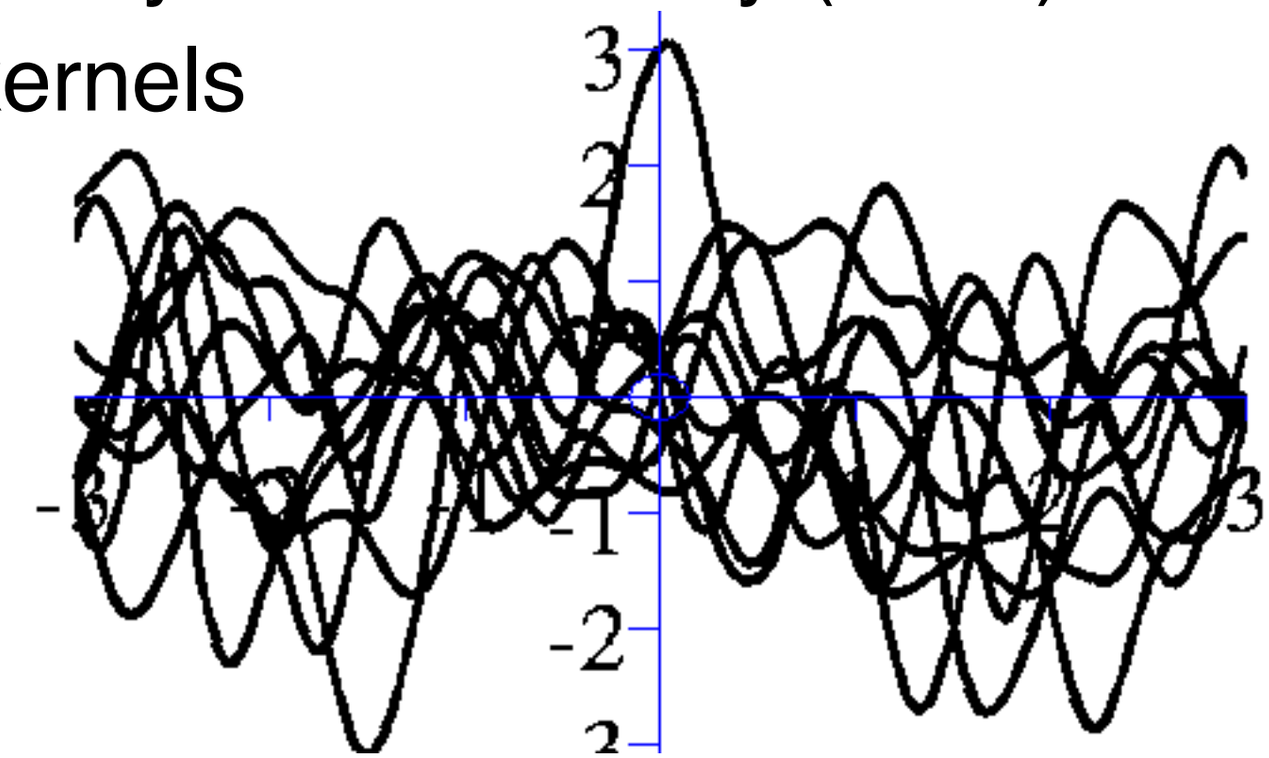
yes!

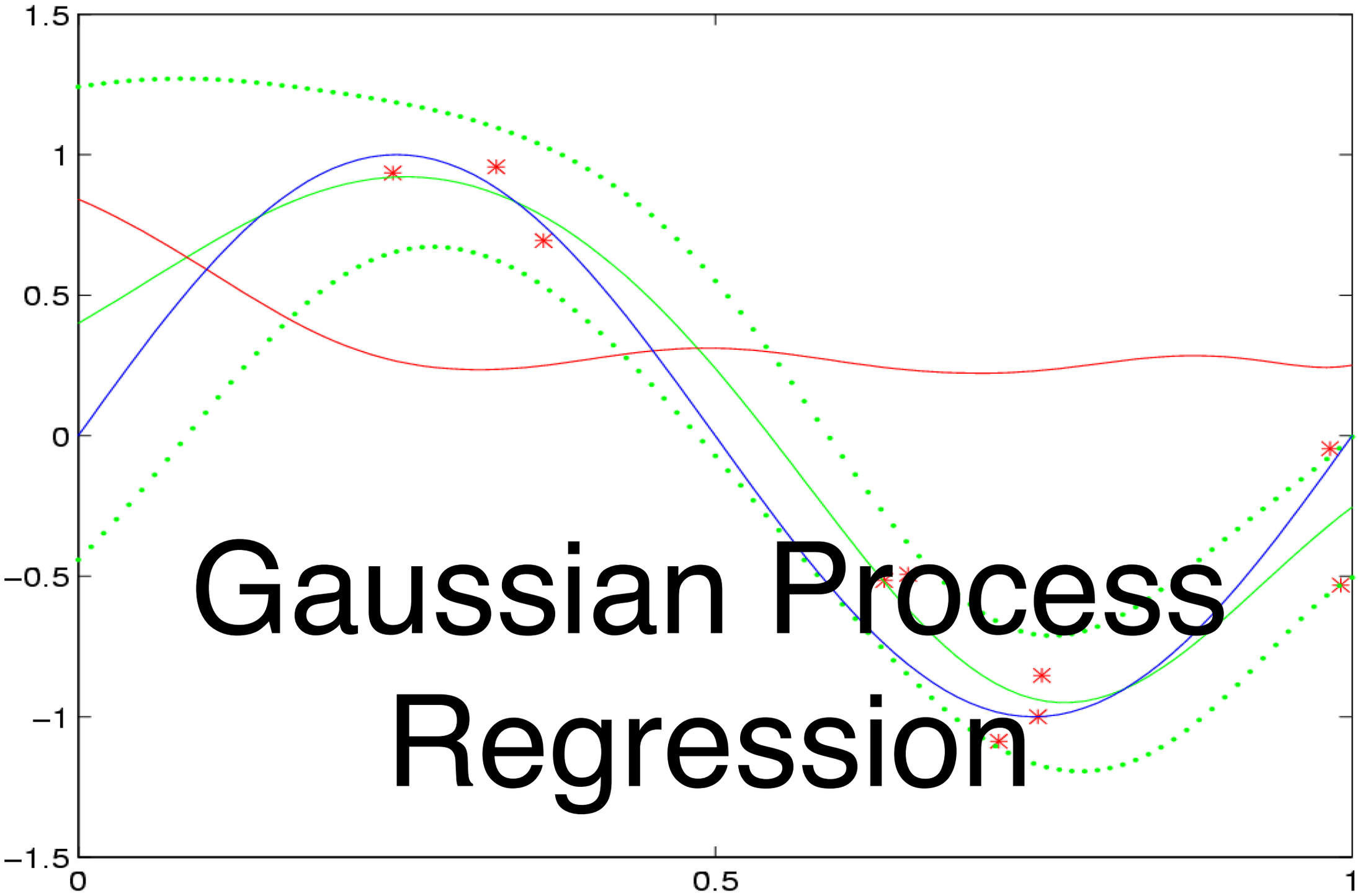
Mini Summary

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

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

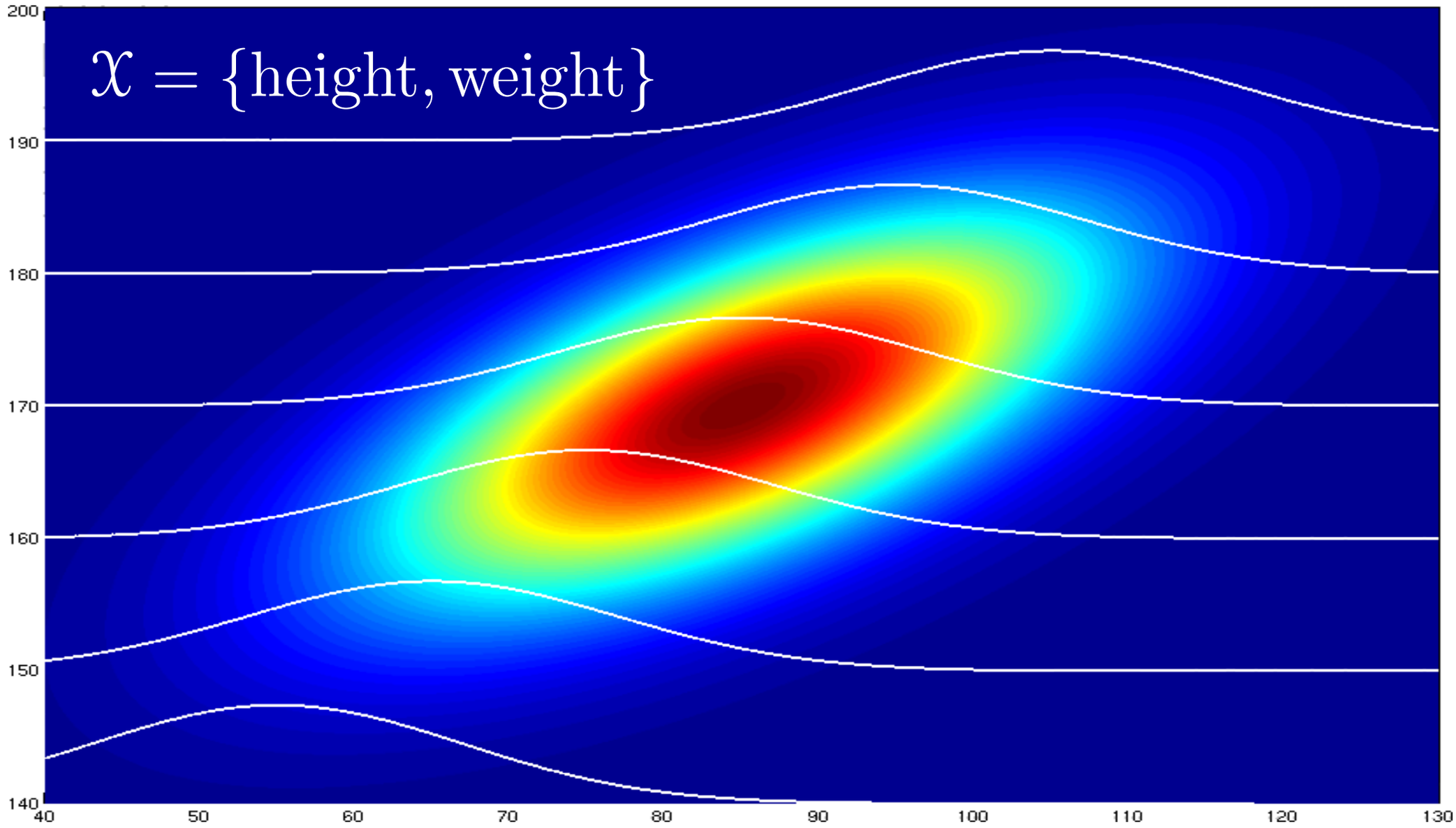
- Generates vectors of arbitrary dimensionality (via X)
- Covariance function via kernels





Gaussian Process Regression

Gaussian Processes for Inference



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

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

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

- 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

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

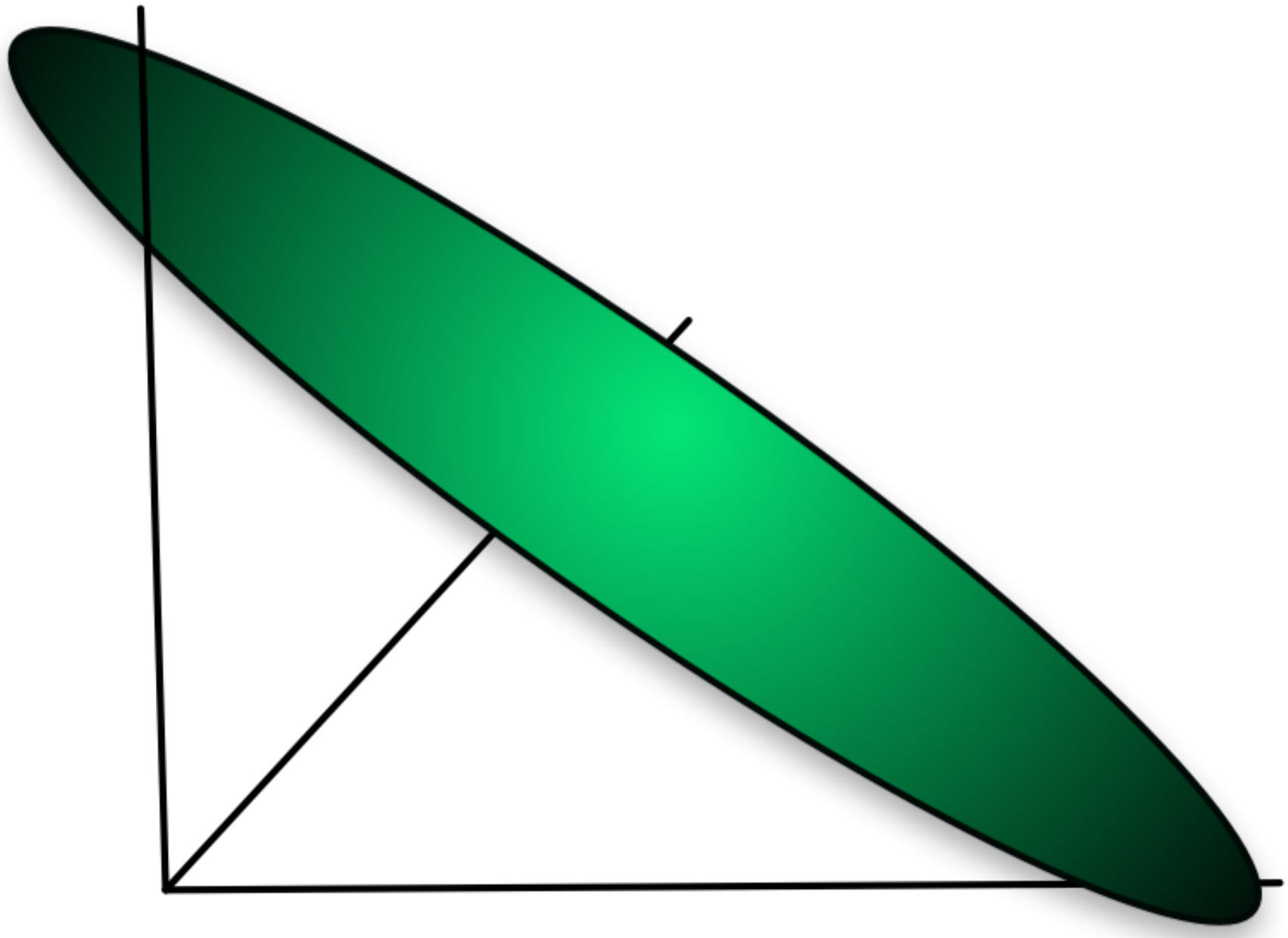
$$\tilde{\mu} = X'^\top \left[X (X^\top X)^{-1} t \right]$$

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

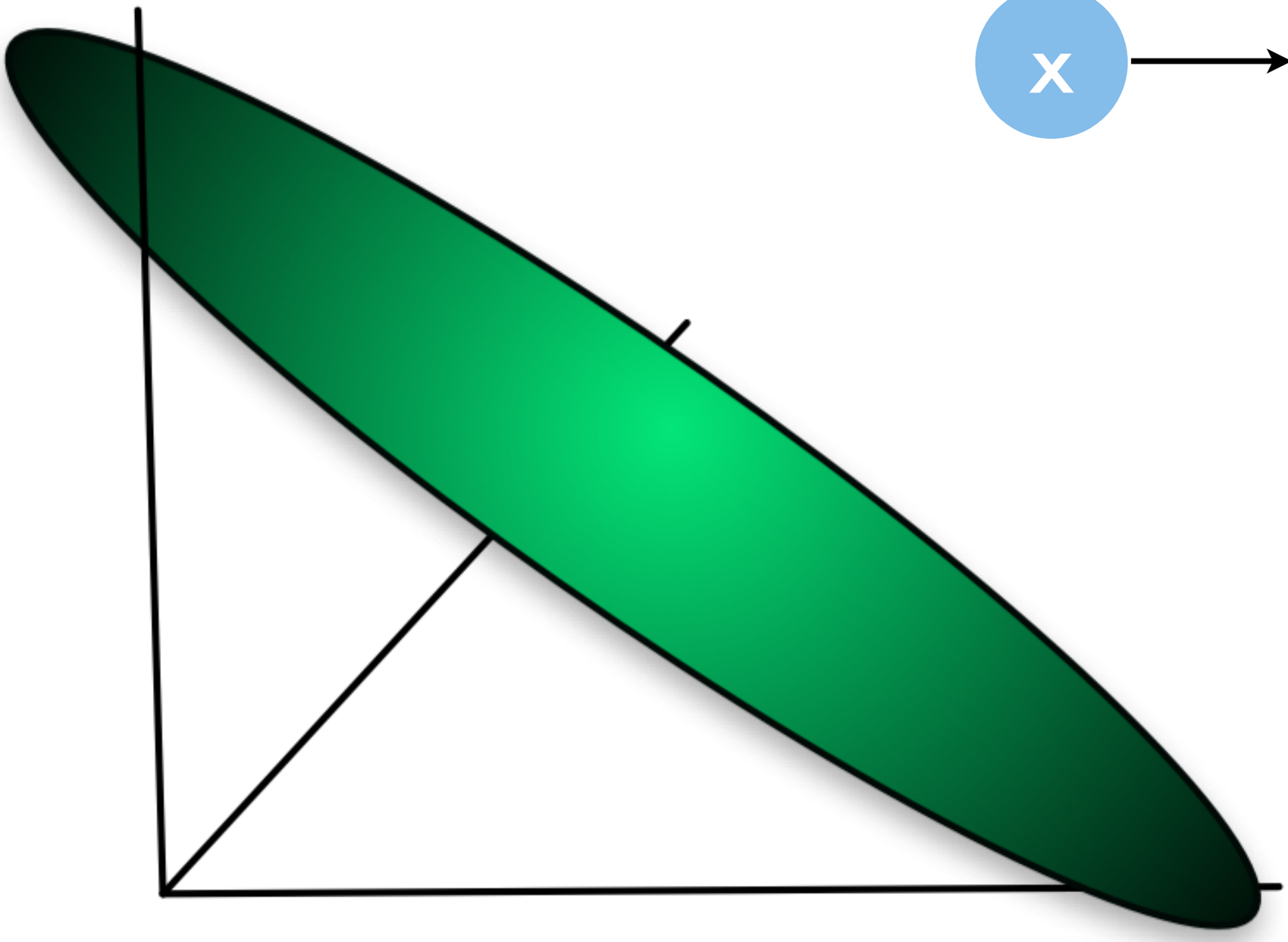
Problem

- 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.

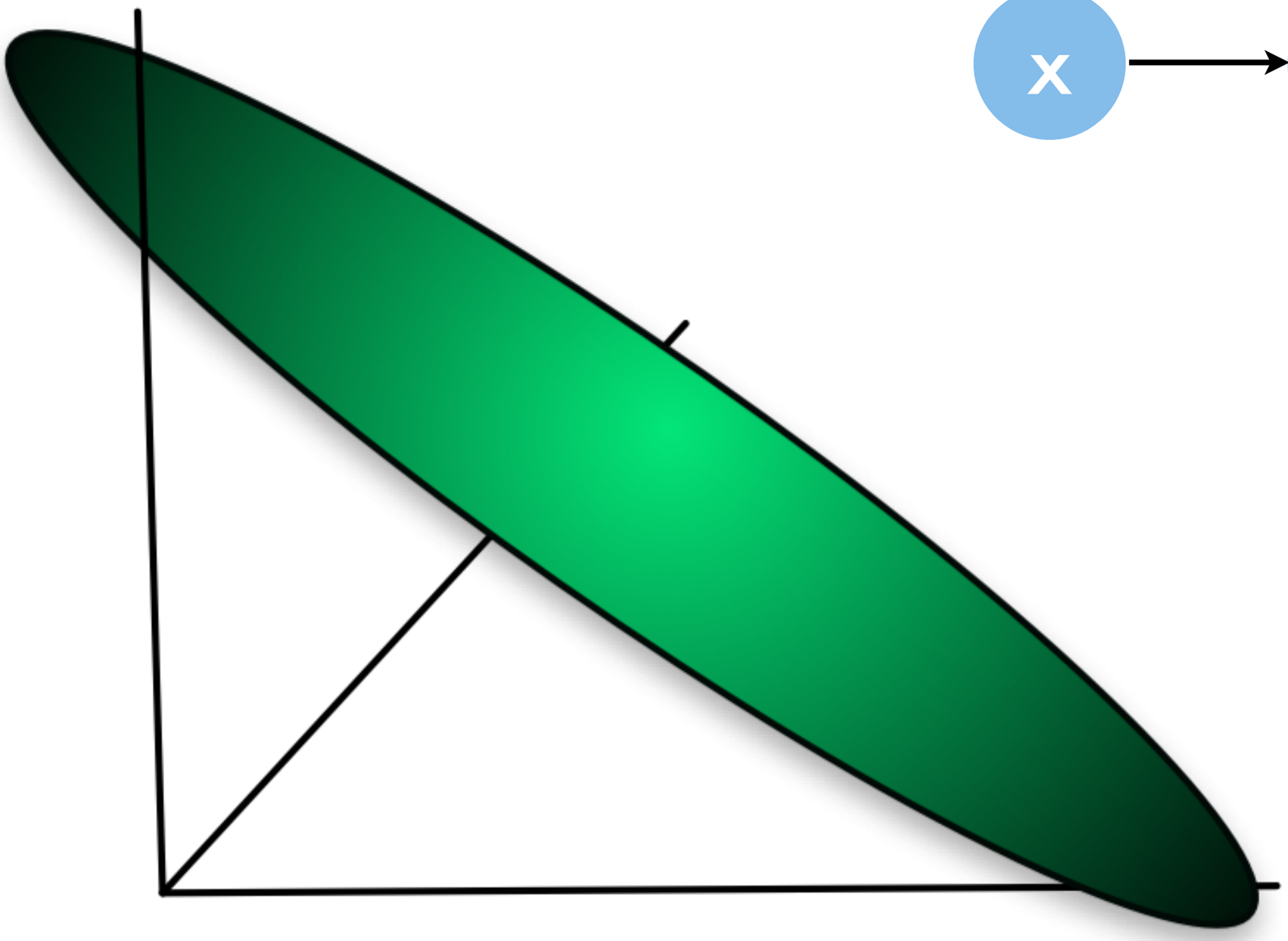
Degenerate Covariance

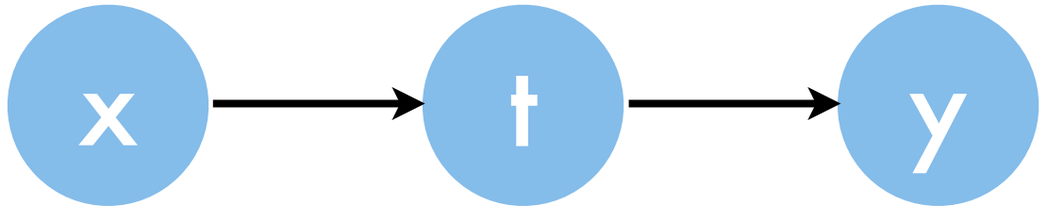


Degenerate Covariance



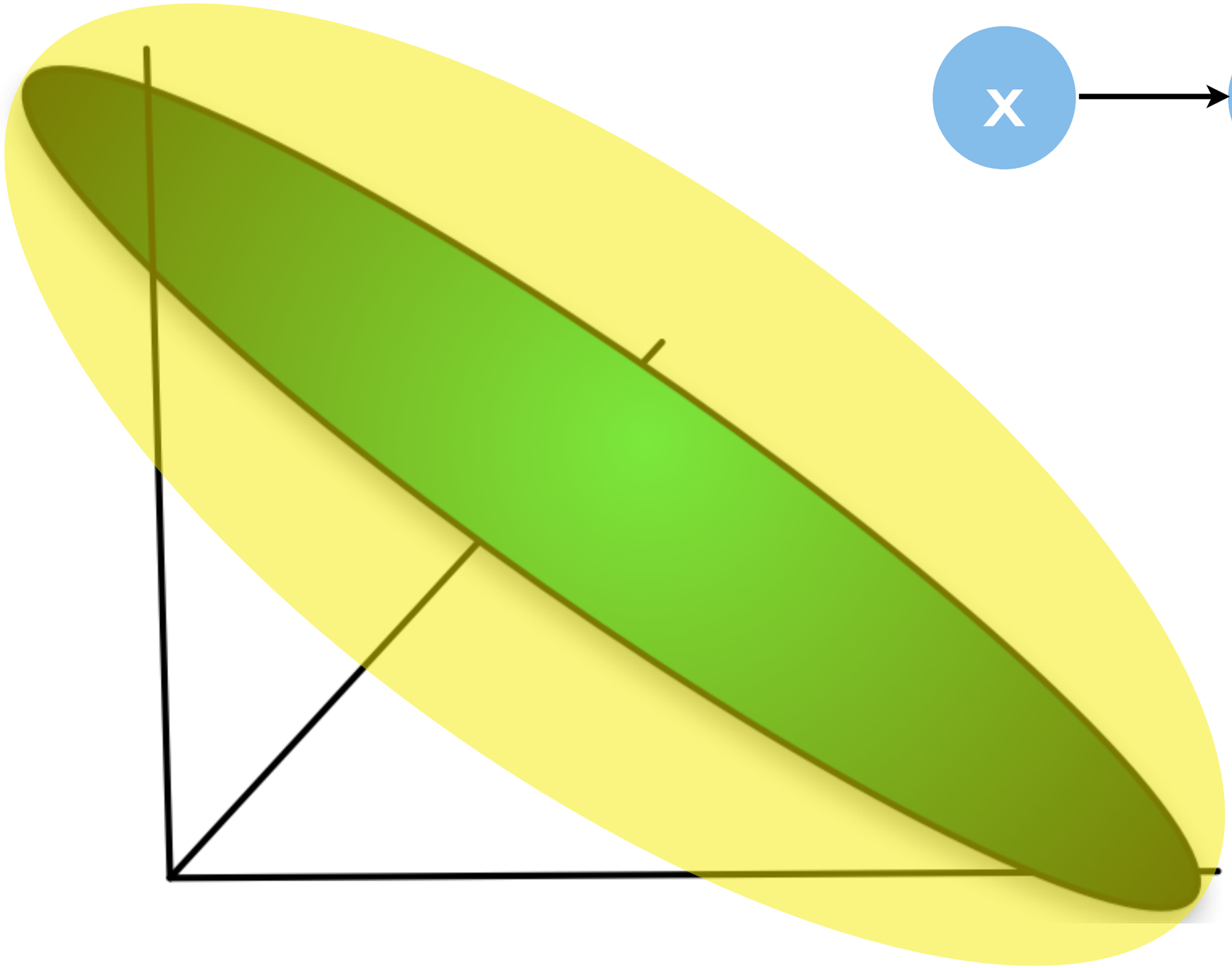
Degenerate Covariance

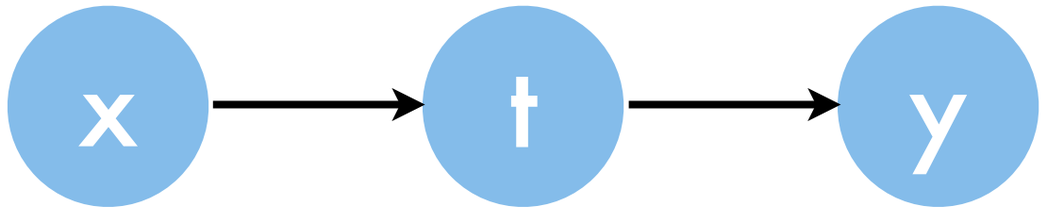




'fatten up' covariance

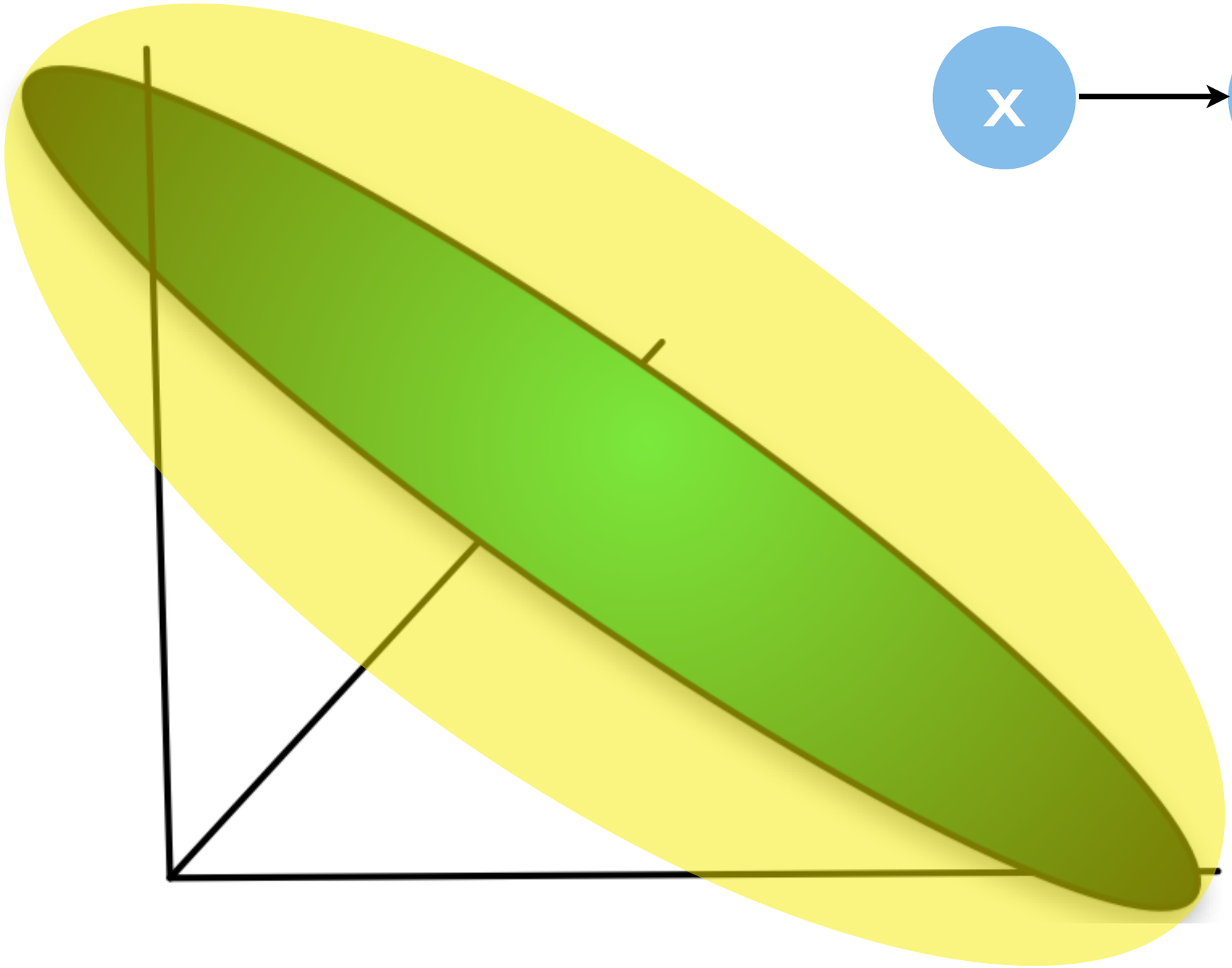
Degenerate Covariance

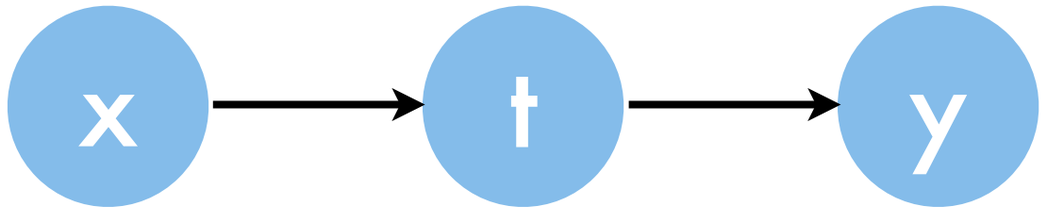




'fatten up' covariance

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

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

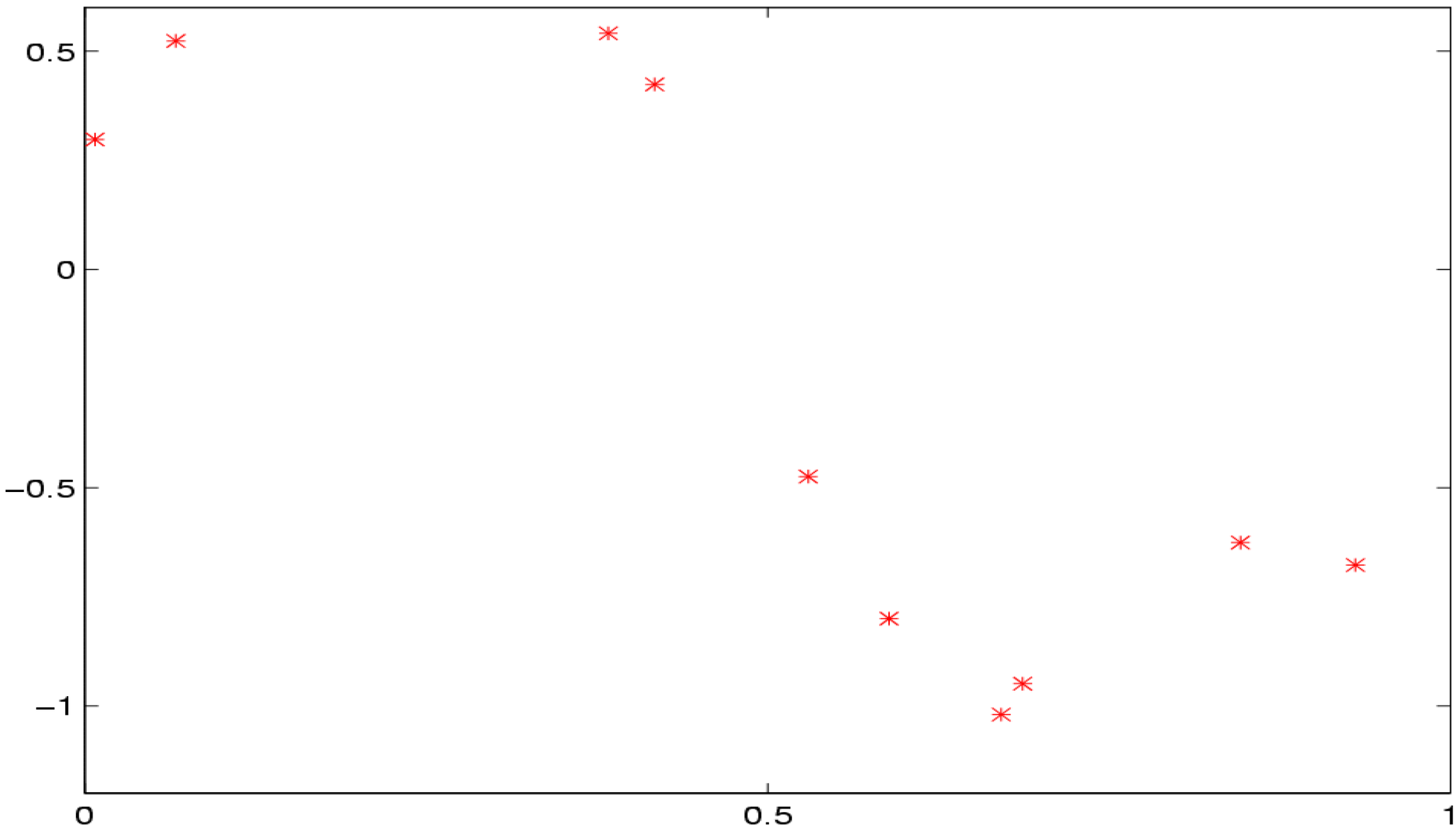
where we can now find a maximum a posteriori solution for t by maximizing the integrand (we will use this later).

Additive Normal Noise

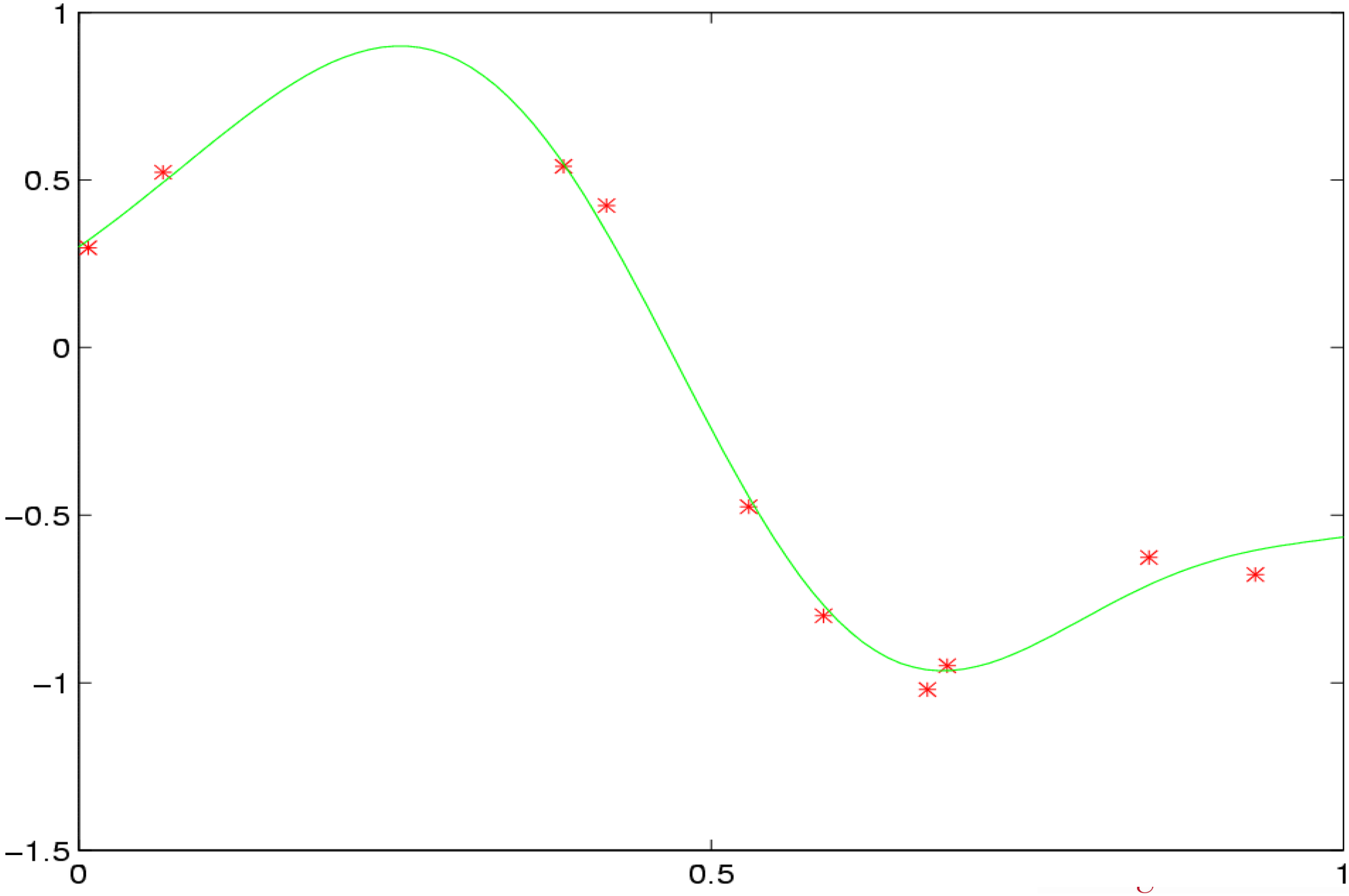
- If $\xi \sim \mathcal{N}(0, \sigma^2)$ then y is the sum of two Gaussian random variables.
- Means and variances **add up**.

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

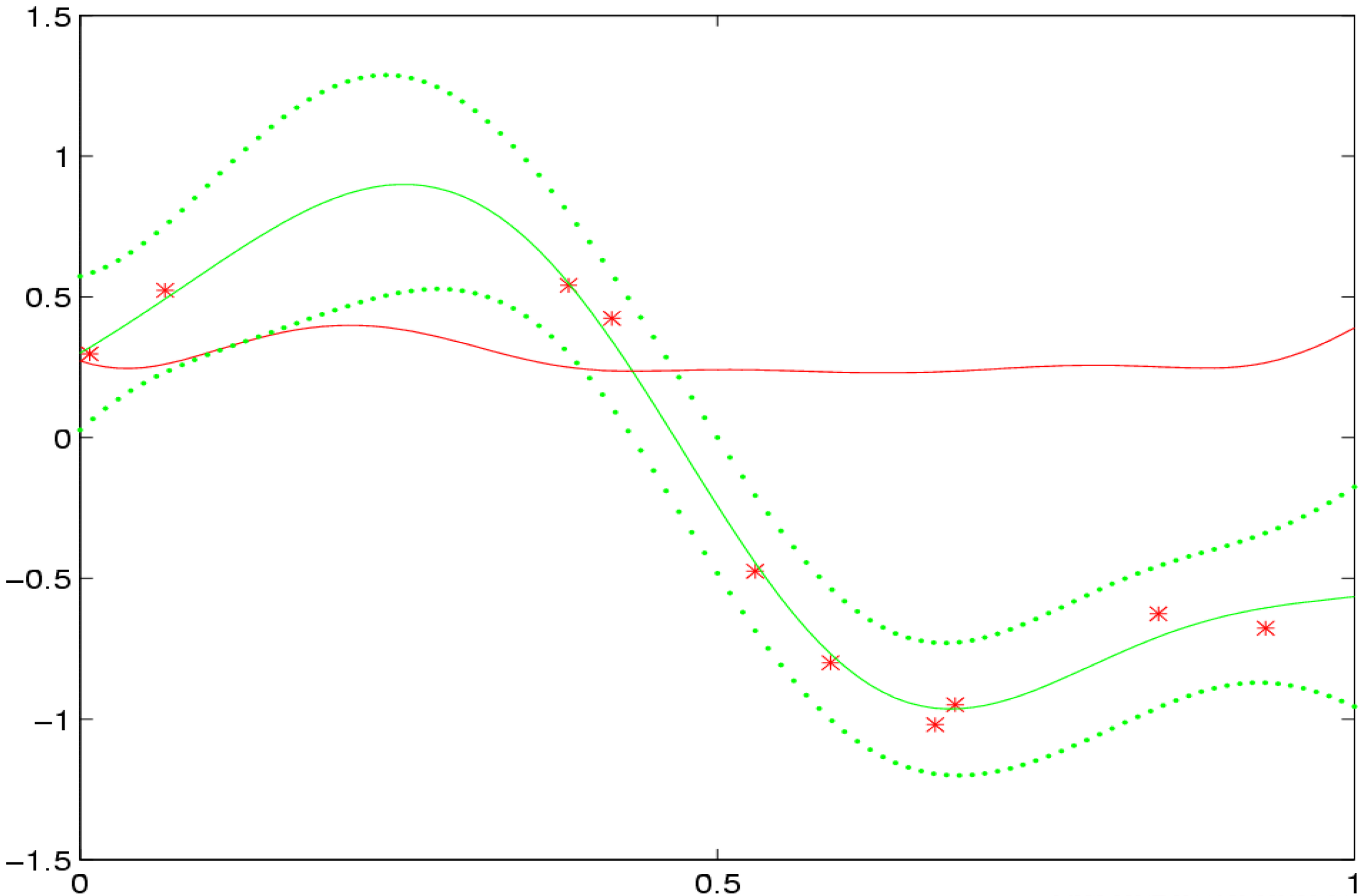
Data



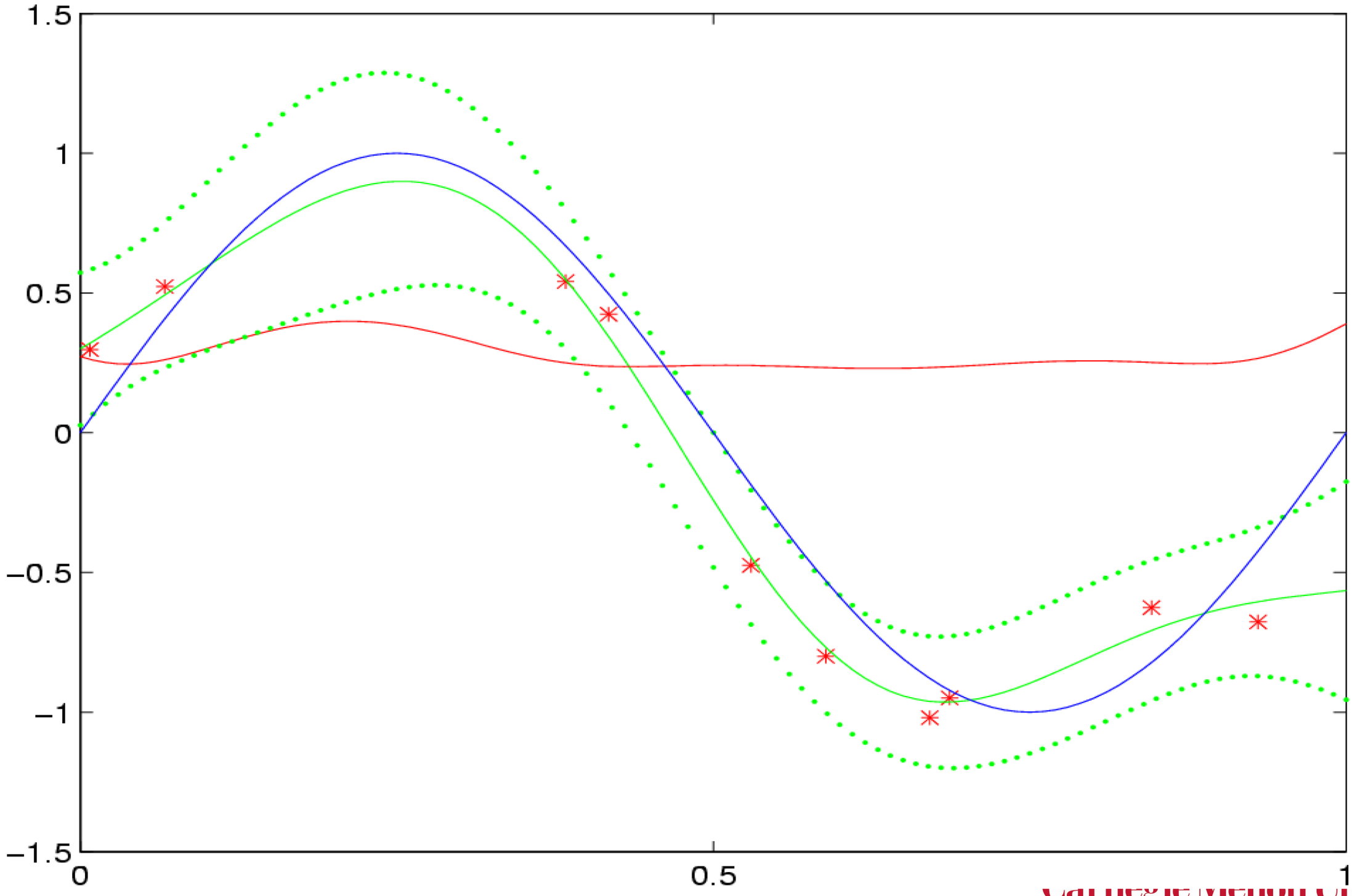
Predictive mean $x(x, X)(K(X, X) + \sigma^2 1)^{-1} y$



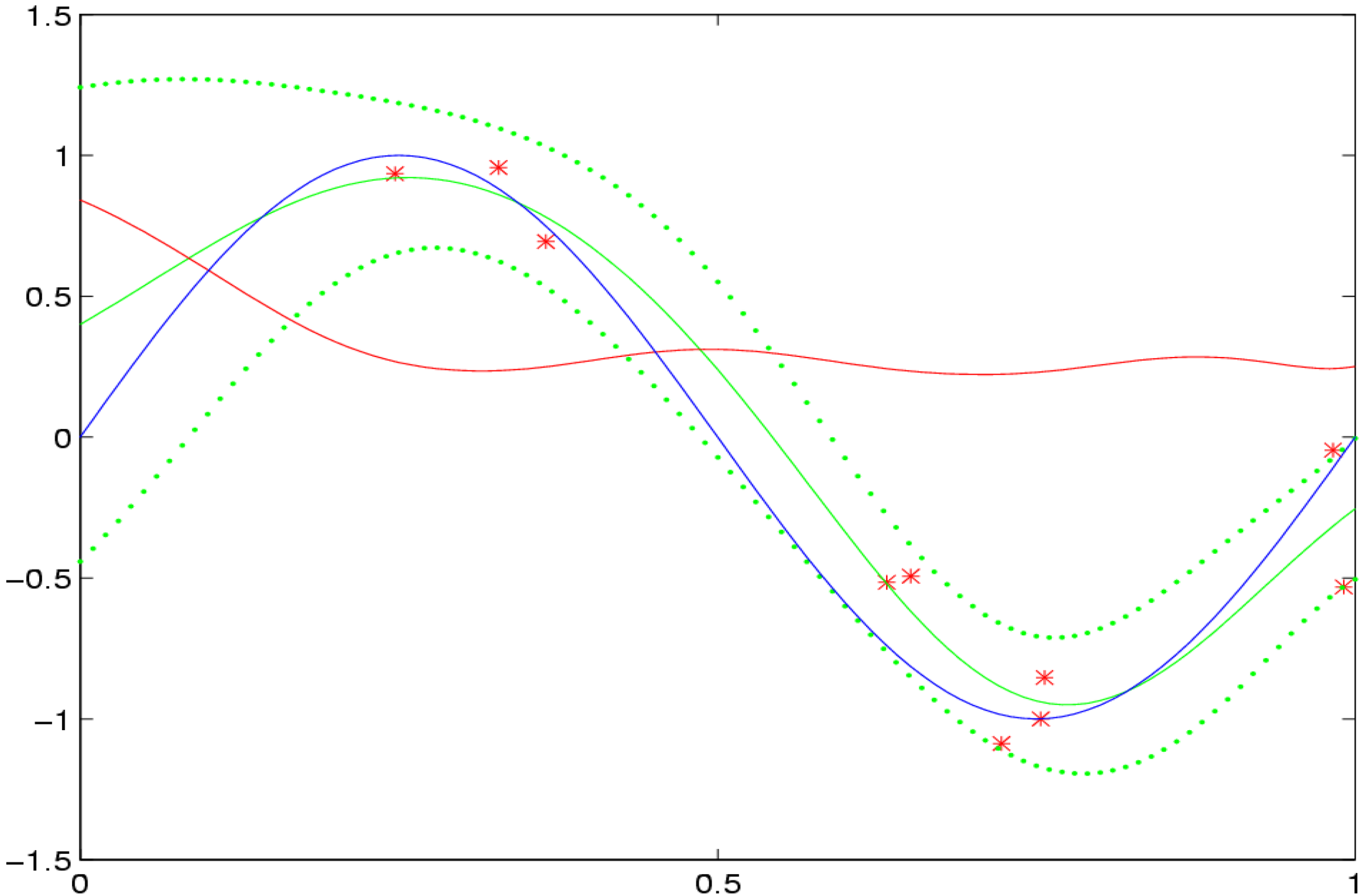
Variance



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

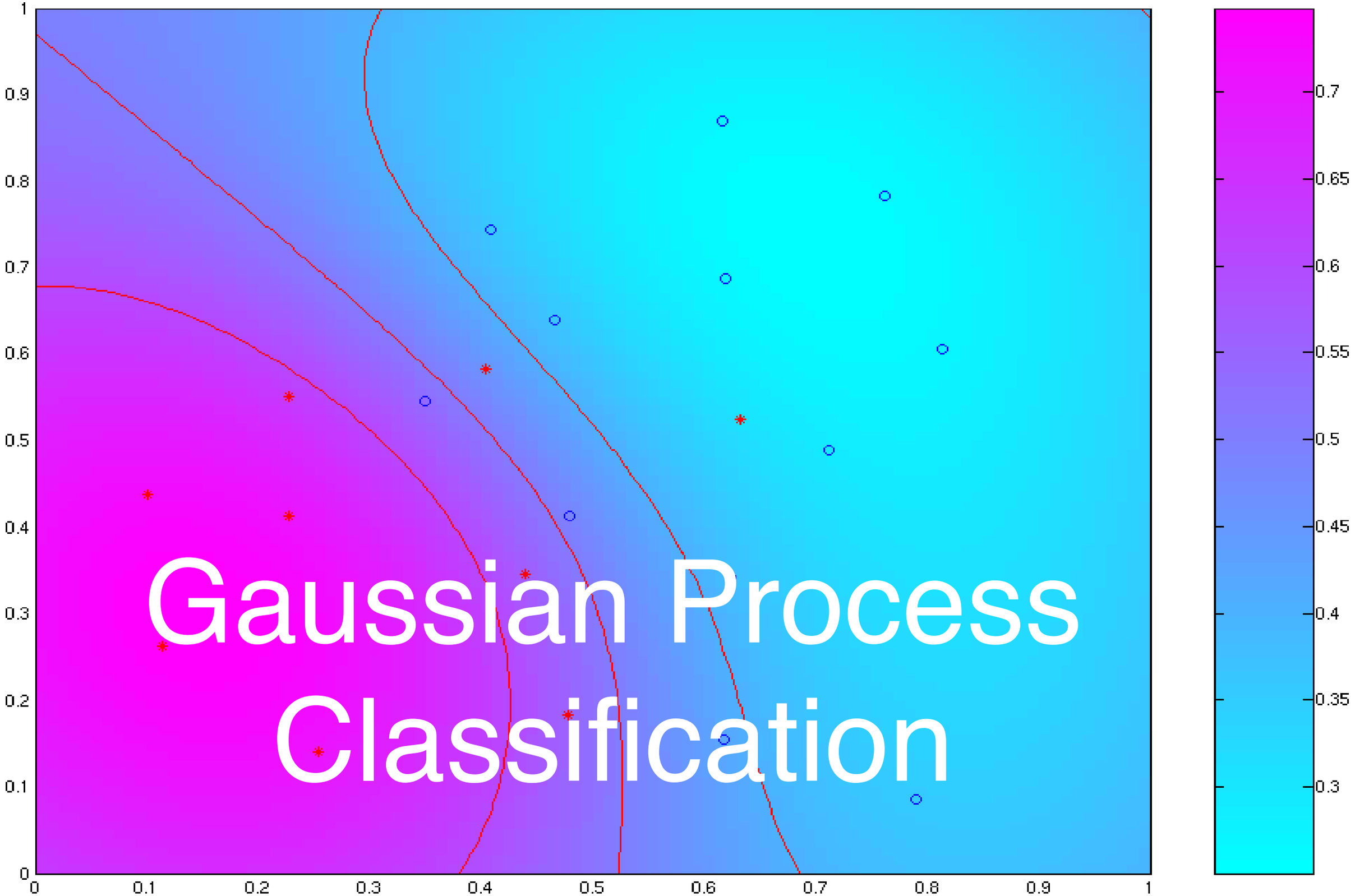
$$\mu \to \mu \text{ and } K \to K + \sigma^2 \mathbf{1}$$

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

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

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

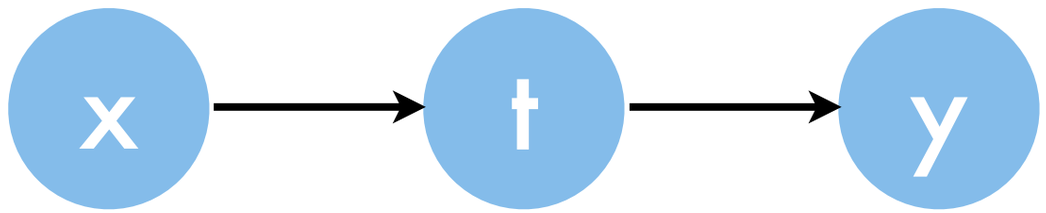
- SVM kernel is GP kernel



Gaussian Process Classification

Gaussian Process Classification

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



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

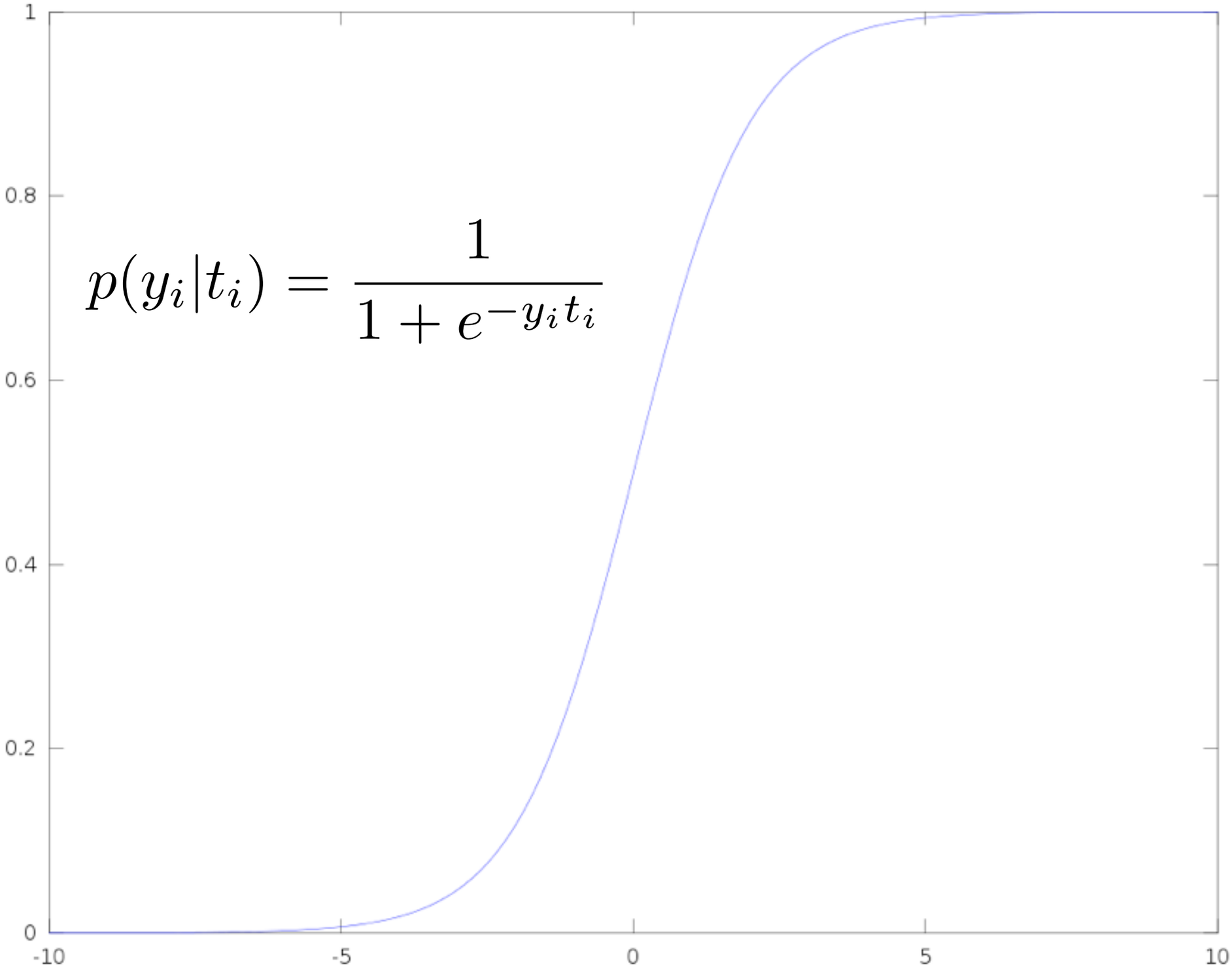
$$\text{i.e. } p(y_i|t_i) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} e^{-\frac{1}{2\sigma^2}(y_i - t_i)^2}$$

- (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

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



Recall - Binomial Distribution

- Features $\phi(x) = x$
- Domain is {-1, 1}
- Normalization

$$g(\theta) = \log\left[e^{-1\cdot\theta} + e^{1\cdot\theta}\right] = \log 2\cosh\theta$$

- Probability

$$p(x|\theta) = \exp(x\cdot\theta - g(\theta)) = \frac{e^{x\theta}}{e^{-\theta}+e^{\theta}} = \frac{1}{1+e^{-2x\theta}}$$

Logistic function

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

- Integrating out t,t'

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

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

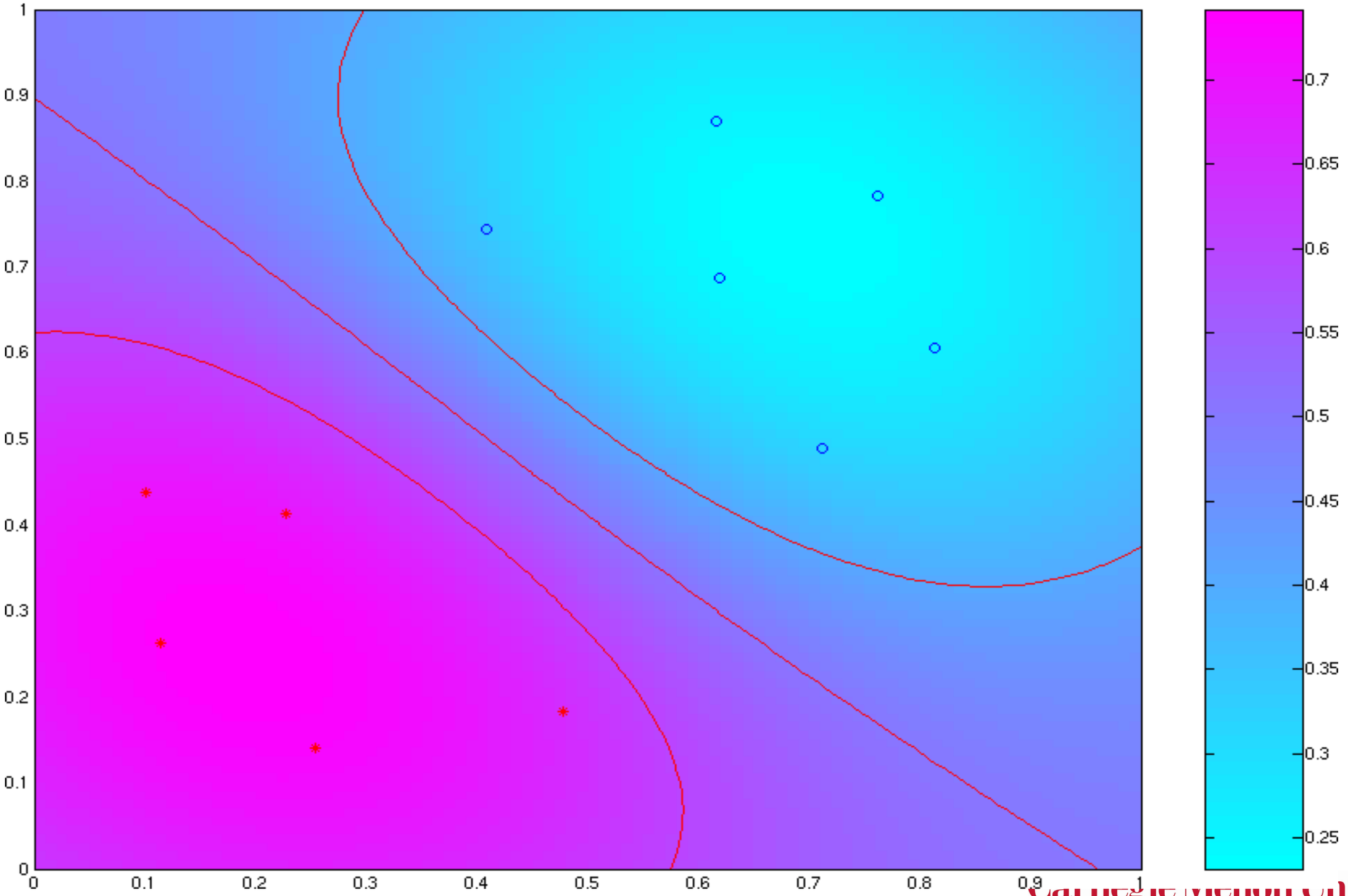
$$t' = K_{tt'}^\top K_{tt}^{-1} t$$

precompute

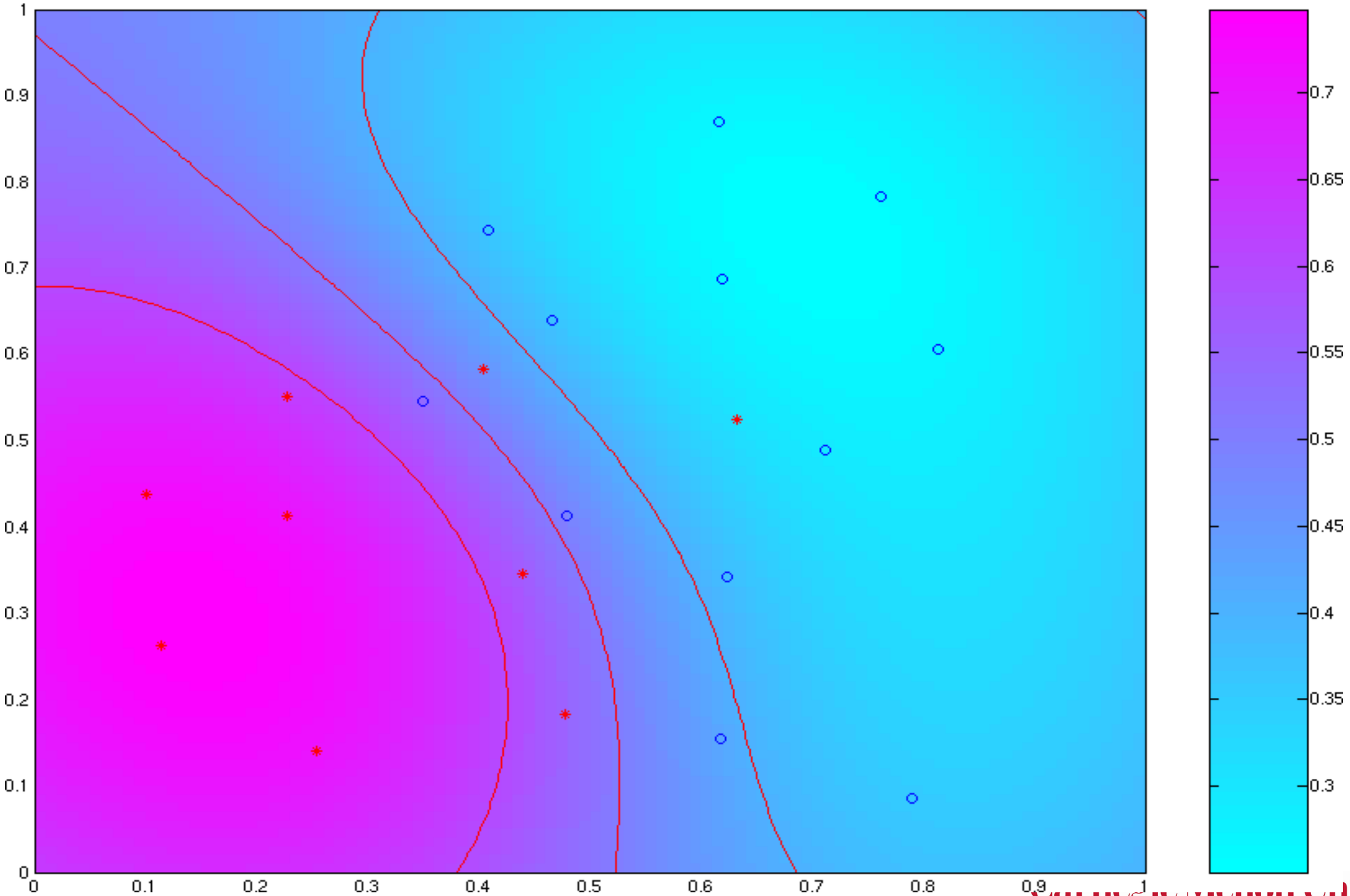
- Step 3 - estimate p(y'|t')

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

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

- 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**

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

(asymptotically) linear

(asymptotically) 0

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

Further reading

- Girosi, 1998; An equivalence between sparse approximation and Support Vector Machines ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1606.pdf
- Smola, Schoelkopf and Mueller, 1998; The connection between regularization operators and Support Vector Kernels http://alex.smola.org/teaching/berkeley2012/slides/Smola1998connection.pdf
- Schoelkopf, Smola, Mueller, 1998; Nonlinear Component Analysis as Kernel Eigenvalue Problem http://www.mitpressjournals.org/doi/abs/10.1162/089976698300017467
- Scholkopf, Smola, Herbrich, 2001; A Generalized Representer Theorem alex.smola.org/papers/2001/SchHerSmo01.pdf
- Teo, Globerson, Roweis, Smola, 2008; Convex Learning with Invariances http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2007_1047.pdf
- Rasmussen, 2006; Gaussian Processes for Machine Learning http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.3414