Scalable Machine Learning

5. (Generalized) Linear Models

Alex Smola
Yahoo! Research and ANU

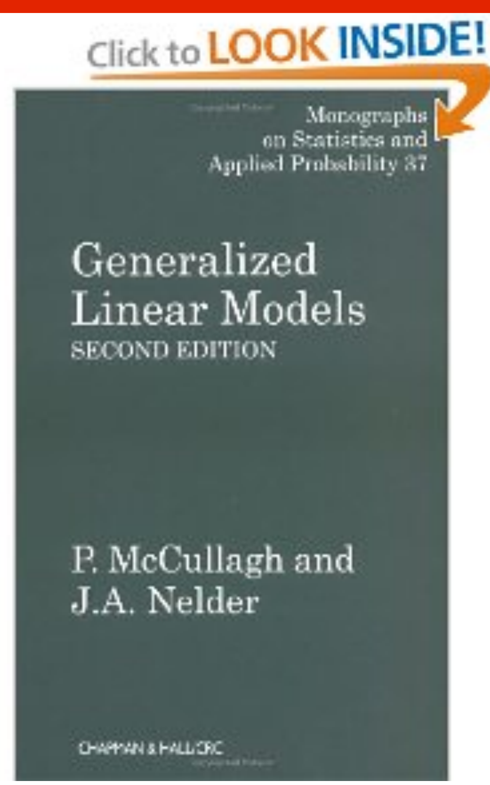
http://alex.smola.org/teaching/berkeley2012
Stat 260 SP 12

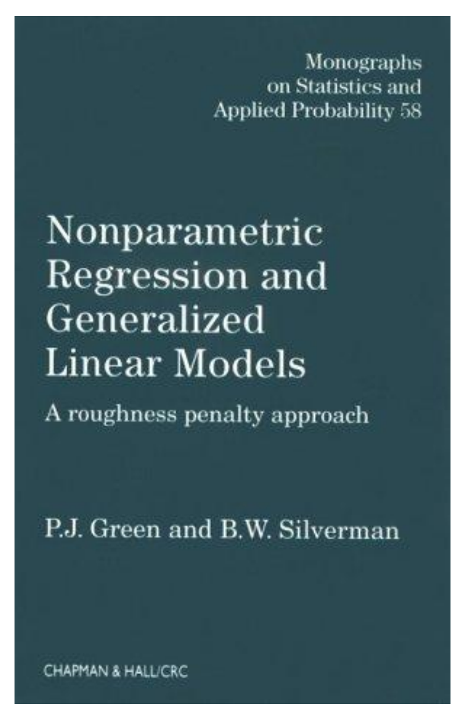
Administrative stuff

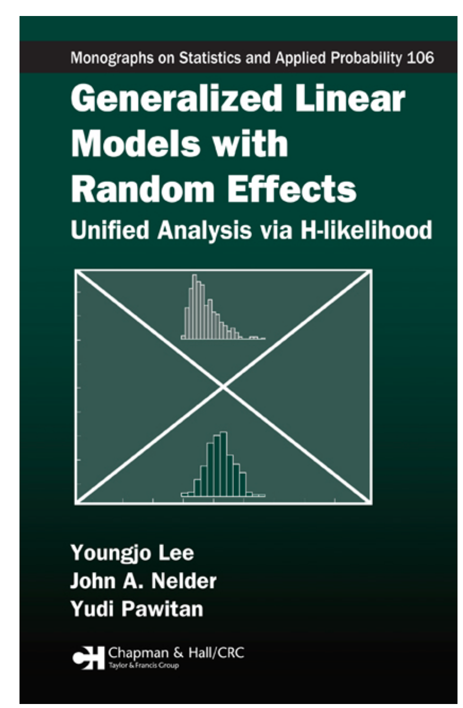
- Solutions will be posted by tomorrow
- New problem set will be available by tomorrow

- Midterm project presentations are on March 13
  - Describe what you will do
  - Why it’s important
  - What you’ve achieved so far
  - Show why you think you’re going to succeed
  - 10 minutes per team (6 slides maximum)
  - Up to 10 pages supporting documentation

5. (Generalized) Linear Models





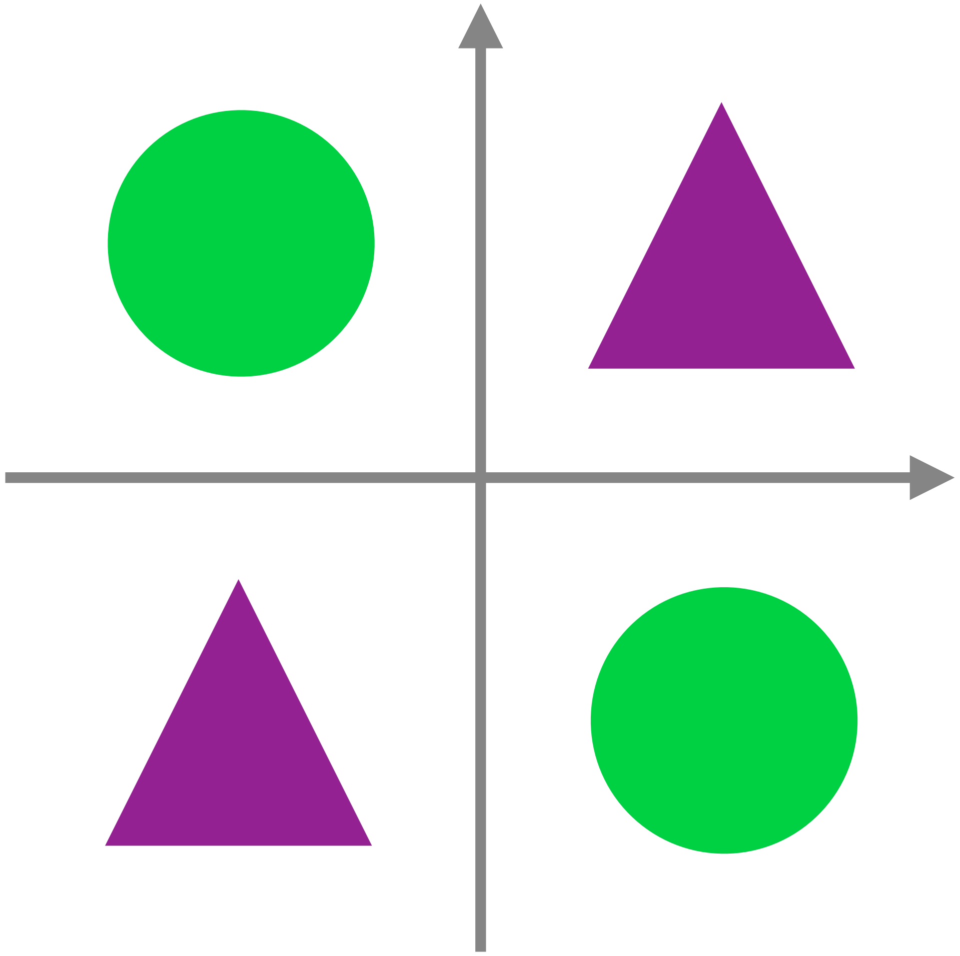


(Generalized) Linear Models

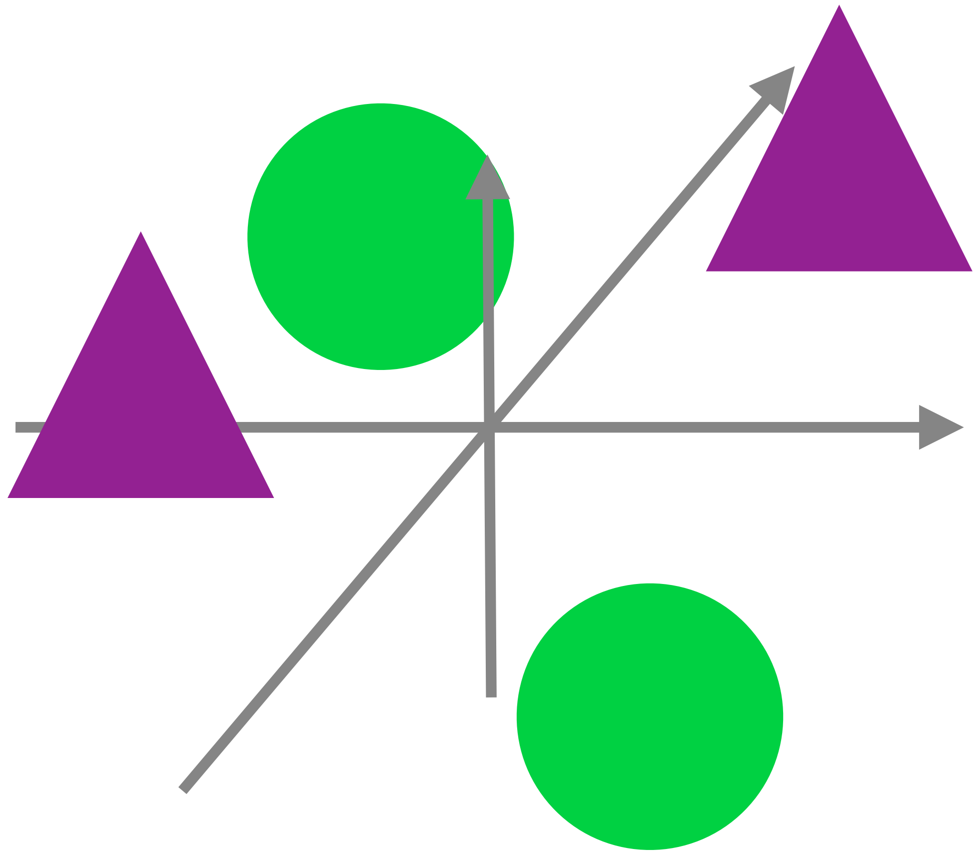
- Kernel trick
  - Simple kernels
  - Kernel PCA
  - Mean Classifier
- Support Vectors
  - Support Vector Machine classification
  - Regression
  - Logistic regression
  - Novelty detection
- Gaussian Process Estimation
  - Regression
  - Classification
  - Heteroscedastic Regression

Kernels - a Preview

Solving XOR



(x_1, x_2)



$(x_1, x_2, x_1 x_2)$

- XOR not linearly separable
- Mapping into 3 dimensions makes it easily solvable

Feature Space Mapping

- Naive Nonlinearization Strategy
  - Express data x in terms of features $\phi(x)$
  - Solve problem in feature space
  - Requires explicit feature computation
- Kernel trick
  - Write algorithm in terms of inner products
  - Replace $\langle x, x' \rangle$ by $k(x, x') := \langle \phi(x), \phi(x') \rangle$
  - Works well for dimension-insensitive methods
  - Kernel matrix K is positive semidefinite

SVM with a polynomial Kernel visualization

Created by:

Udi Aharoni

Polynomial Kernels

- Linear

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

- Quadratic

$$k(x,x') := \left\langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), ({x'_1}^2, {x'_2}^2, \sqrt{2}x'_1x'_2) \right\rangle = \langle x, x' \rangle^2$$

- Homogeneous polynomial

$$k(x,x') := \langle x, x' \rangle^p = \sum_{|\alpha|=p} \prod_i \alpha_i!(x_i x'_i)^{\alpha_i} \text{ with } \alpha \in \mathbb{N}_0^d$$

inner product

- Inhomogeneous polynomial

$$k(x,x') := (\langle x, x' \rangle + c)^p = \sum_{i=0}^{p} \binom{p}{i} \langle x, x' \rangle^i$$

More Kernels

- Gaussian Kernel

$$k(x, x') := \exp\left(-\gamma \|x - x'\|^2\right)$$

  can check that this is convolution of Gaussians

- Brownian Bridge

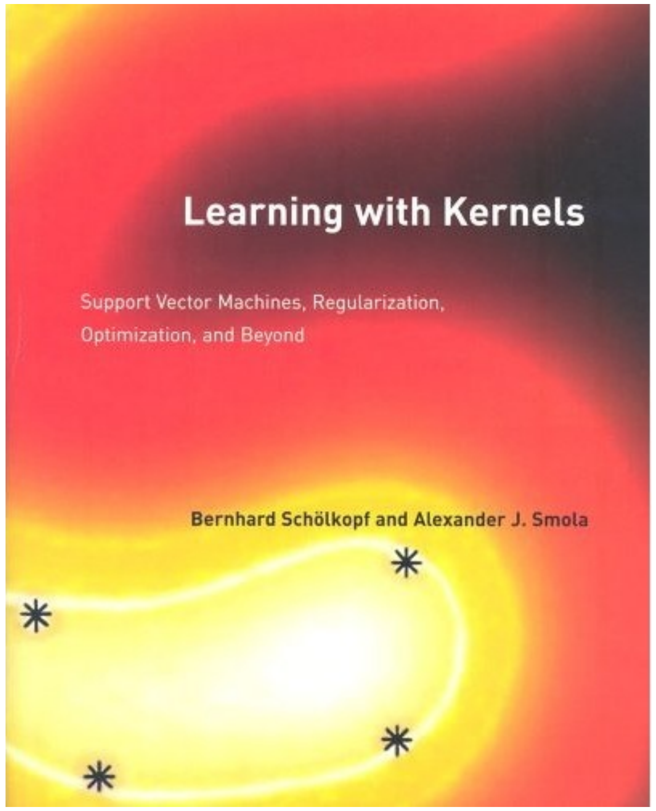
$$k(x, x') := \min(x, x') \text{ for } x, x' \geq 0$$

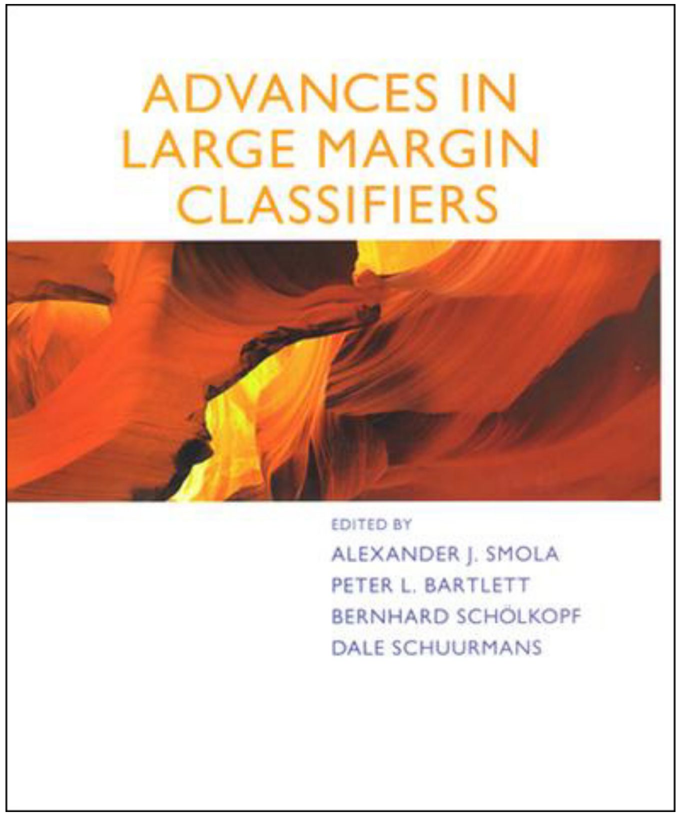
- Set intersection

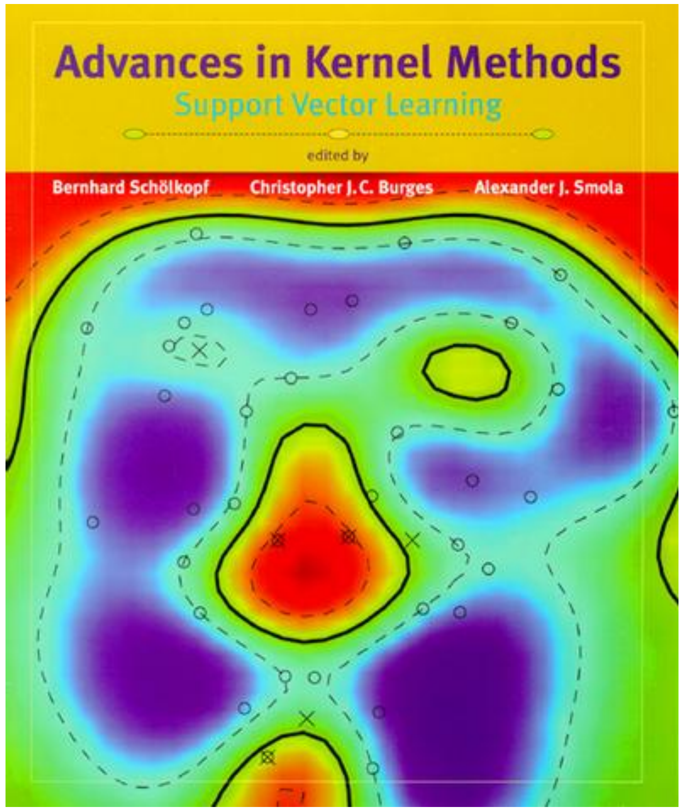
$$k(A, B) := |A \cap B|$$

- Strings, more fancy set kernels, graphs, etc.

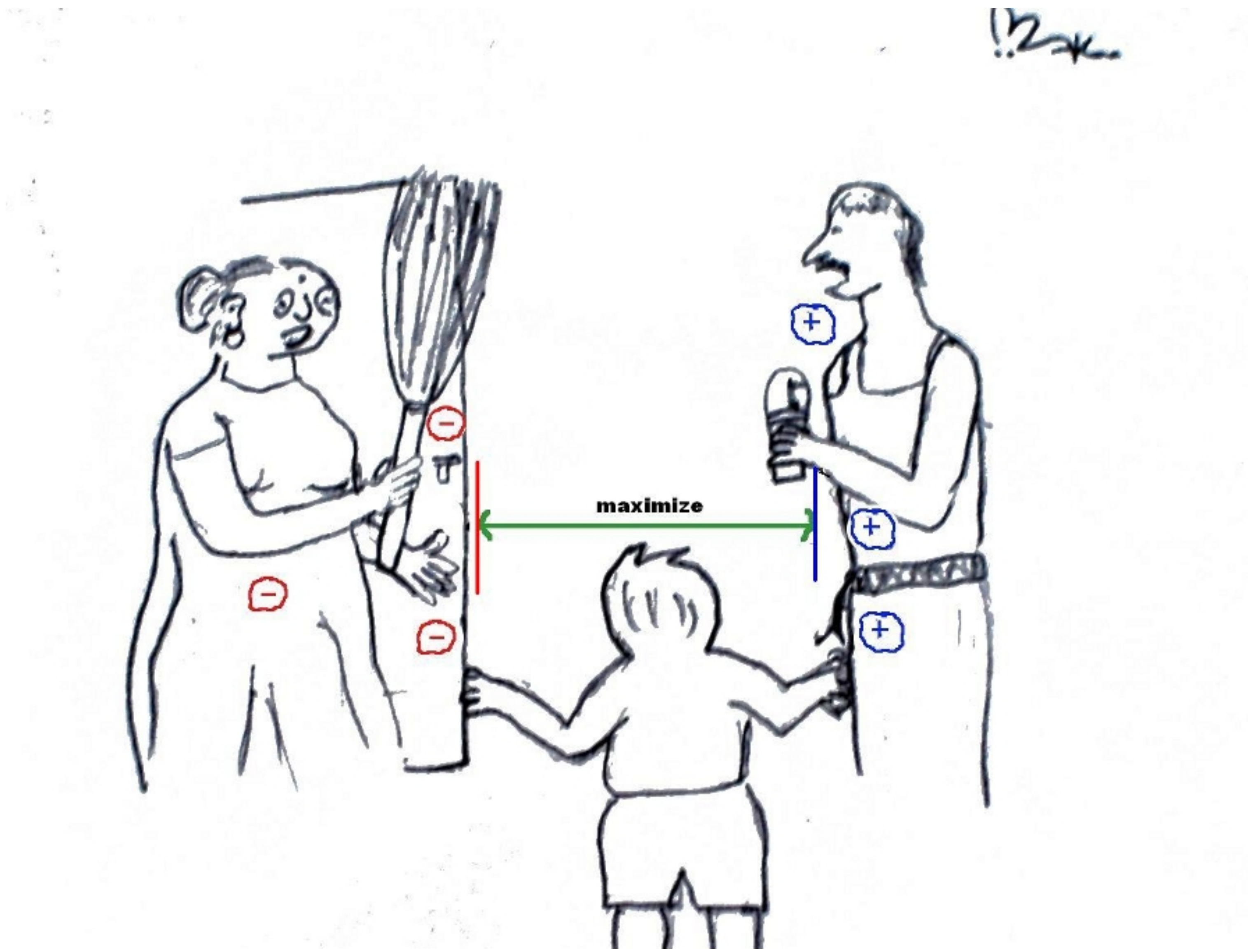
Support Vector Machines





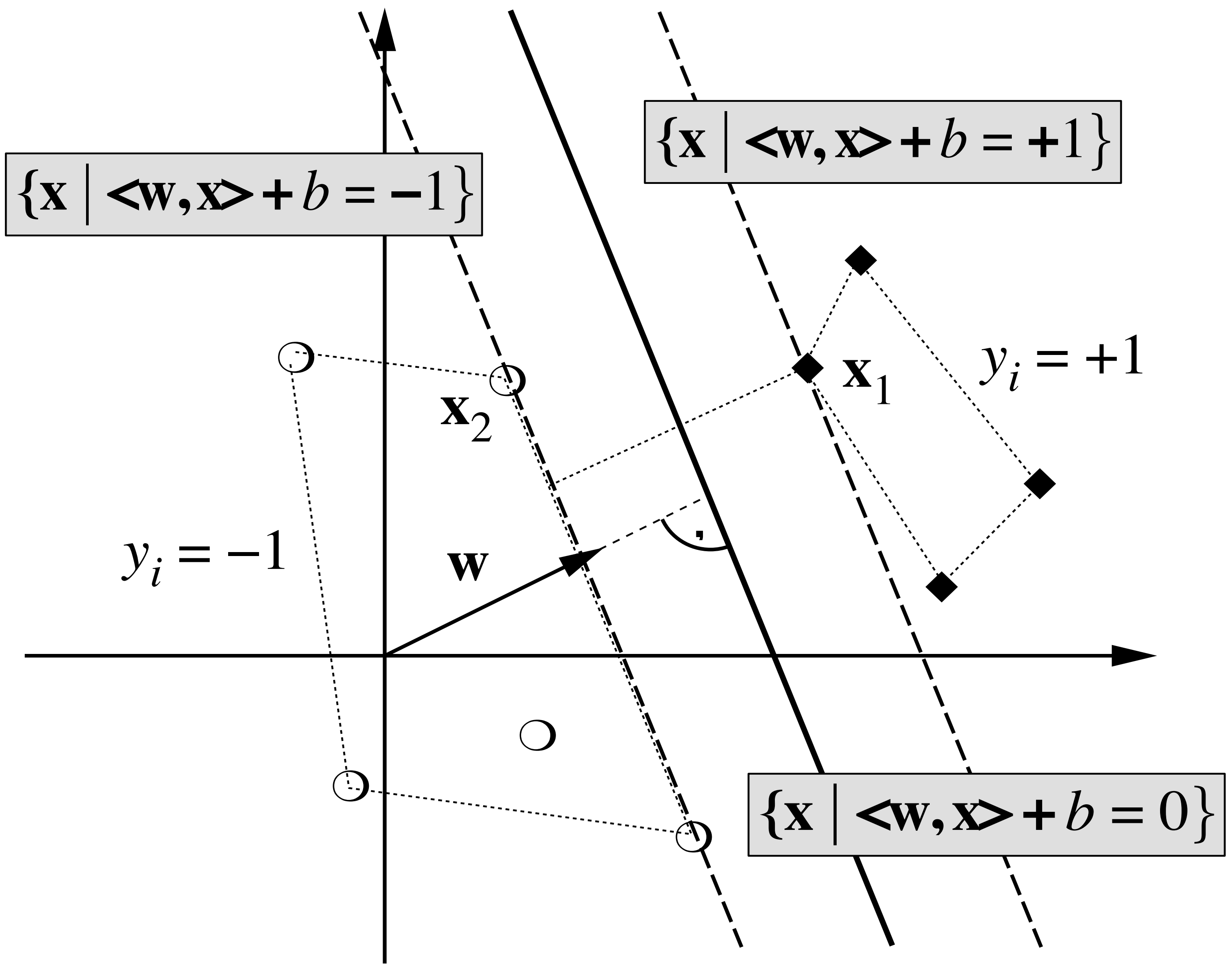


Classification



http://maktoons.blogspot.com/2009/03/support-vector-machine.html

Support Vectors



$$\langle w, x_1 \rangle + b = 1$$

$$\langle w, x_2 \rangle + b = -1$$

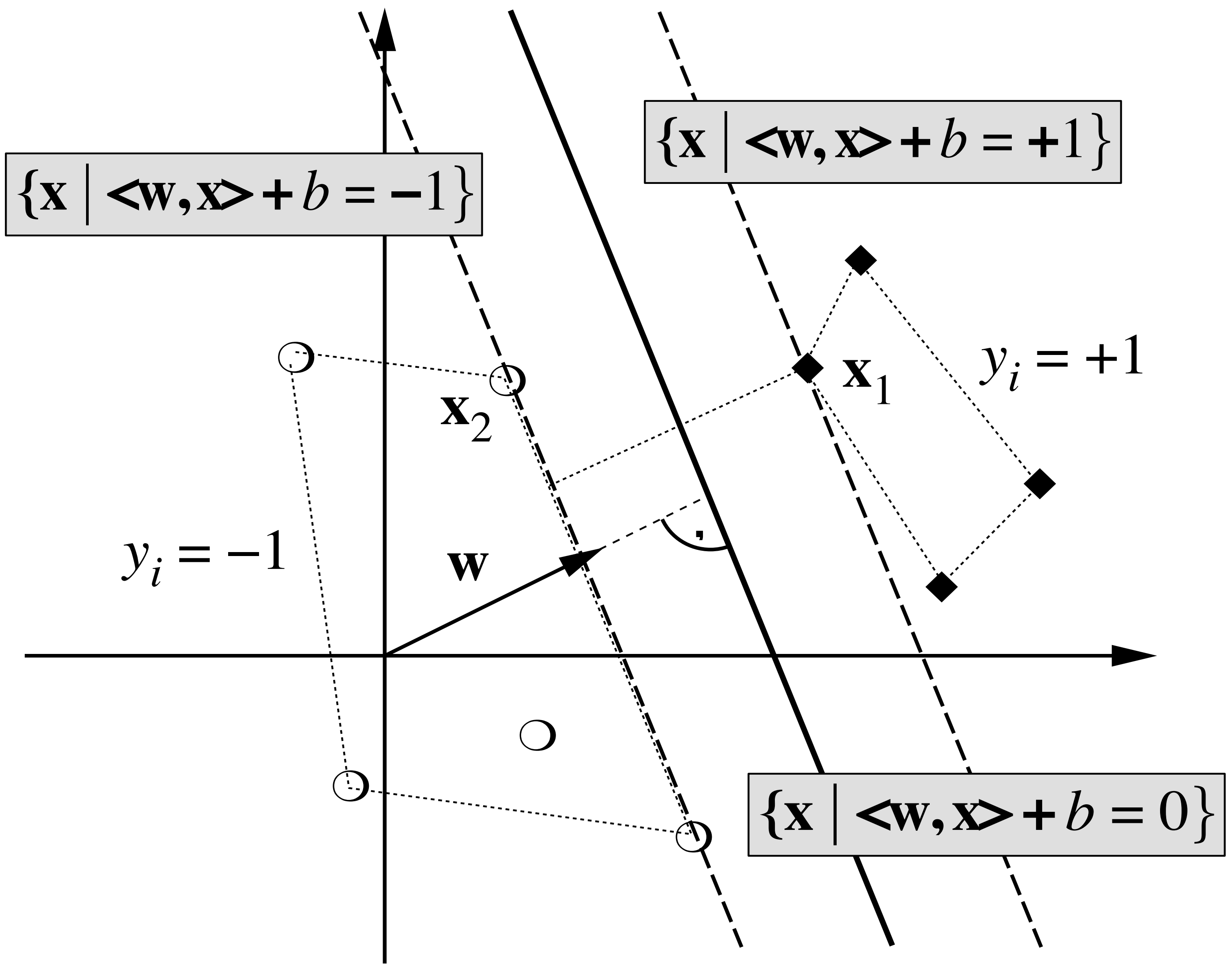
$$\text{hence } \langle w, x_1 - x_2 \rangle = 2$$

$$\text{hence } \left\langle \frac{w}{\|w\|}, x_1 - x_2 \right\rangle = \frac{2}{\|w\|}$$

margin

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \text{ subject to } y_i \left[\langle w, x_i \rangle + b \right] \geq 1$$

Support Vectors



dual problem

$$\underset{\alpha}{\text{minimize}} \ \frac{1}{2}\alpha^\top K \alpha - 1^\top \alpha$$

$$\text{subject to} \ \sum_i \alpha_i y_i = 0$$

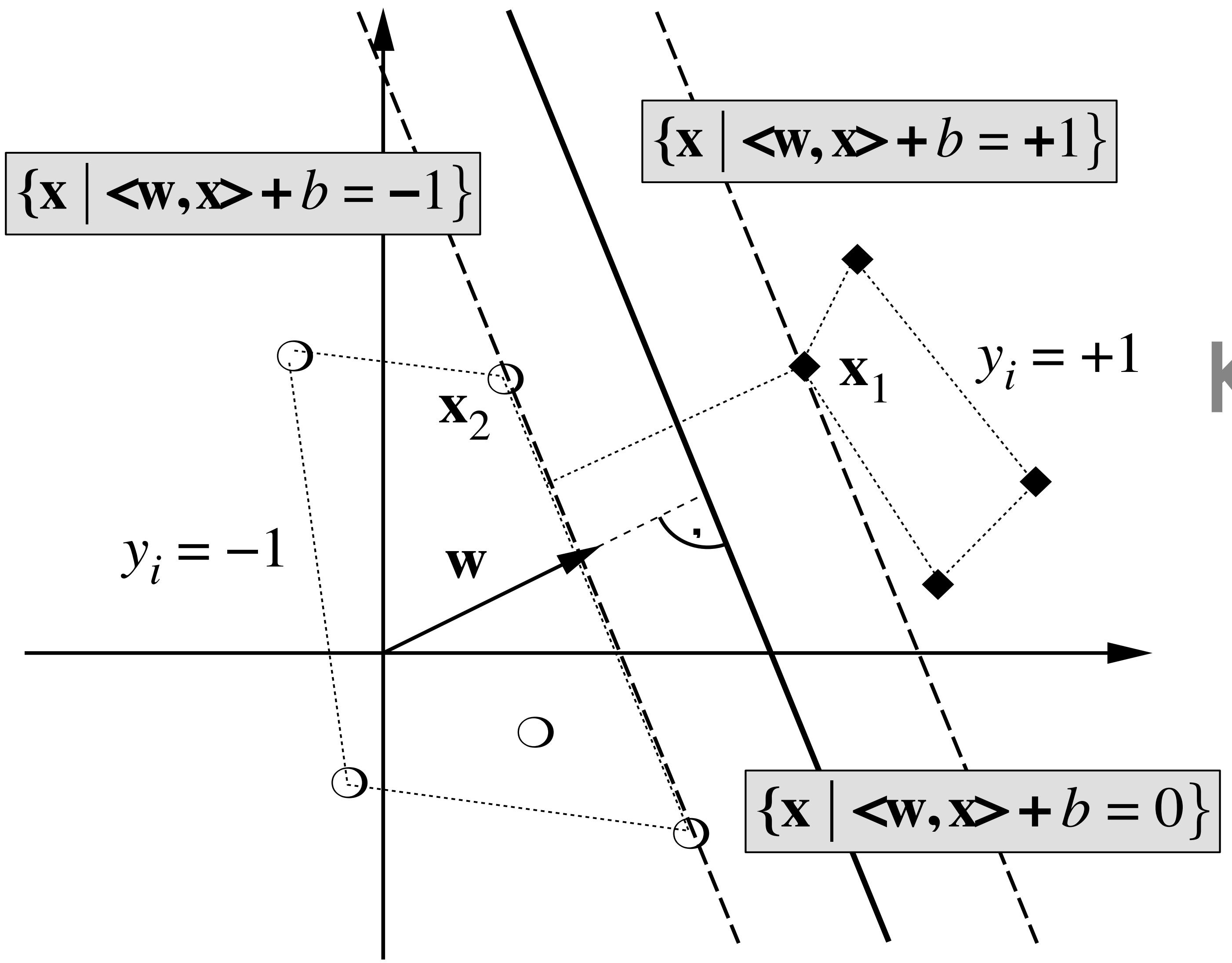
$$\alpha_i \geq 0$$

$$K_{ij} = y_i y_j \langle x_i, x_j \rangle$$

$$w = \sum_i \alpha_i y_i x_i$$

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \ \text{subject to} \ y_i \left[\langle w, x_i \rangle + b\right] \geq 1$$

Karush Kuhn Tucker conditions



KKT optimality condition

$$\alpha_i \left[y_i(\langle x_i, w \rangle + b) \geq 1 \right] = 0$$

$$y_i(\langle x_i, w \rangle + b) > 1 \text{ implies } \alpha_i = 0$$

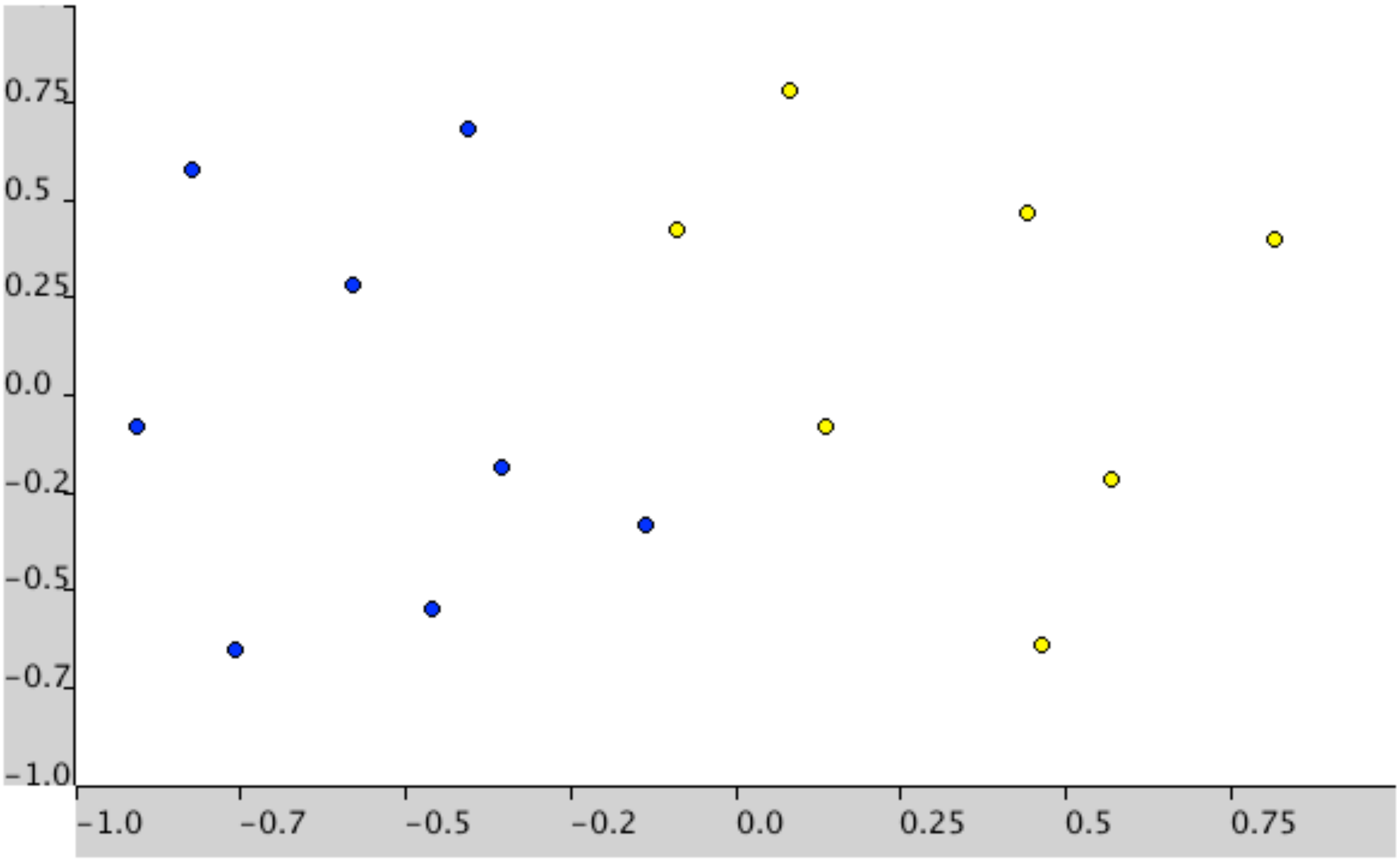
$$\alpha_i > 0 \text{ implies } y_i(\langle x_i, w \rangle + b) = 1$$

Properties

- Weight vector w as weighted linear combination of instances
- Only points on margin matter (we can ignore the rest and get same solution)
- Only inner products matter
  - Quadratic program
  - We can replace the inner product by a kernel
- Keeps instances away from the margin

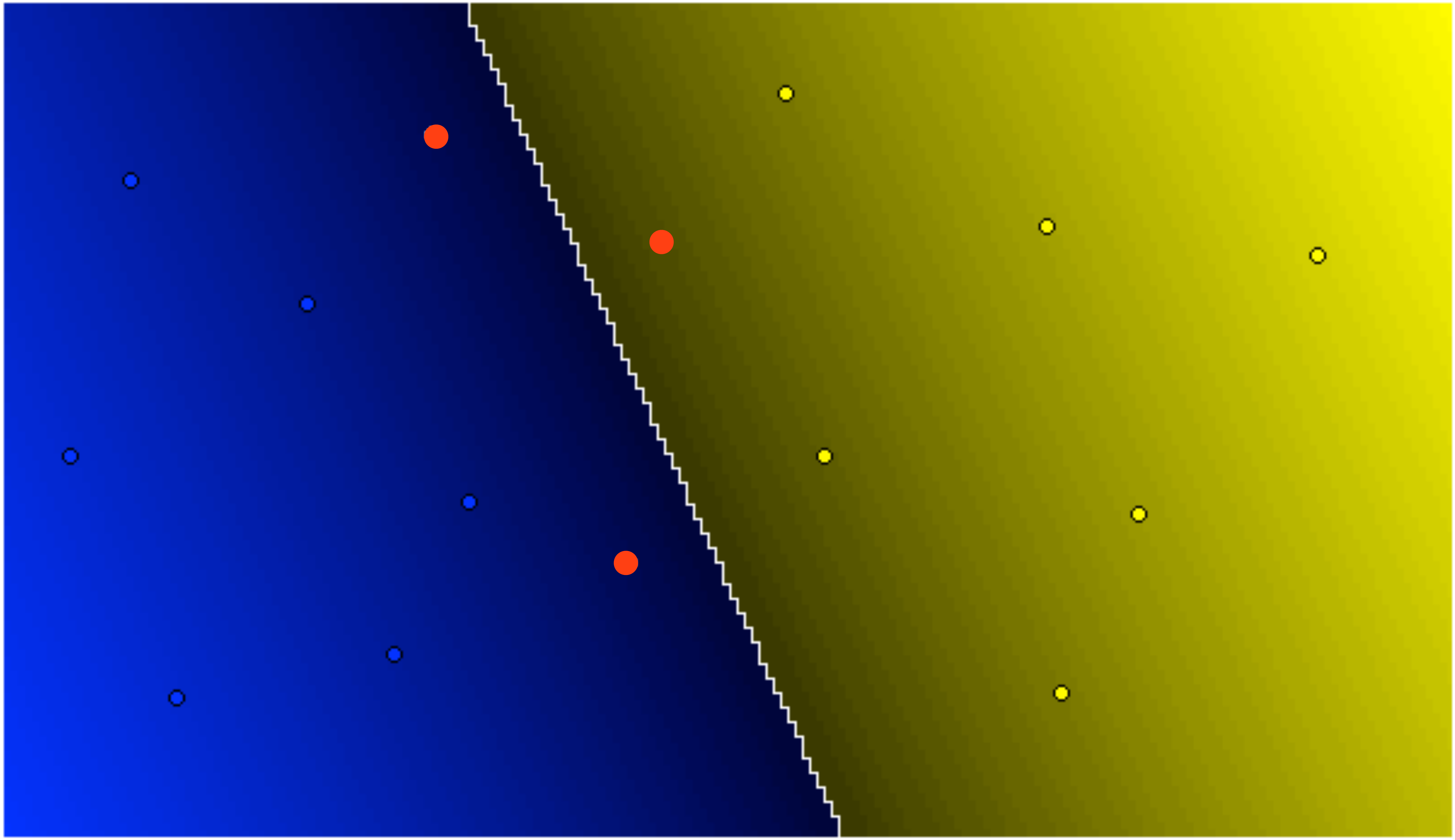
Java demo: http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml

Example

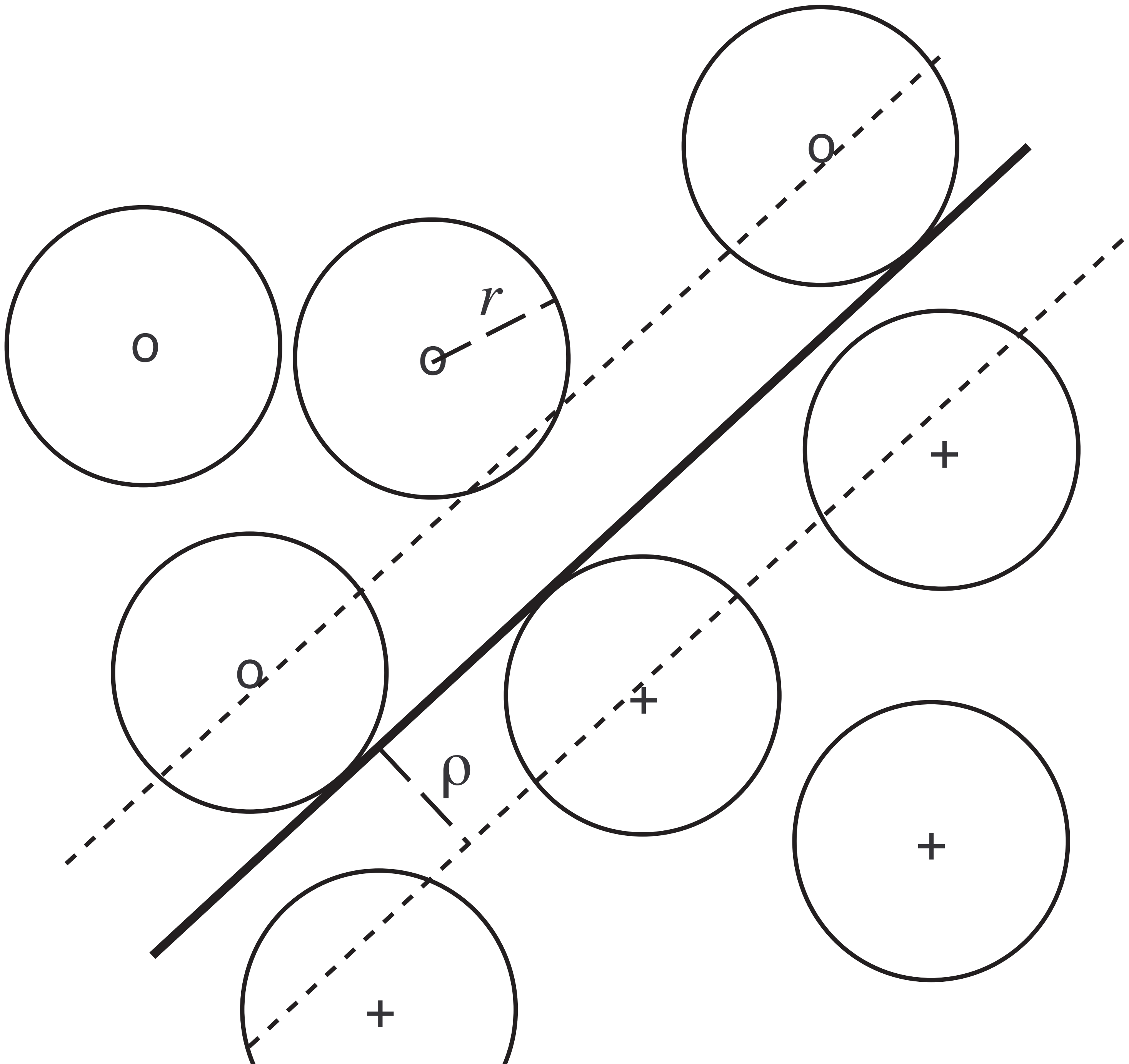


Example

Number of Support Vectors: **3** (-ve: 2, +ve: 1) Total number of points: 15



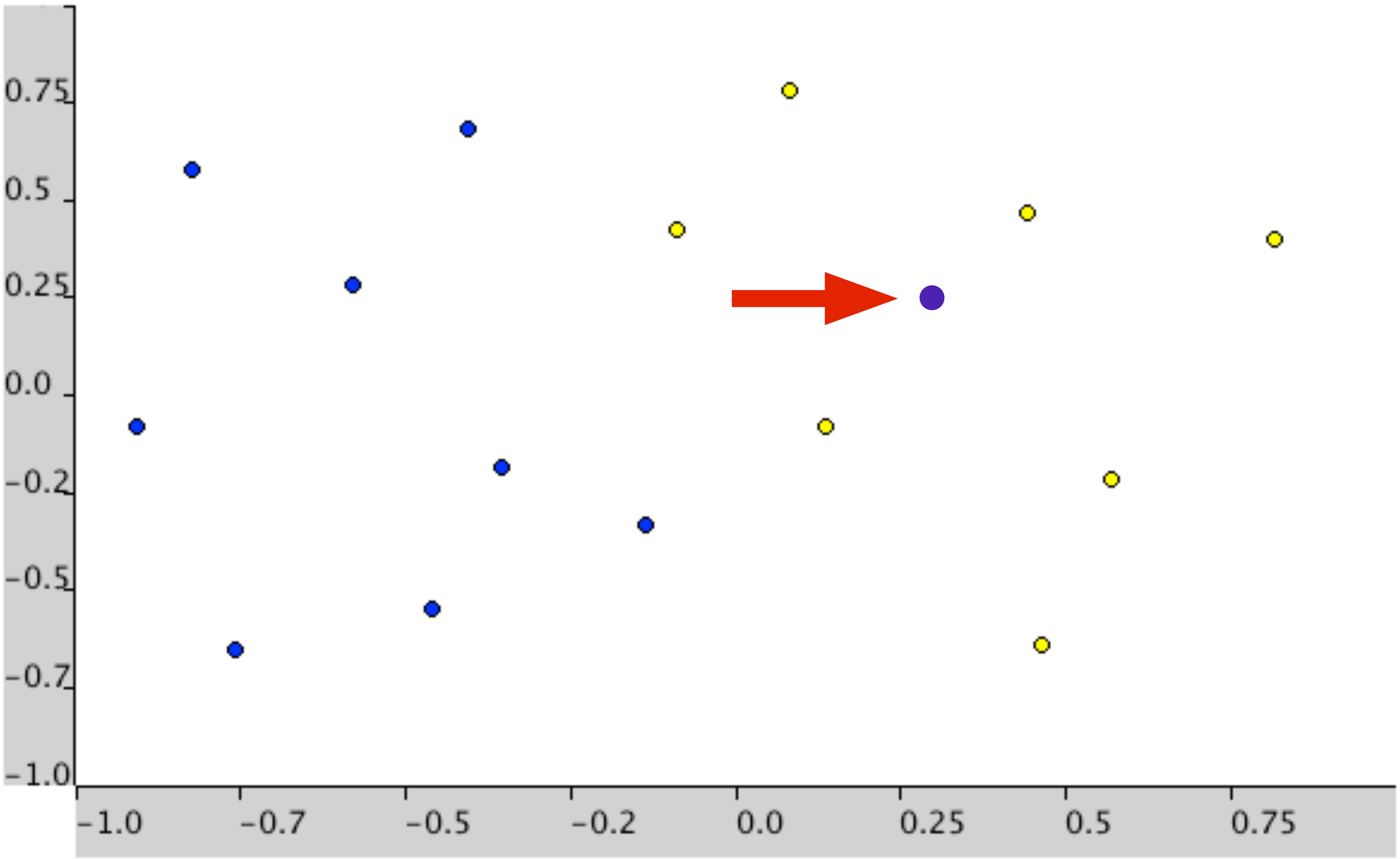
Why large margins?



- Maximum robustness relative to uncertainty
- Symmetry breaking
- Independent of correctly classified instances
- Easy to find for easy problems

Inseparable data

Quadratic program has no feasible solution



Adding slack variables

- Hard margin problem

$$\underset{w,b}{\text{minimize}} \frac{1}{2}\|w\|^2 \text{ subject to } y_i\left[\langle w, x_i\rangle + b\right] \geq 1$$

- With slack variables

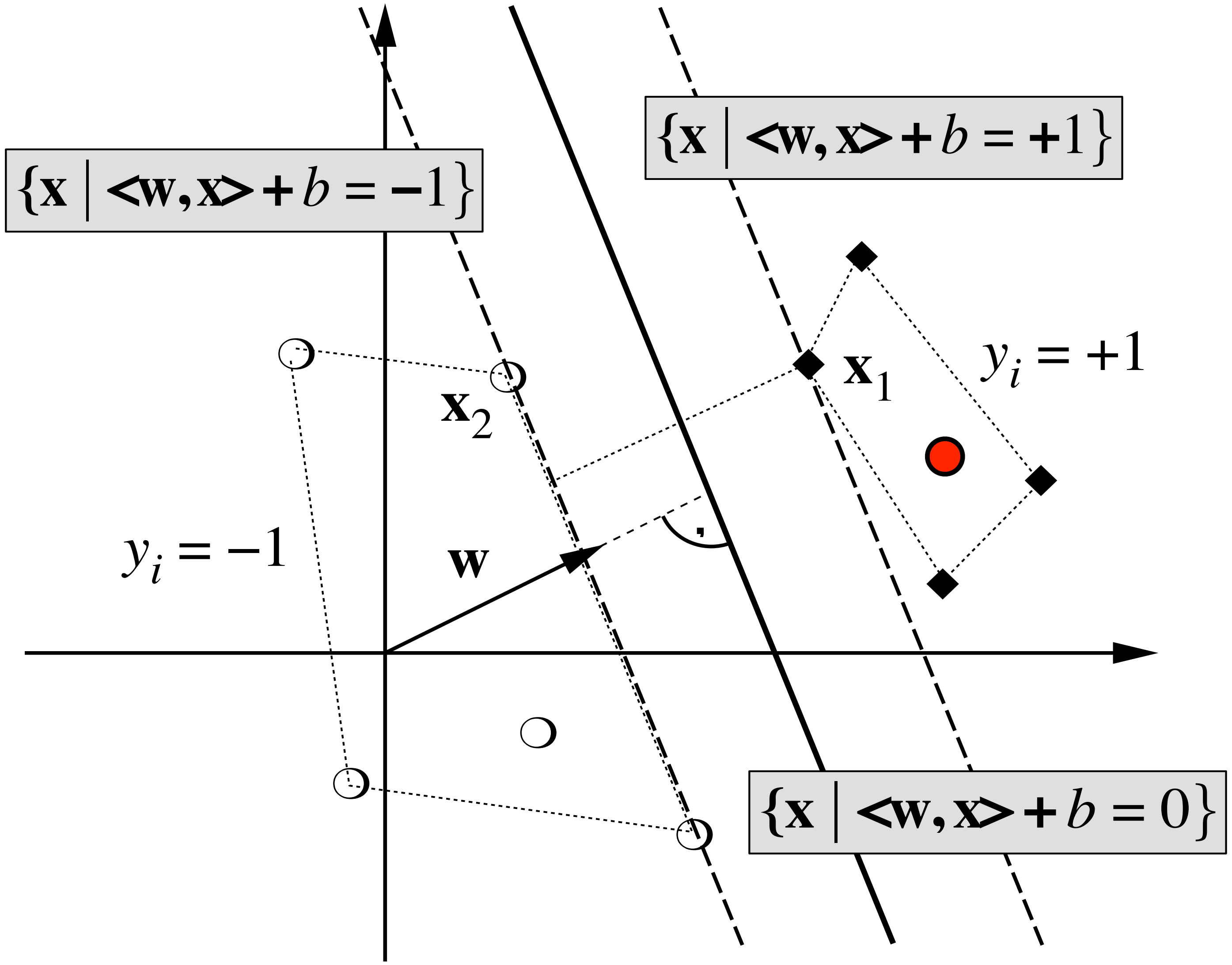
$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i\left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

problem is always feasible. Proof:

$w = 0$ and $b = 0$ and $\xi_i = 1$ (also yields upper bound)

Support Vectors



dual problem

$$\underset{\alpha}{\text{minimize}} \;\; \frac{1}{2}\alpha^\top K\alpha - 1^\top \alpha$$

$$\text{subject to } \sum_i \alpha_i y_i = 0$$

$$\alpha_i \in [0, C]$$

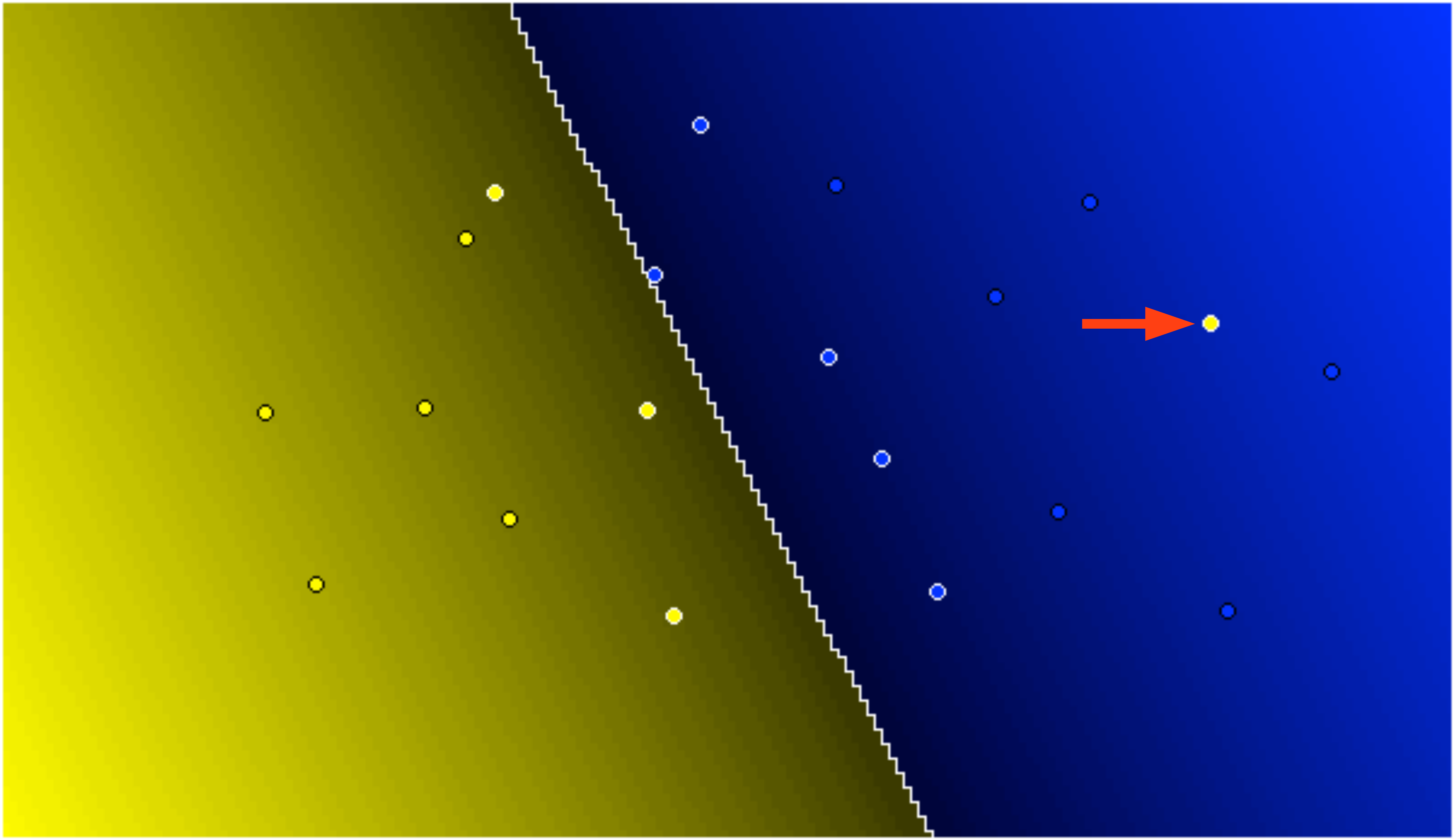
$$K_{ij} = y_i y_j \left\langle x_i, x_j \right\rangle$$

$$w = \sum_i \alpha_i y_i x_i$$

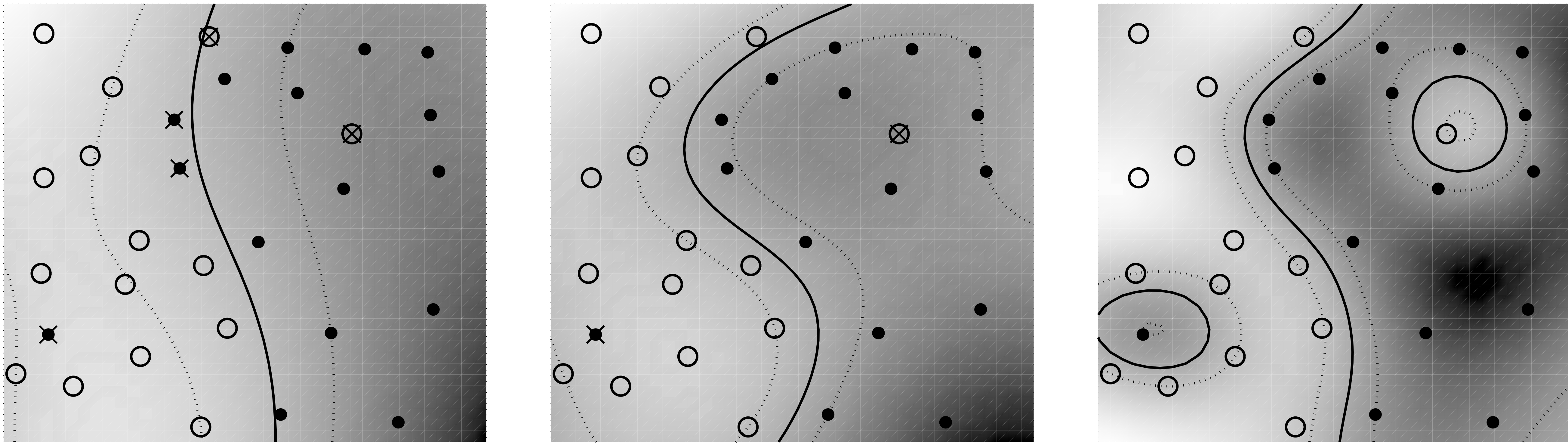
$$\underset{w,b}{\text{minimize}} \;\; \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i \left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

Classification with errors



Nonlinear separation



- Increasing C allows for more nonlinearities
- Decreases number of errors
- SV boundary need not be contiguous

Loss function point of view

- Constrained quadratic program

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to } y_i\left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

- Risk minimization setting

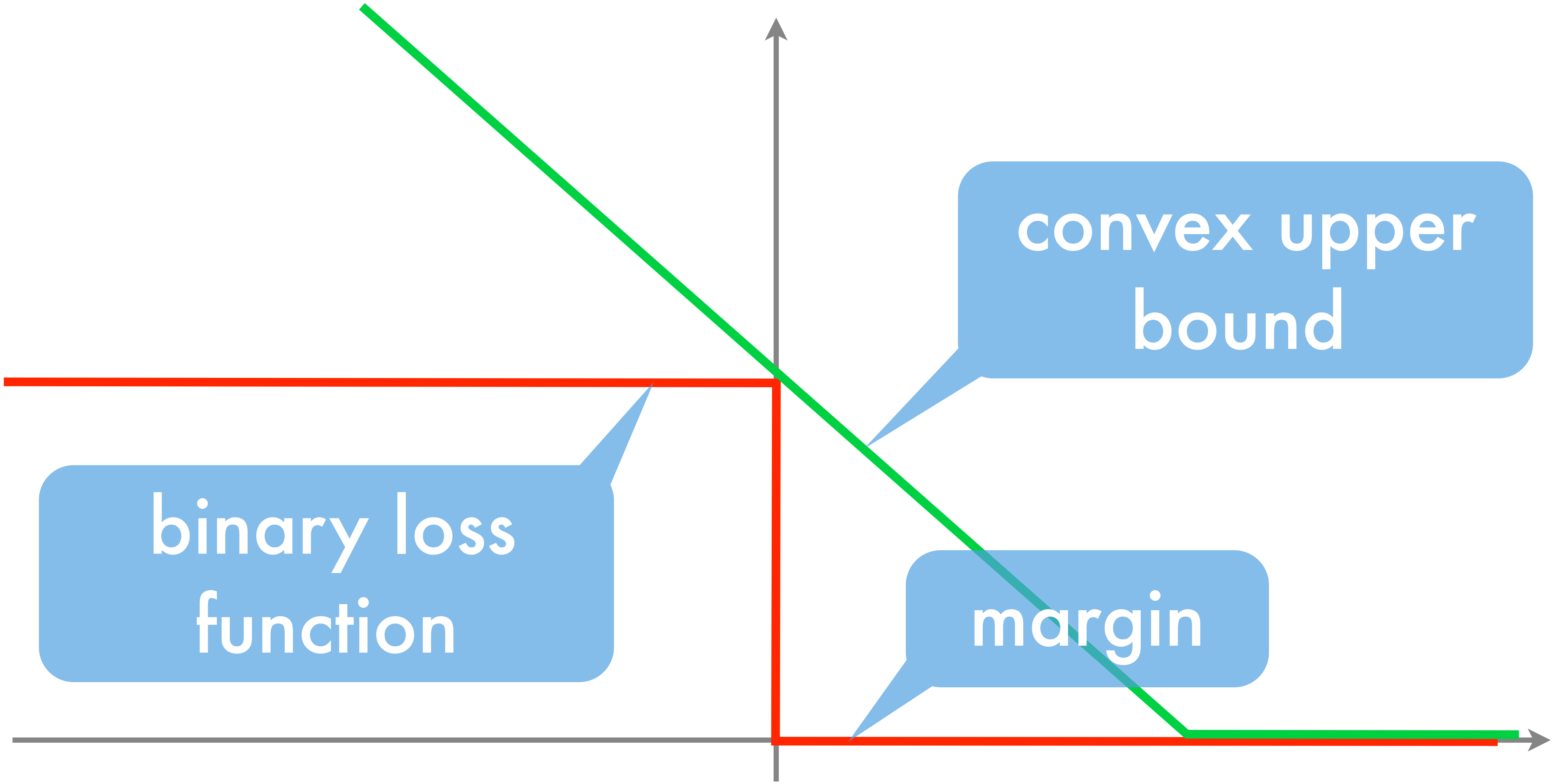
$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_i \max\left[0, 1 - y_i\left[\langle w, x_i\rangle + b\right]\right]$$

empirical risk

Follows from finding minimal slack variable for given (w,b) pair.

Soft margin as proxy for binary

- **Soft margin loss** $\max(0, 1 - yf(x))$
- **Binary loss** $\{yf(x) < 0\}$



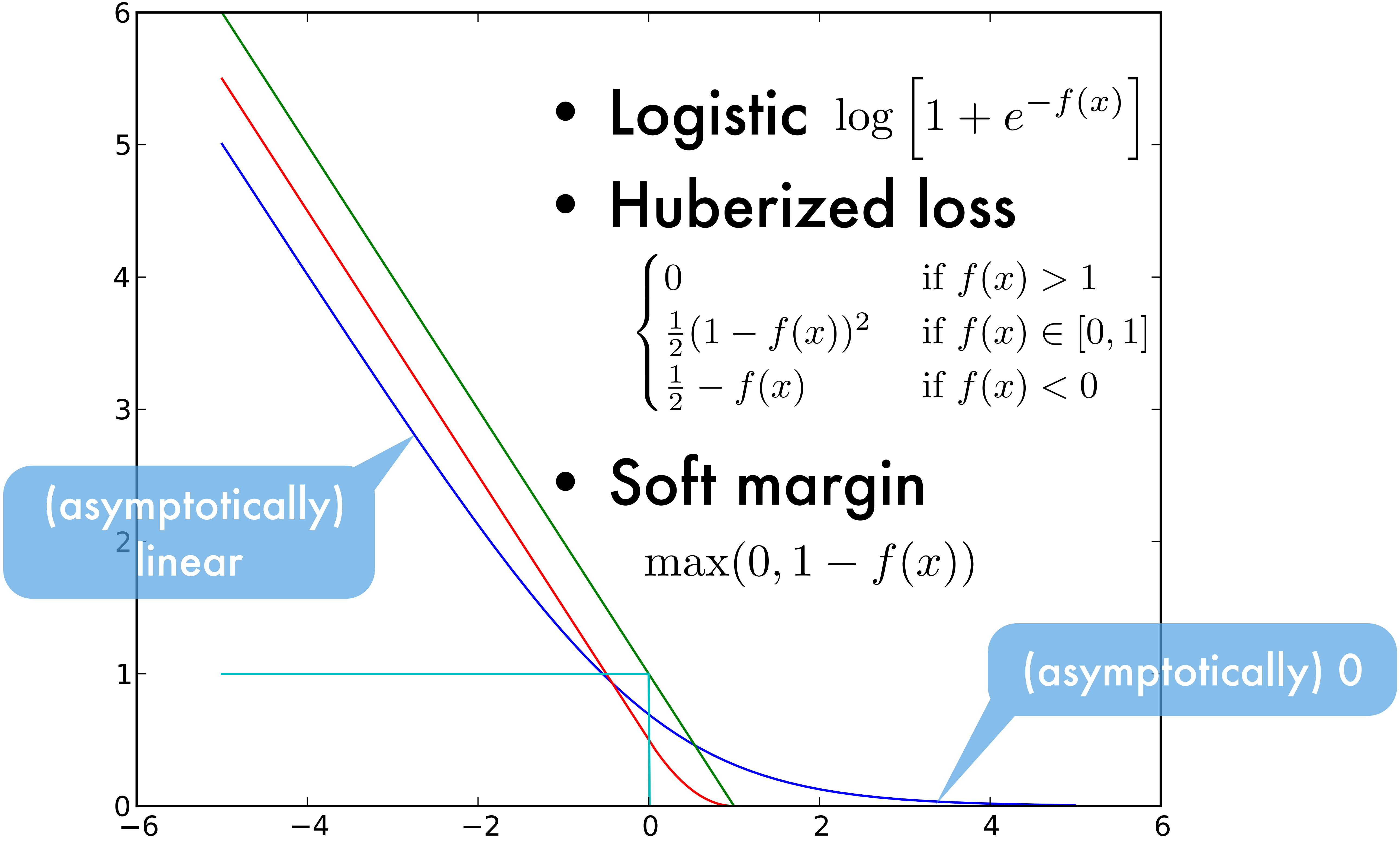
More loss functions

- **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**

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



Risk minimization view

- Find function f minimizing classification error

$$R[f] := \mathbf{E}_{x,y \sim p(x,y)} \left[\{ y f(x) > 0 \} \right]$$

- Compute empirical average

$$R_{\mathrm{emp}}[f] := \frac{1}{m} \sum_{i=1}^{m} \{ y_i f(x_i) > 0 \}$$

  - Minimization is nonconvex
  - Overfitting as we minimize empirical error
- Compute convex upper bound on the loss
- Add regularization for capacity control

regularization

$$R_{\mathrm{reg}}[f] := \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i f(x_i)) + \lambda \Omega[f]$$

how to control λ

Regression

"Under hypnosis you revealed that in your last eight lives you were ... er ... a cat."

Regression Estimation

- Find function f minimizing regression error

$$R[f] := \mathbf{E}_{x,y \sim p(x,y)} \left[l(y, f(x)) \right]$$

- Compute empirical average

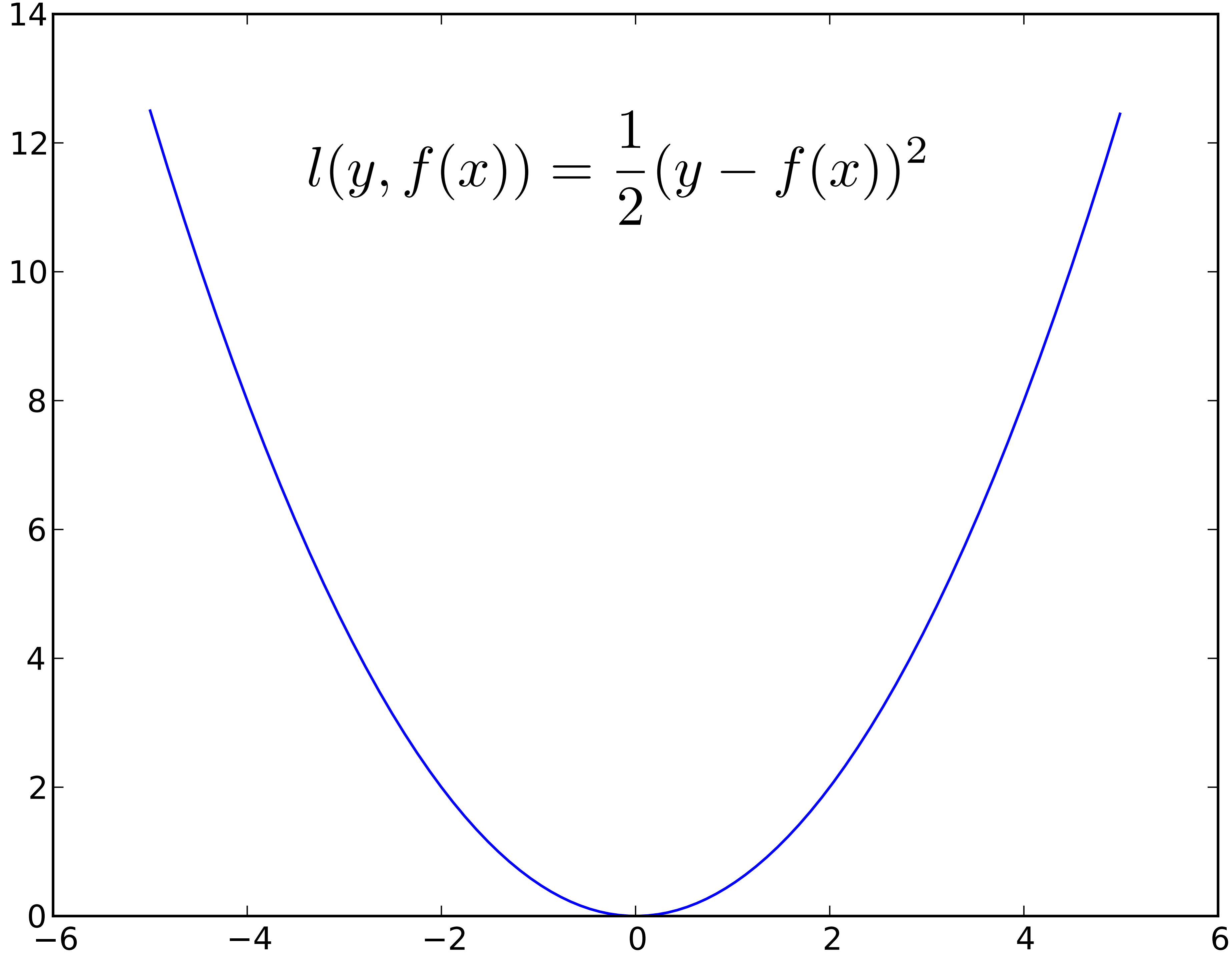
$$R_{\mathrm{emp}}[f] := \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(x_i))$$

Overfitting as we minimize empirical error

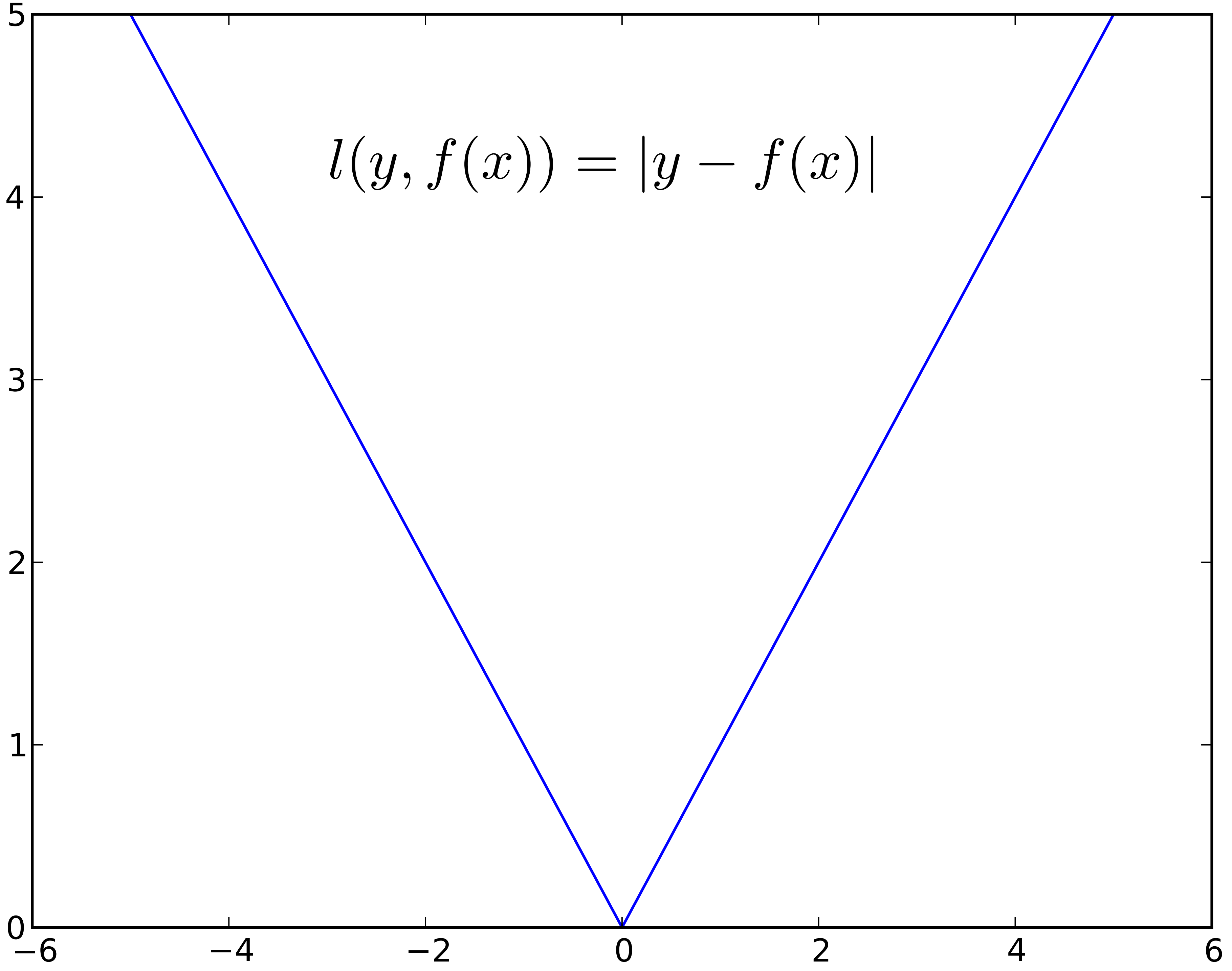
- Add regularization for capacity control

$$R_{\mathrm{reg}}[f] := \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(x_i)) + \lambda \Omega[f]$$

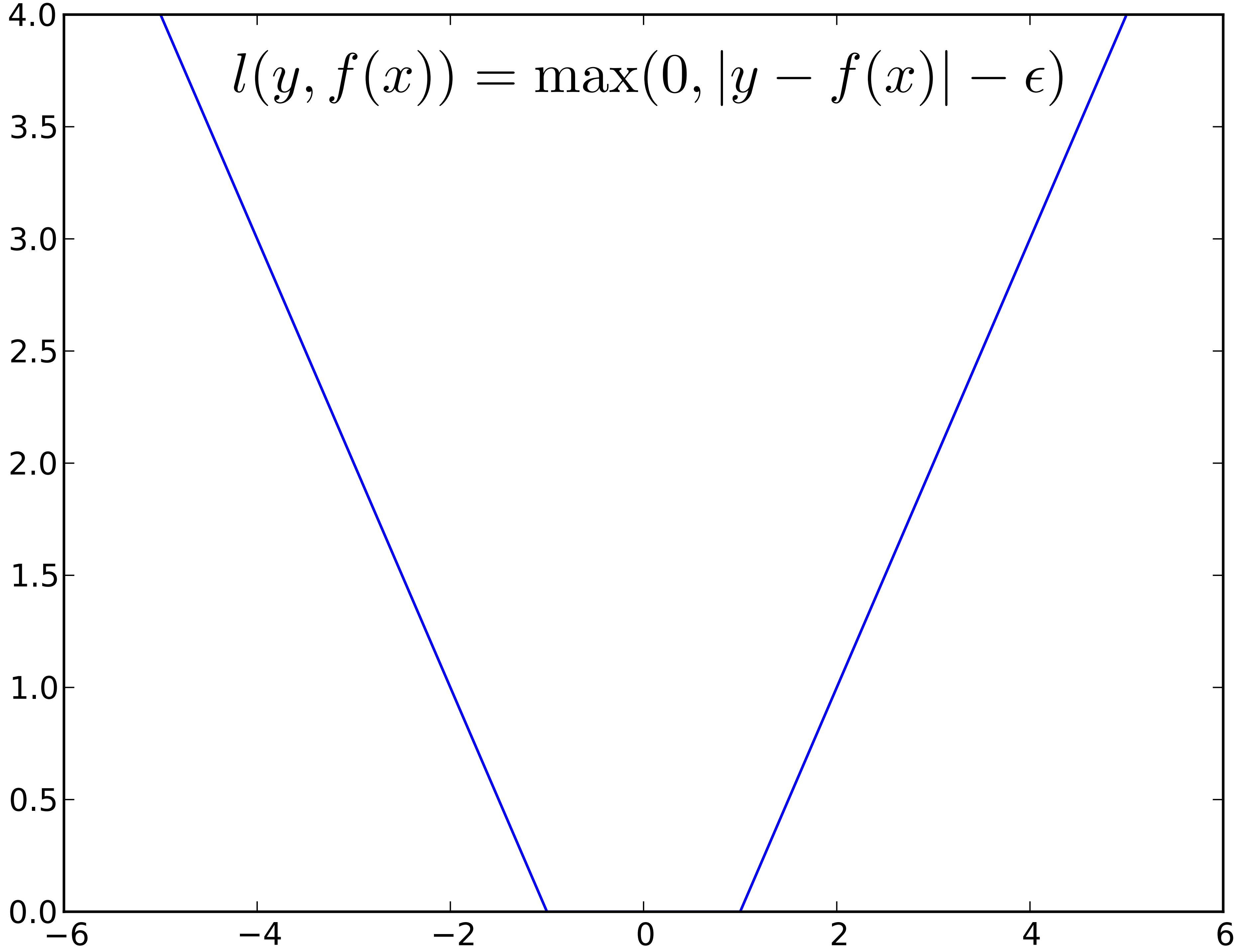
Squared loss



l1 loss



ε-insensitive Loss



Penalized least mean squares

- Optimization problem

$$\underset{w}{\text{minimize}} \frac{1}{m} \sum_{i=1}^{m} (y_i - \langle x_i, w \rangle)^2 + \frac{\lambda}{2} \|w\|^2$$

- Solution

$$\partial_w [\ldots] = \frac{1}{m} \sum_{i=1}^{m} \left[x_i x_i^\top w - x_i y_i \right] + \lambda w$$

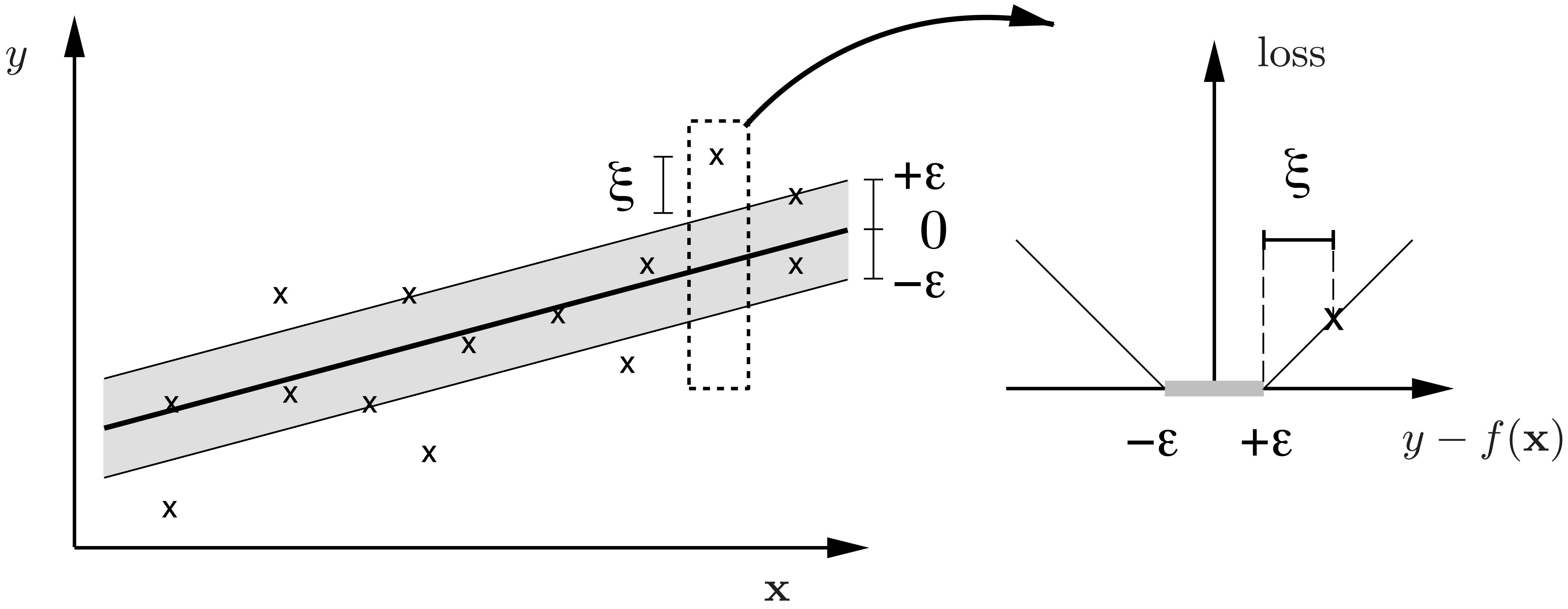
$$= \left[\frac{1}{m} X X^\top + \lambda \mathbf{1} \right] w - \frac{1}{m} X y = 0$$

$$\text{hence } w = \left[X X^\top + \lambda m \mathbf{1} \right]^{-1} X y$$

only inner product between X matters

matrix inverse use CG or SMW

SVM Regression
(ϵ-insensitive loss)



don't care about deviations within the tube

SVM Regression (ϵ-insensitive loss)

- Optimization Problem (as constrained QP)

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}[\xi_i + \xi_i^*]$$

$$\text{subject to} \ \langle w, x_i\rangle + b \le y_i + \epsilon + \xi_i \ \text{ and } \xi_i \ge 0$$

$$\langle w, x_i\rangle + b \ge y_i - \epsilon - \xi_i^* \ \text{ and } \xi_i^* \ge 0$$

- Lagrange Function

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}[\xi_i + \xi_i^*] - \sum_{i=1}^{m}[\eta_i\xi_i + \eta_i^*\xi_i^*] +$$

$$\sum_{i=1}^{m}\alpha_i\left[\langle w, x_i\rangle + b - y_i - \epsilon - \xi_i\right] + \sum_{i=1}^{m}\alpha_i^*\left[y_i - \epsilon - \xi_i^* - \langle w, x_i\rangle - b\right]$$

SVM Regression (ϵ-insensitive loss)

- First order conditions

$$\partial_w L = 0 = w + \sum_i [\alpha_i - \alpha_i^*] x_i$$

$$\partial_b L = 0 = \sum_i [\alpha_i - \alpha_i^*]$$

$$\partial_{\xi_i} L = 0 = C - \eta_i - \alpha_i$$

$$\partial_{\xi_i^*} L = 0 = C - \eta_i^* - \alpha_i^*$$

- Dual problem

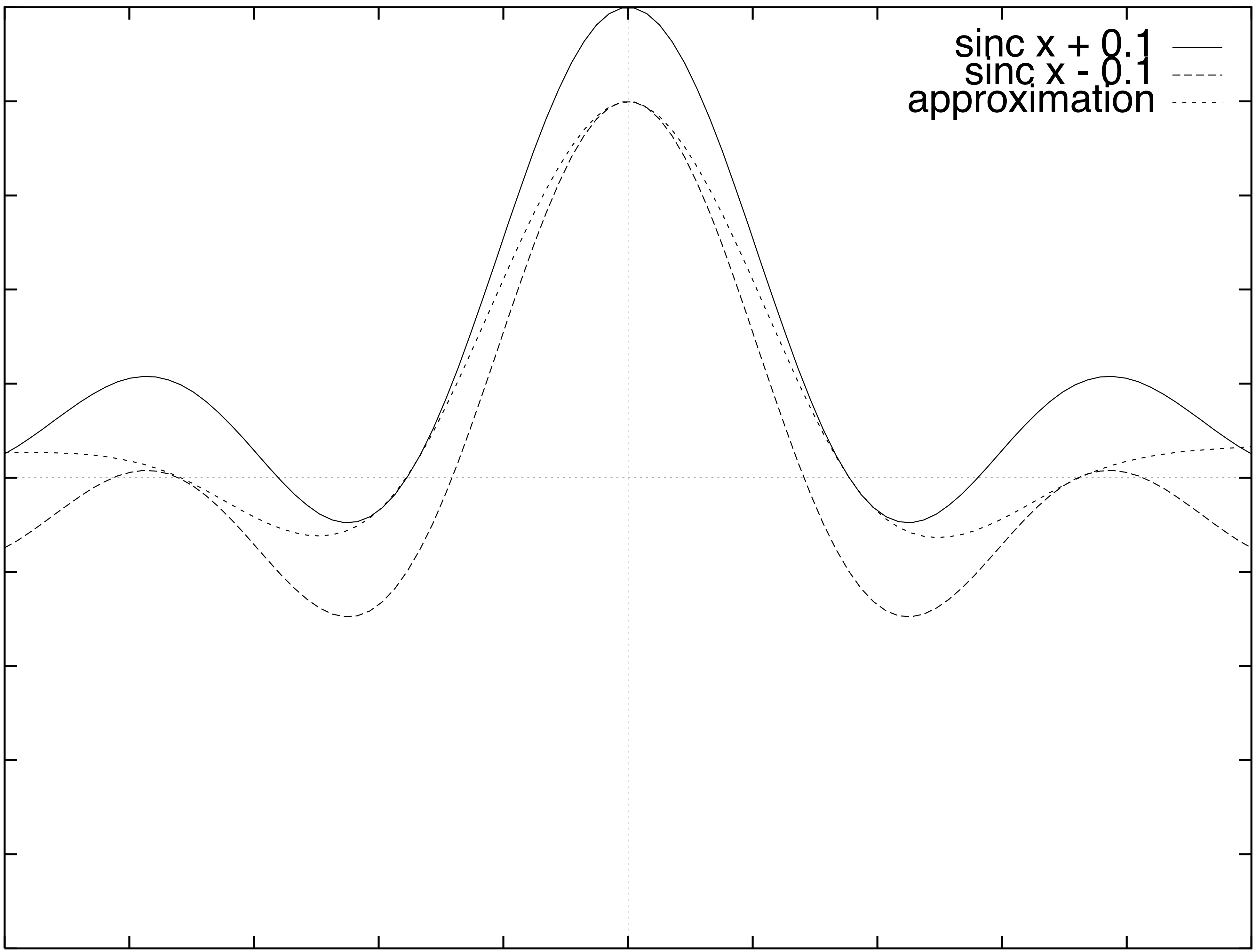
$$\underset{\alpha,\alpha^*}{\text{minimize}} \;\; \frac{1}{2}(\alpha - \alpha^*)^\top K(\alpha - \alpha^*) + \epsilon 1^\top(\alpha + \alpha^*) + y^\top(\alpha - \alpha^*)$$

$$\text{subject to } 1^\top(\alpha - \alpha^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

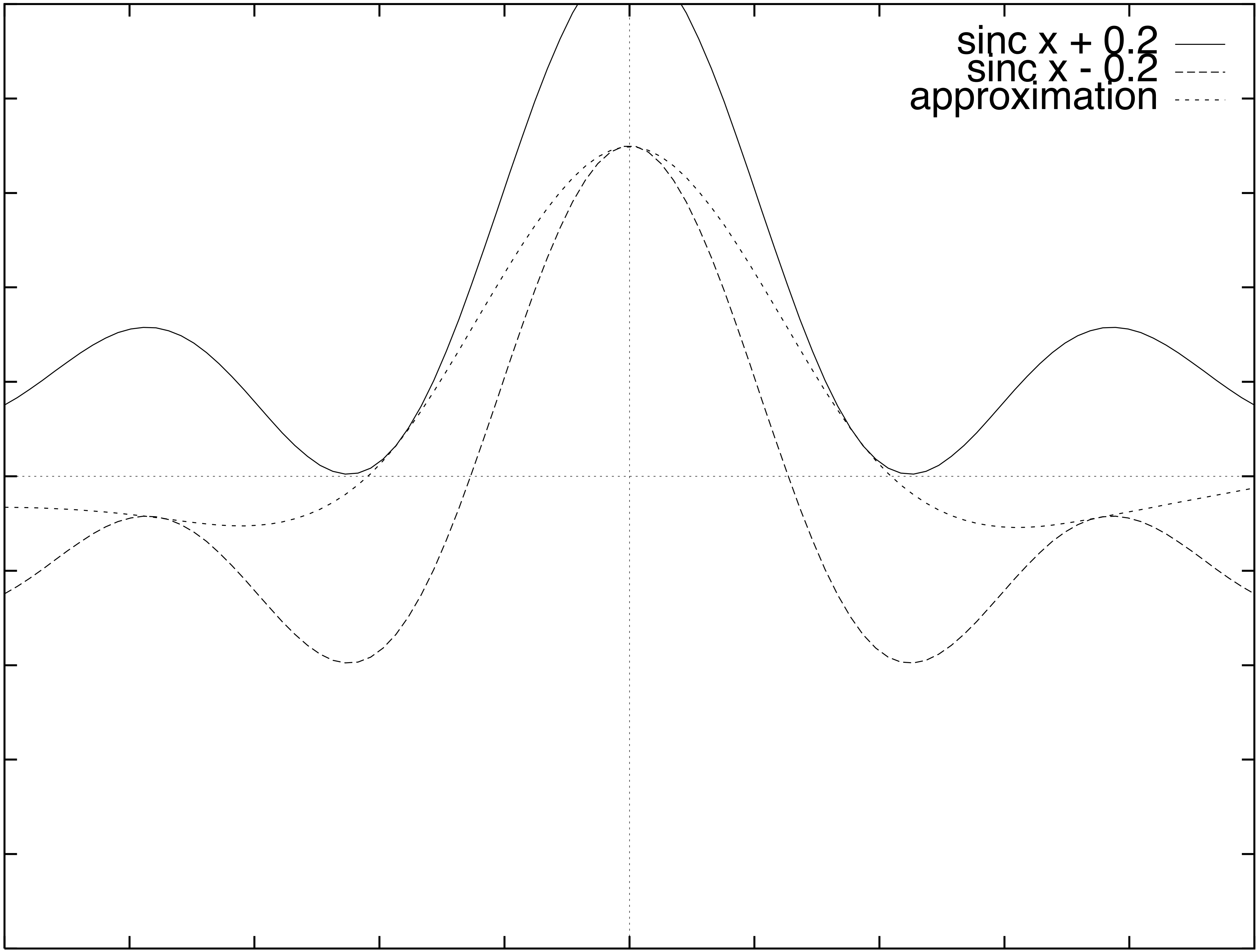
Properties

- Ignores ‘typical’ instances with small error
- Only upper or lower bound active at any time (we cannot violate both bounds simultaneously)
- Quadratic Program in 2n variables can be solved as cheaply as standard SVM problem
- Robustness with respect to outliers
  - l1 loss yields same problem without epsilon
  - Huber’s robust loss yields similar problem but with added quadratic penalty on coefficients

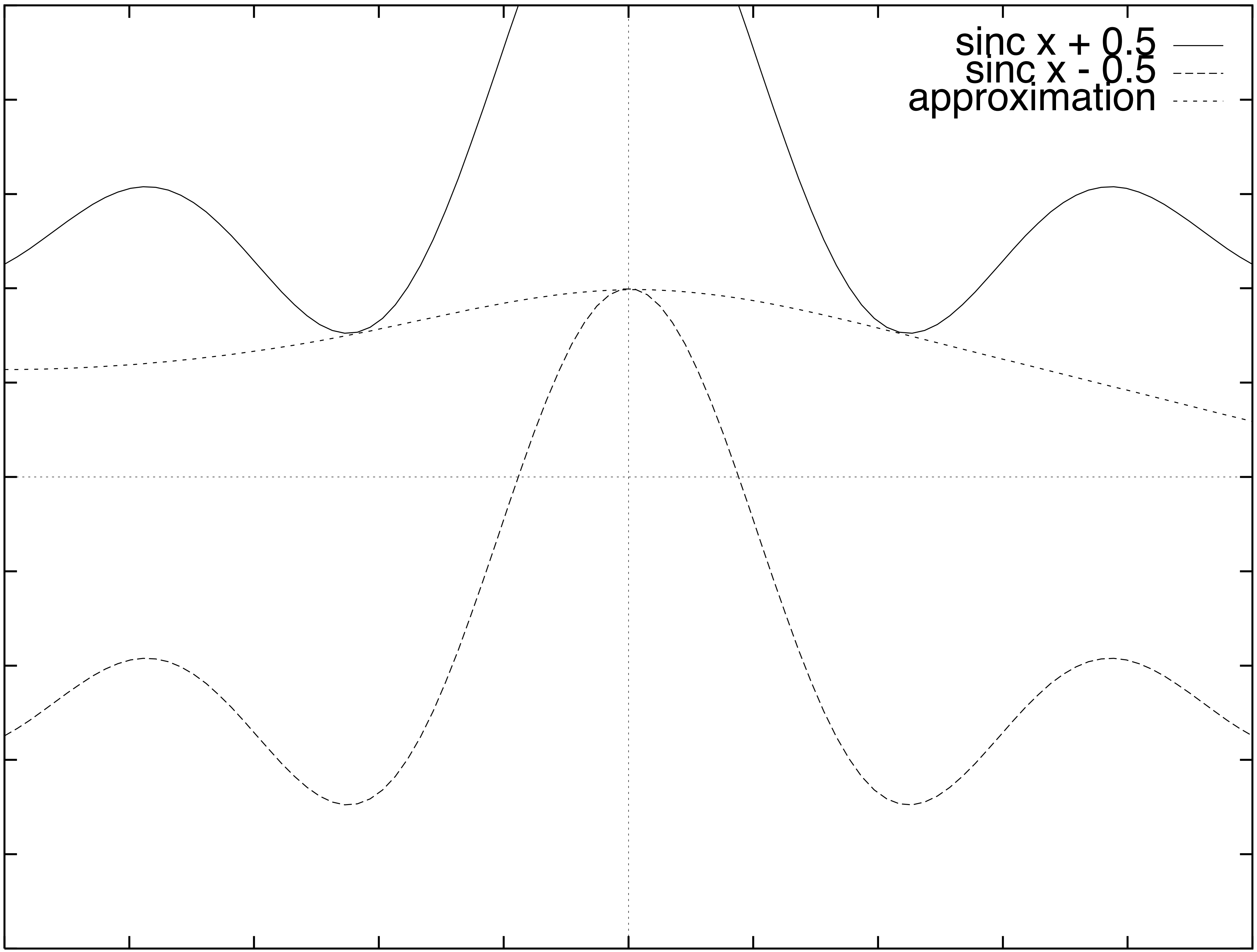
Regression example



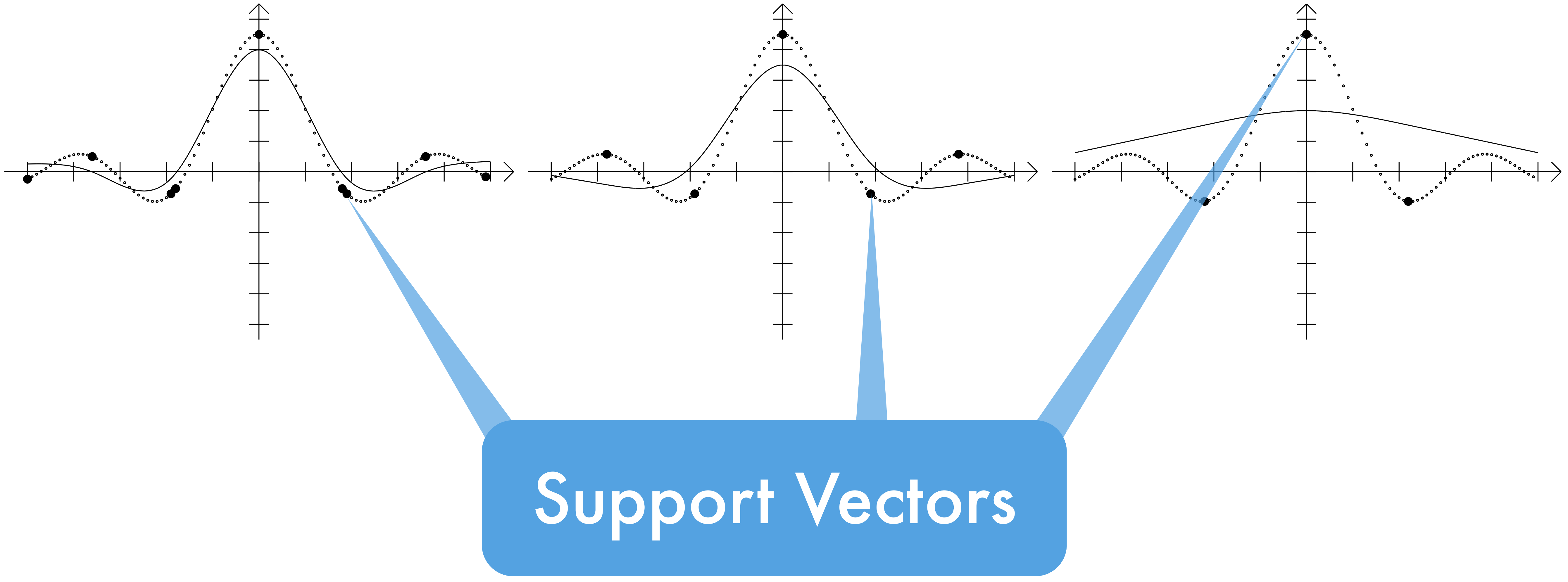
Regression example



Regression example

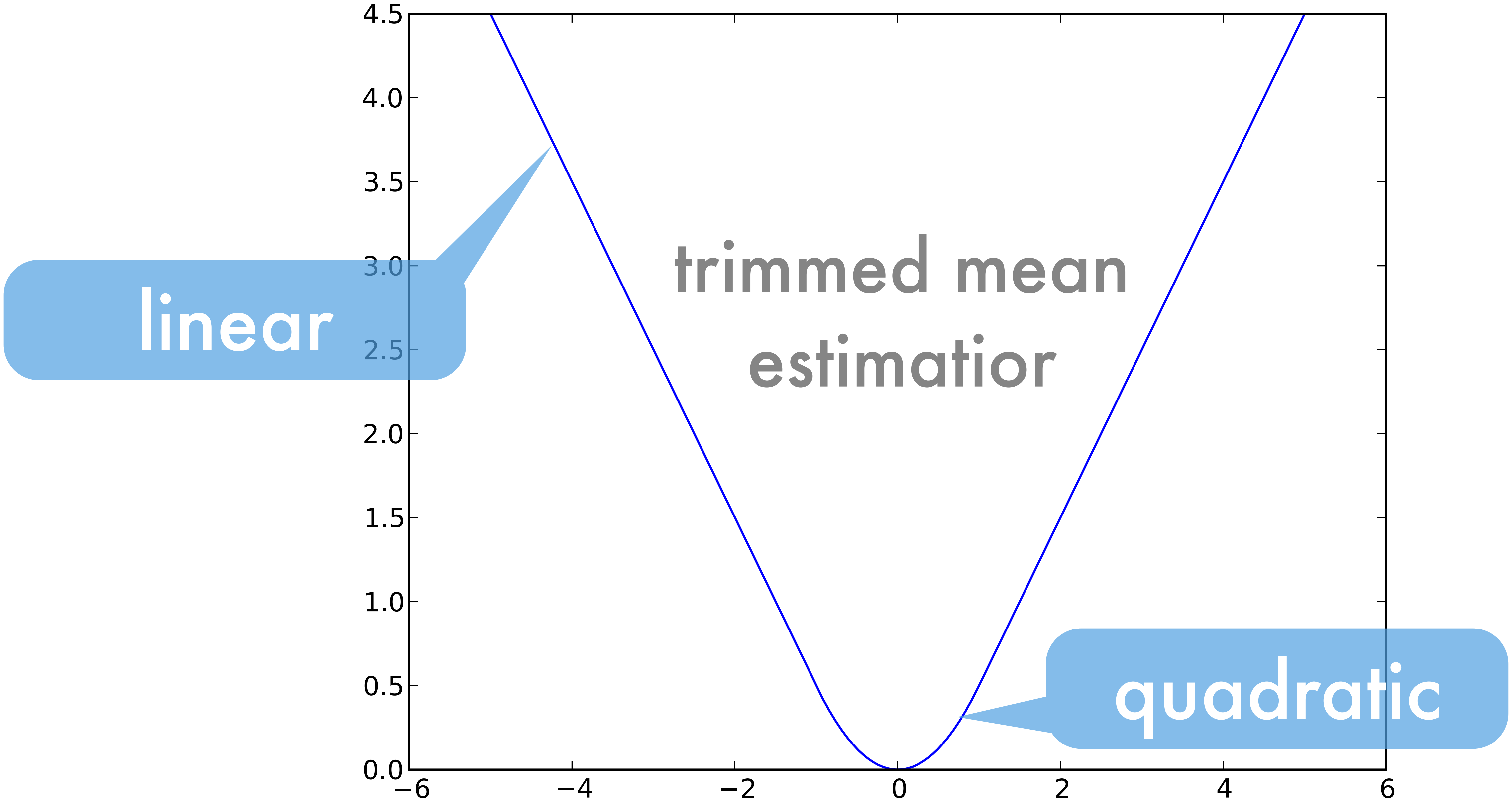


Regression example

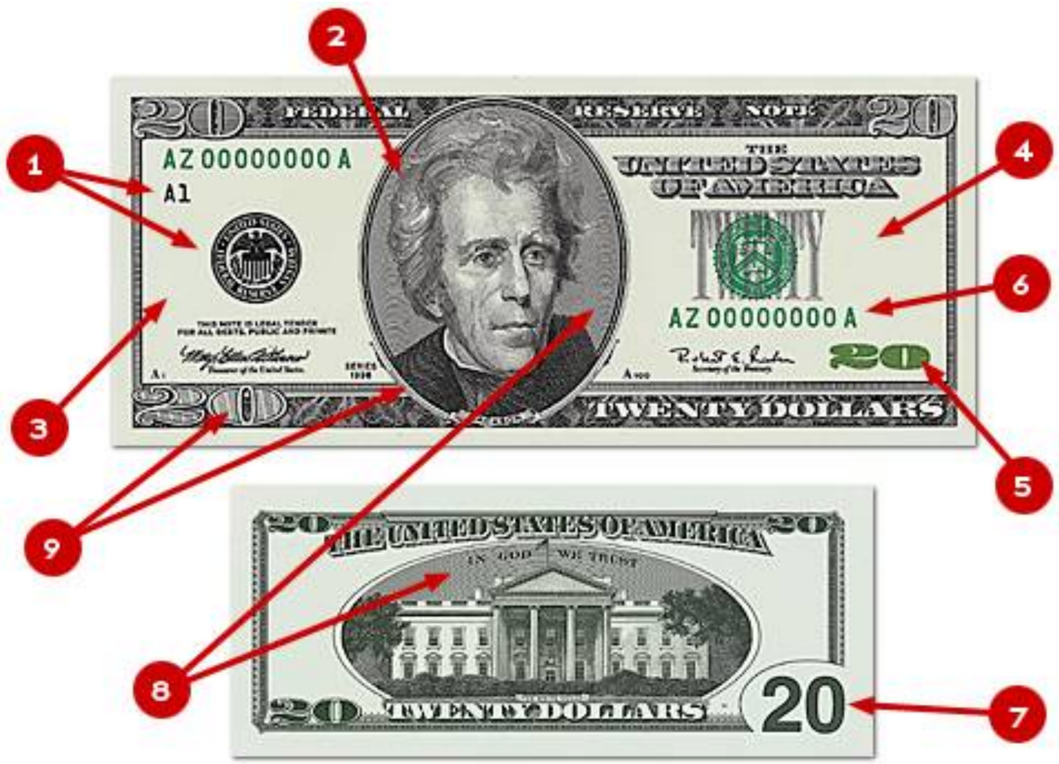


Huber's robust loss

$$l(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{if } |y - f(x)| < 1 \\ |y - f(x)| - \frac{1}{2} & \text{otherwise} \end{cases}$$



Novelty Detection

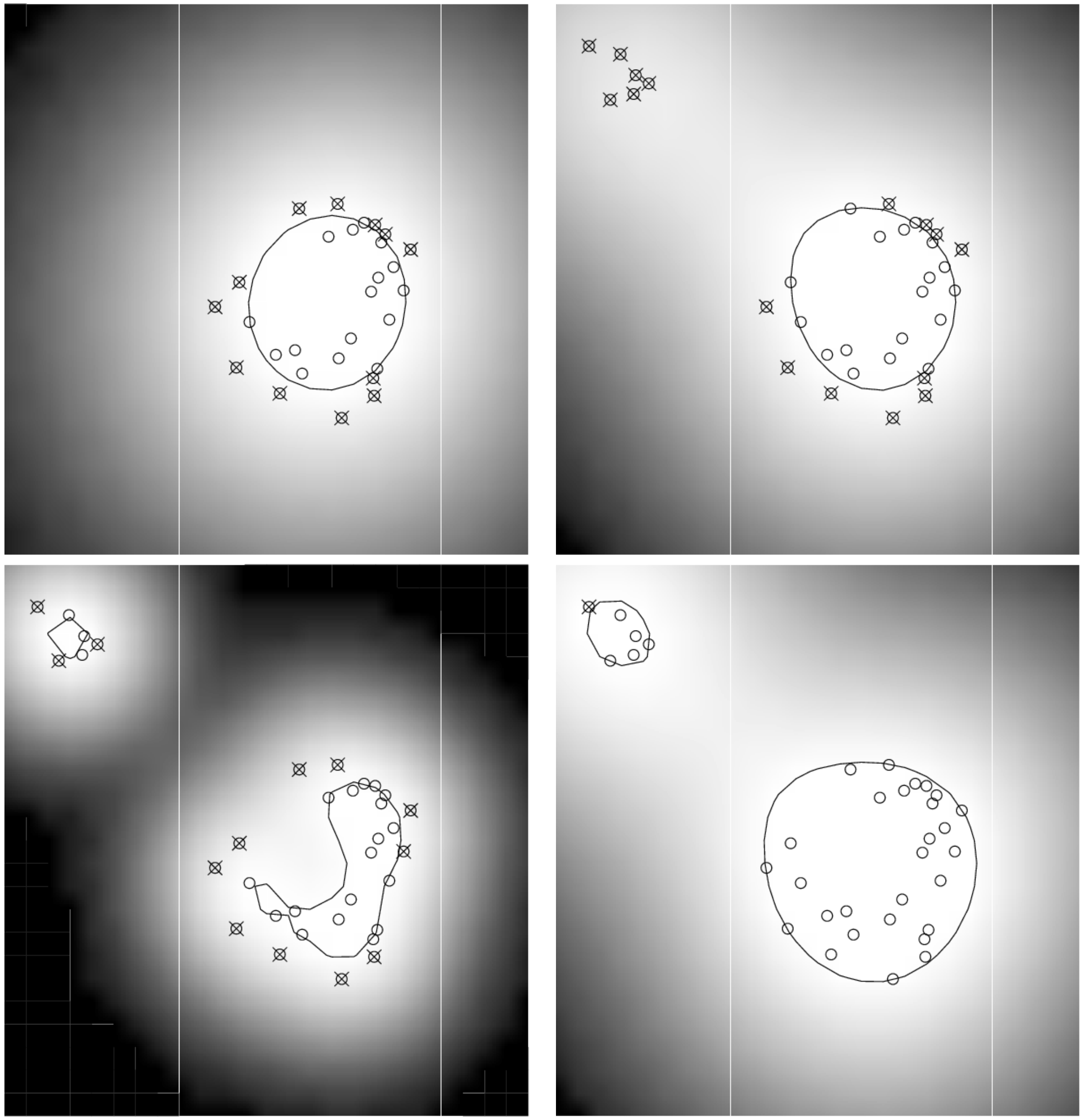


Basic Idea

Data

Observations (x_i) generated from some $\mathrm{P}(x)$, e.g.,

- network usage patterns
- handwritten digits
- alarm sensors
- factory status



Task

Find unusual events, clean database, distinguish typical examples.

Applications

Network Intrusion Detection

Detect whether someone is trying to hack the network, downloading tons of MP3s, or doing anything else *unusual* on the network.

Jet Engine Failure Detection

You can't destroy jet engines just to see *how* they fail.

Database Cleaning

We want to find out whether someone stored bogus information in a database (typos, etc.), mislabelled digits, ugly digits, bad photographs in an electronic album.

Fraud Detection

Credit Cards, Telephone Bills, Medical Records

Self calibrating alarm devices

Car alarms (adjusts itself to where the car is parked), home alarm (furniture, temperature, windows, etc.)

Novelty Detection via Density Estimation

Key Idea

- Novel data is one that we don't see frequently.
- It must lie in low density regions.

Step 1: Estimate density

- Observations $x_1, \ldots, x_m$
- Density estimate via Parzen windows

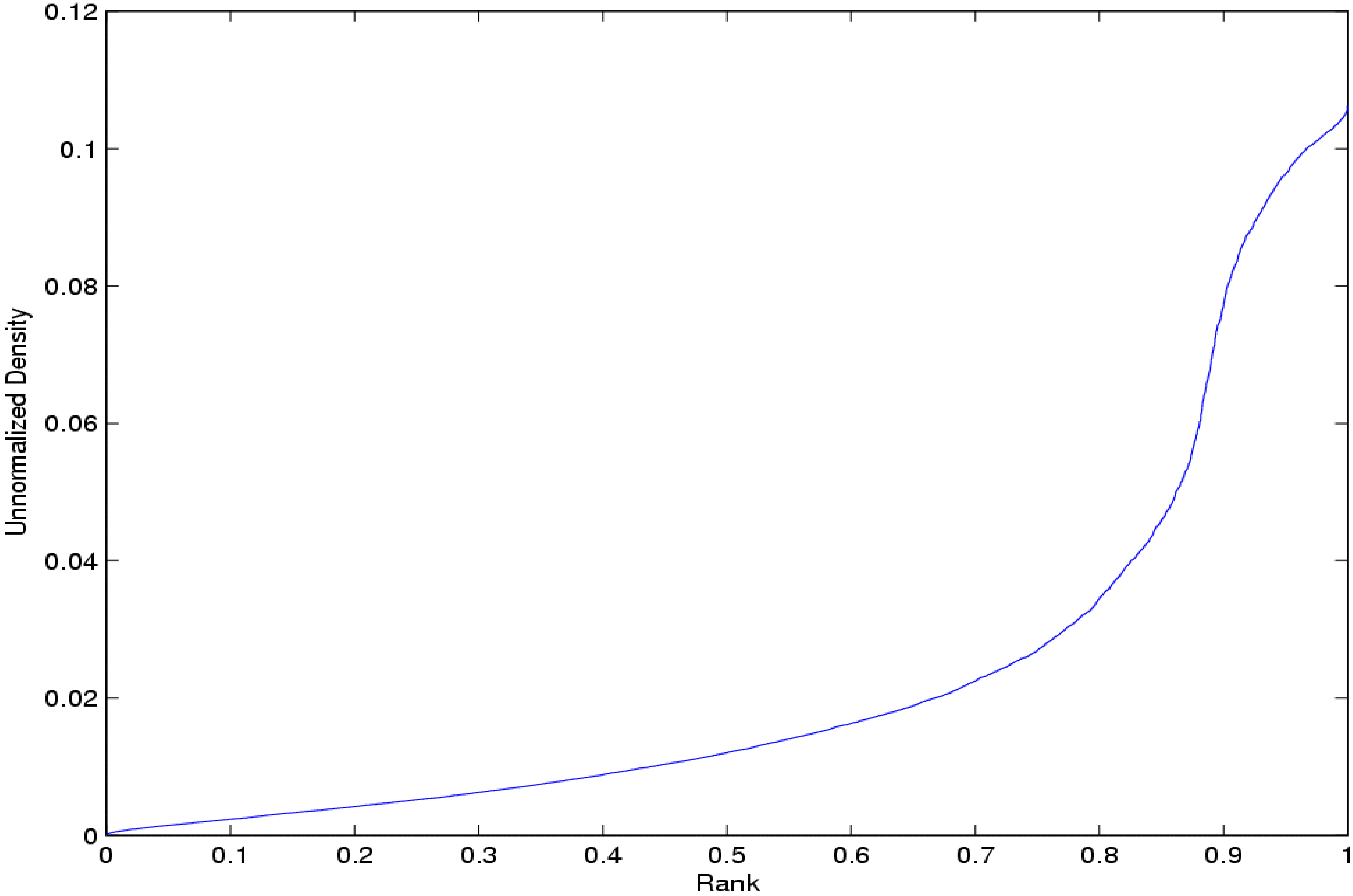
Step 2: Thresholding the density

- Sort data according to density and use it for rejection
- Practical implementation: compute

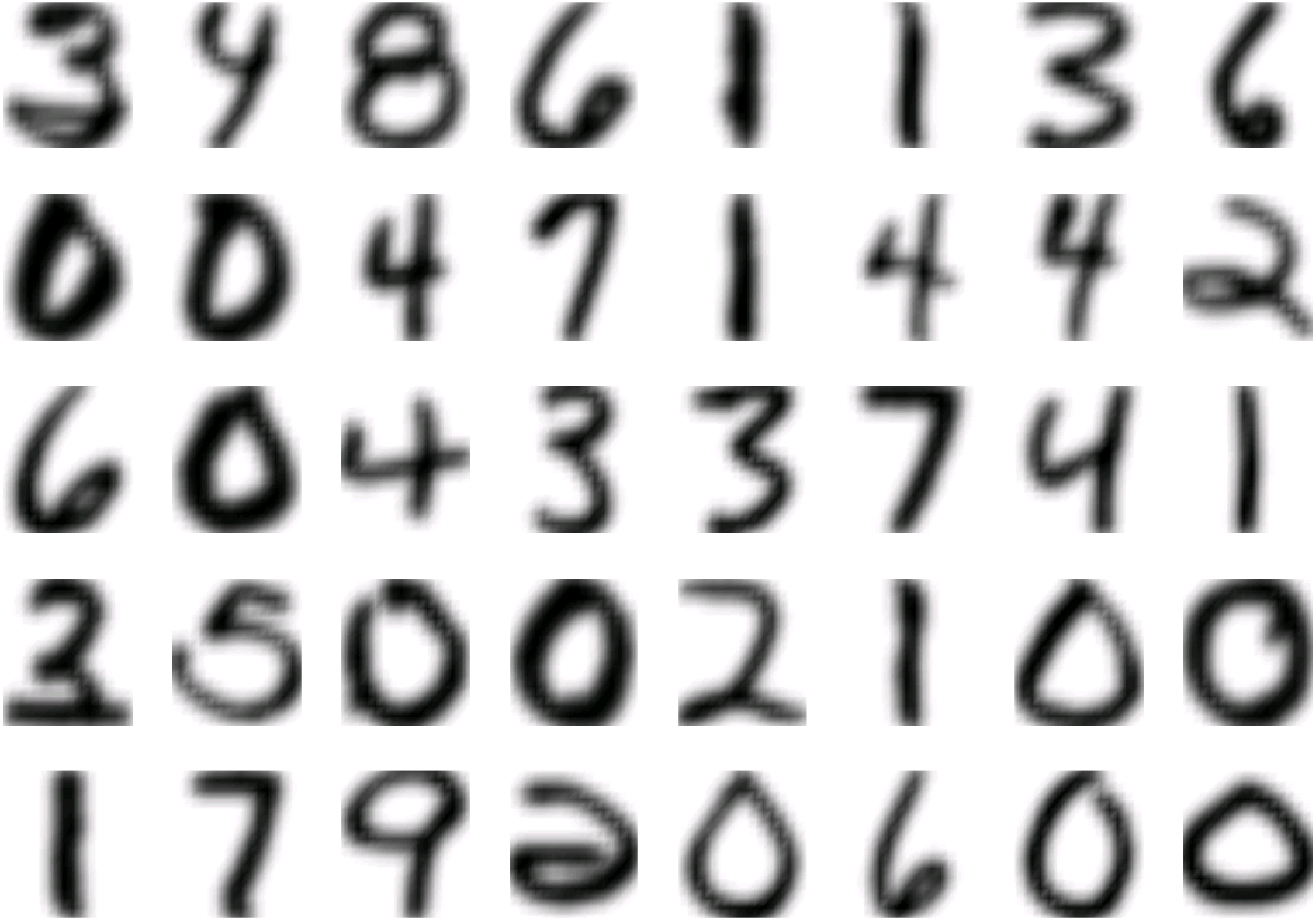
$$p(x_i) = \frac{1}{m} \sum_j k(x_i, x_j) \text{ for all } i$$

  and sort according to magnitude.
- Pick smallest $p(x_i)$ as novel points.

Order Statistics of Densities



Typical Data



Outliers

A better way

Problems

- We do not care about estimating the density properly in regions of high density (waste of capacity).
- We only care about the relative density for thresholding purposes.
- We want to eliminate a certain fraction of observations and tune our estimator specifically for this fraction.

Solution

- Areas of low density can be approximated as the **level set** of an auxiliary function. No need to estimate $p(x)$ directly — use proxy of $p(x)$.
- Specifically: find $f(x)$ such that x is novel if $f(x) \leq c$ where c is some constant, i.e. $f(x)$ describes the amount of novelty.

Problems with density estimation

Maximum a Posteriori

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{m} g(\theta) - \langle \phi(x_i), \theta \rangle + \frac{1}{2\sigma^2} \|\theta\|^2$$

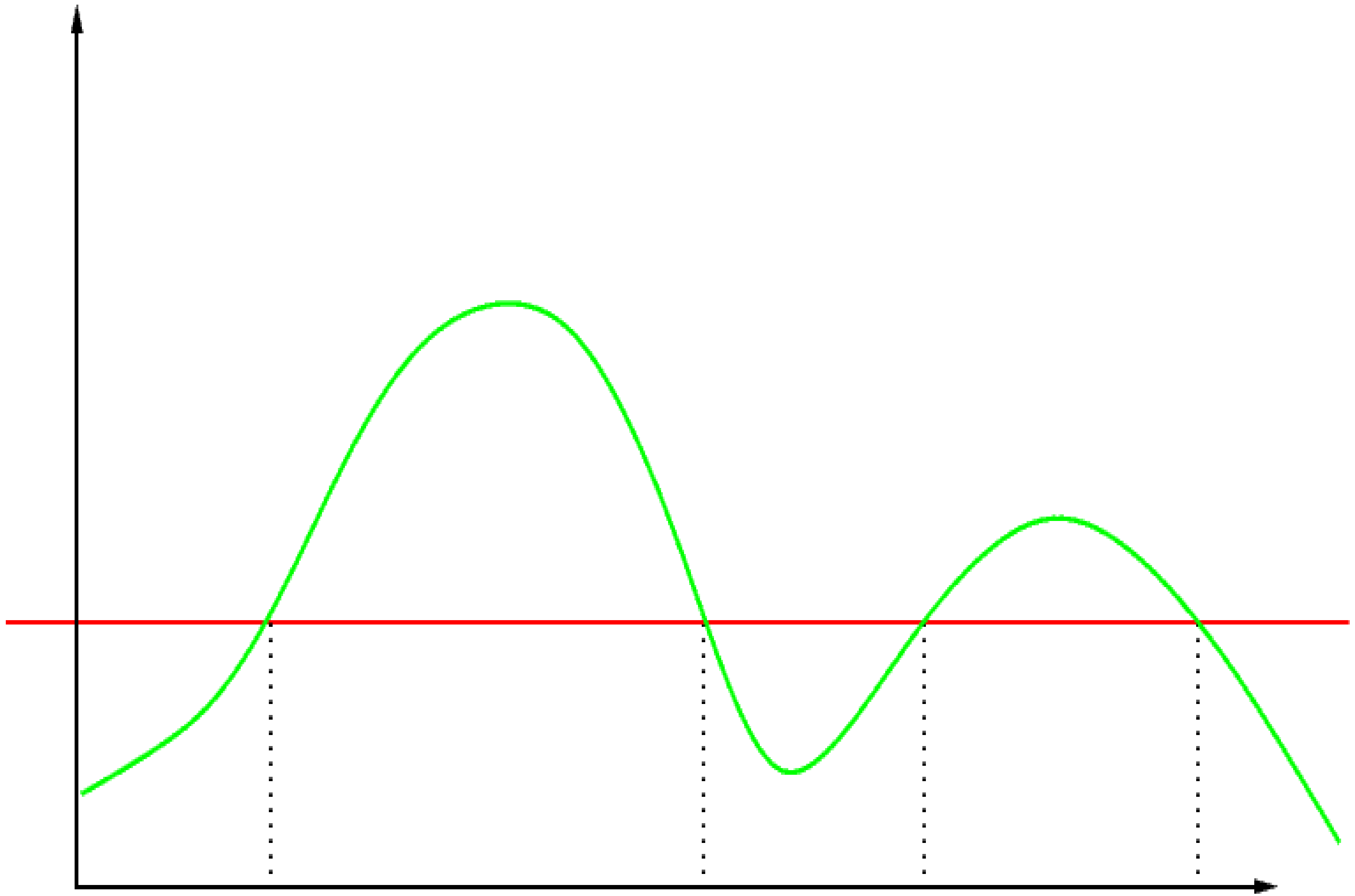
Advantages

- Convex optimization problem
- Concentration of measure

Problems

- Normalization $g(\theta)$ may be painful to compute
- For density estimation we need no normalized $p(x|\theta)$
- No need to perform particularly well in high density regions

Thresholding



Optimization Problem

Optimization Problem

MAP $$\sum_{i=1}^{m} -\log p(x_i|\theta) + \frac{1}{2\sigma^2}\|\theta\|^2$$

Novelty $$\sum_{i=1}^{m} \max\left(-\log \frac{p(x_i|\theta)}{\exp(\rho - g(\theta))}, 0\right) + \frac{1}{2}\|\theta\|^2$$

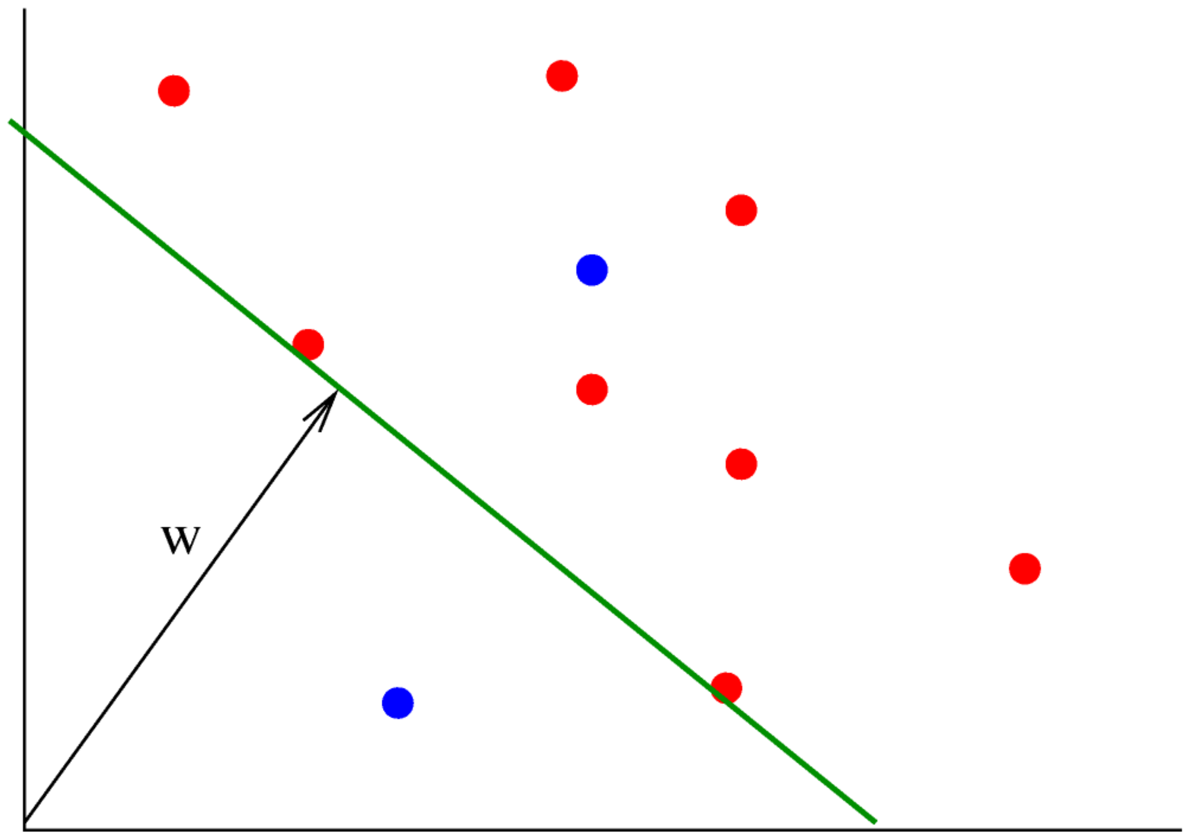
$$\sum_{i=1}^{m} \max(\rho - \langle\phi(x_i), \theta\rangle, 0) + \frac{1}{2}\|\theta\|^2$$

Advantages

- No normalization $g(\theta)$ needed
- No need to perform particularly well in high density regions (estimator focuses on low-density regions)
- Quadratic program

Maximum Distance Hyperplane

Idea Find hyperplane, given by $f(x) = \langle w, x \rangle + b = 0$ that has **maximum distance from origin** yet is still closer to the origin than the observations.



Hard Margin

$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$
$$\text{subject to} \quad \langle w, x_i \rangle \geq 1$$

Soft Margin

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} \xi_i$$
$$\text{subject to} \quad \langle w, x_i \rangle \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

Optimization Problem

Primal Problem

minimize $$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

subject to $\langle w, x_i\rangle - 1 + \xi_i \geq 0$ and $\xi_i \geq 0$

Lagrange Function L

- Subtract constraints, multiplied by Lagrange multipliers (α_i and η_i), from Primal Objective Function.
- Lagrange function L has **saddlepoint** at optimum.

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(\langle w, x_i\rangle - 1 + \xi_i\right) - \sum_{i=1}^{m}\eta_i\xi_i$$

subject to $\alpha_i, \eta_i \geq 0$.

Dual Problem

Optimality Conditions

$$\partial_w L = w - \sum_{i=1}^{m} \alpha_i x_i = 0 \implies w = \sum_{i=1}^{m} \alpha_i x_i$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \implies \alpha_i \in [0, C]$$

Now **substitute** the optimality conditions **back into** L.

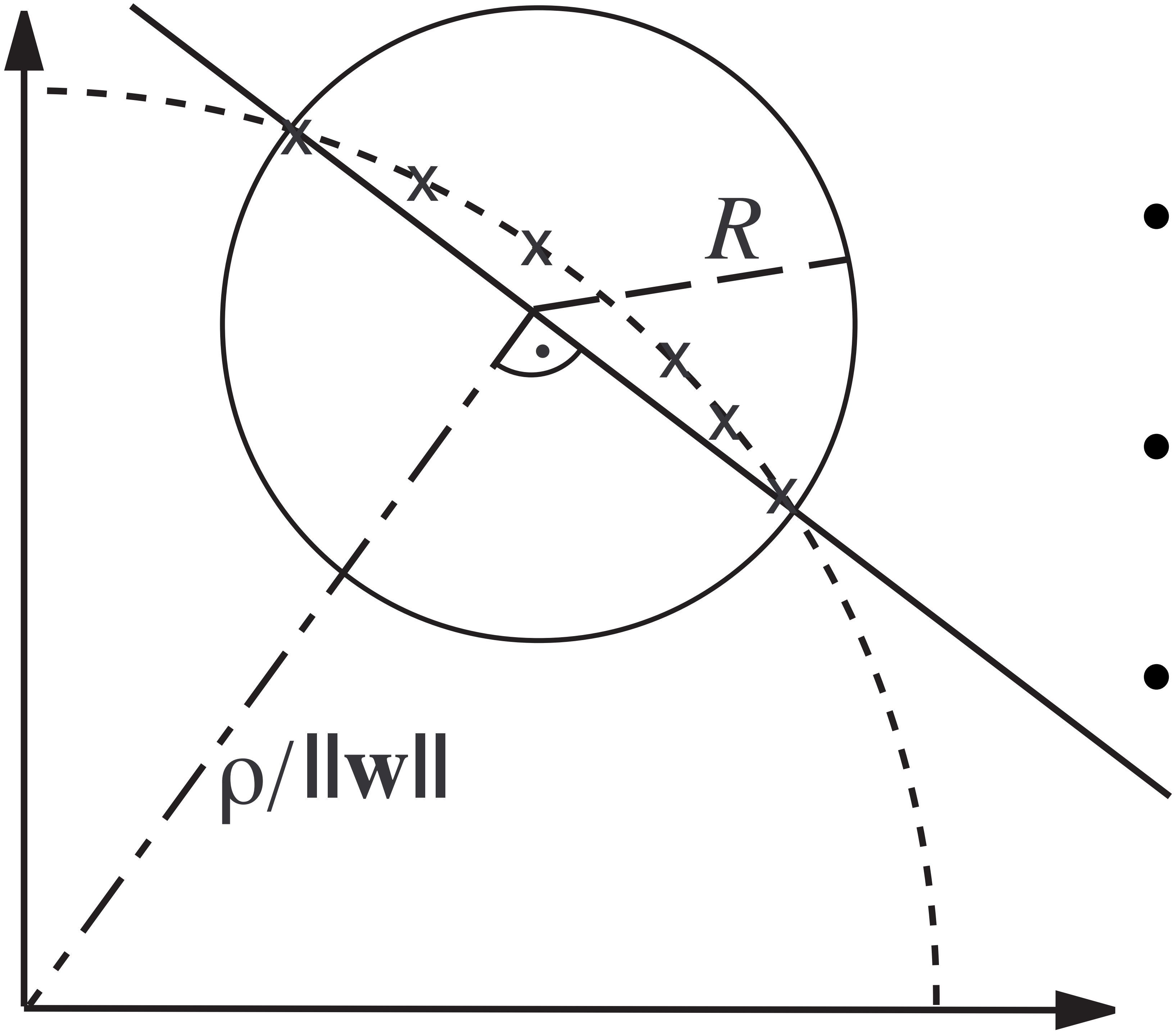
Dual Problem

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{m} \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{m} \alpha_i$$

$$\text{subject to} \quad \alpha_i \in [0, C]$$

All this is only possible due to the convexity of the primal problem.

Minimum enclosing ball



- Observations on surface of ball
- Find minimum enclosing ball
- Equivalent to single class SVM

Adaptive thresholds

Problem

- Depending on C, the number of novel points will vary.
- We would like to **specify the fraction** ν beforehand.

Solution

Use hyperplane separating data from the origin

$$H := \{x | \langle w, x \rangle = \rho\}$$

where the threshold ρ is **adaptive**.

Intuition

- Let the hyperplane shift by shifting ρ
- Adjust it such that the 'right' number of observations is considered novel.
- Do this automatically

Optimization Problem

Primal Problem

$$\text{minimize } \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \xi_i - m\nu\rho$$

$$\text{where } \langle w, x_i \rangle - \rho + \xi_i \geq 0$$

$$\xi_i \geq 0$$

Dual Problem

$$\text{minimize } \frac{1}{2}\sum_{i=1}^{m} \alpha_i \alpha_j \langle x_i, x_j \rangle$$

$$\text{where } \alpha_i \in [0, 1] \text{ and } \sum_{i=1}^{m} \alpha_i = \nu m.$$

The ν-property theorem

- Optimization problem

$$\underset{w}{\text{minimize}} \;\; \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \xi_i - m\nu\rho$$

$$\text{subject to } \langle w, x_i \rangle \geq \rho - \xi_i \text{ and } \xi_i \geq 0$$

- Solution satisfies
  - At most a fraction of ν points are novel
  - At most a fraction of (1-ν) points aren't novel
  - Fraction of points on boundary vanishes for large m (for non-pathological kernels)

Proof

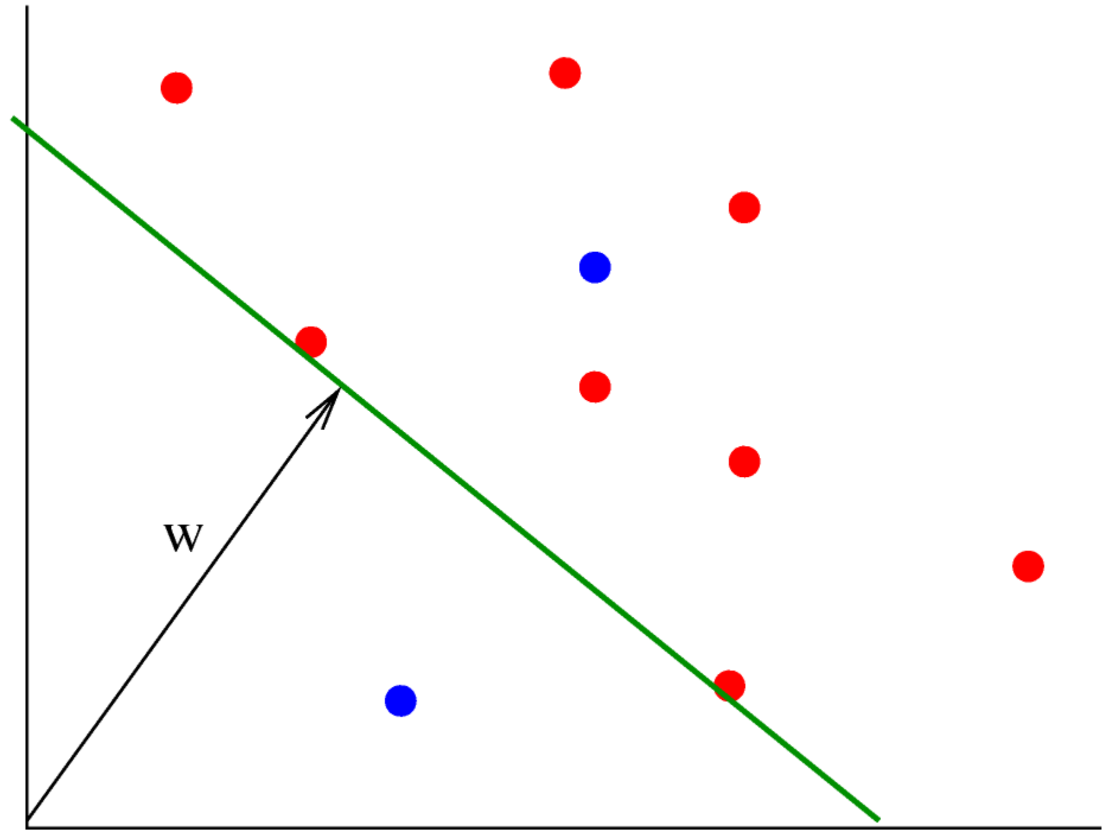
- Move boundary at optimality
  - For smaller threshold m- points on wrong side of margin contribute $\delta(m_- - \nu m) \leq 0$
  - For larger threshold m+ points not on 'good' side of margin yield

    $$\delta(m_+ - \nu m) \geq 0$$

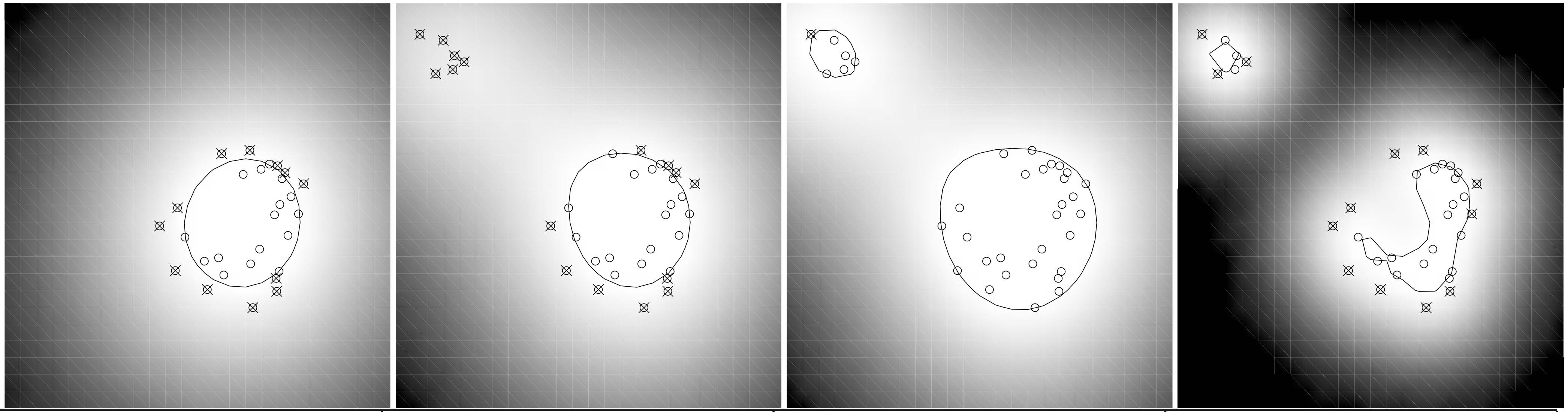
  - Combining inequalities

    $$\frac{m_-}{m} \leq \nu \leq \frac{m_+}{m}$$

- Margin set of measure 0



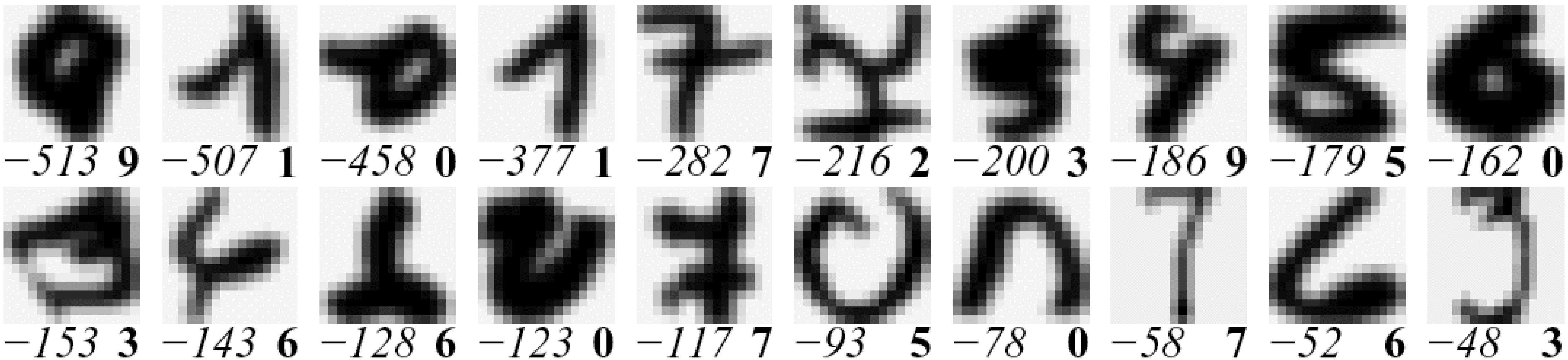
Toy example



ν, width c	0.5, 0.5	0.5, 0.5	0.1, 0.5	0.5, 0.1
frac. SVs/OLs	0.54, 0.43	0.59, 0.47	0.24, 0.03	0.65, 0.38
margin $\rho/\|\mathbf{w}\|$	0.84	0.70	0.62	0.48

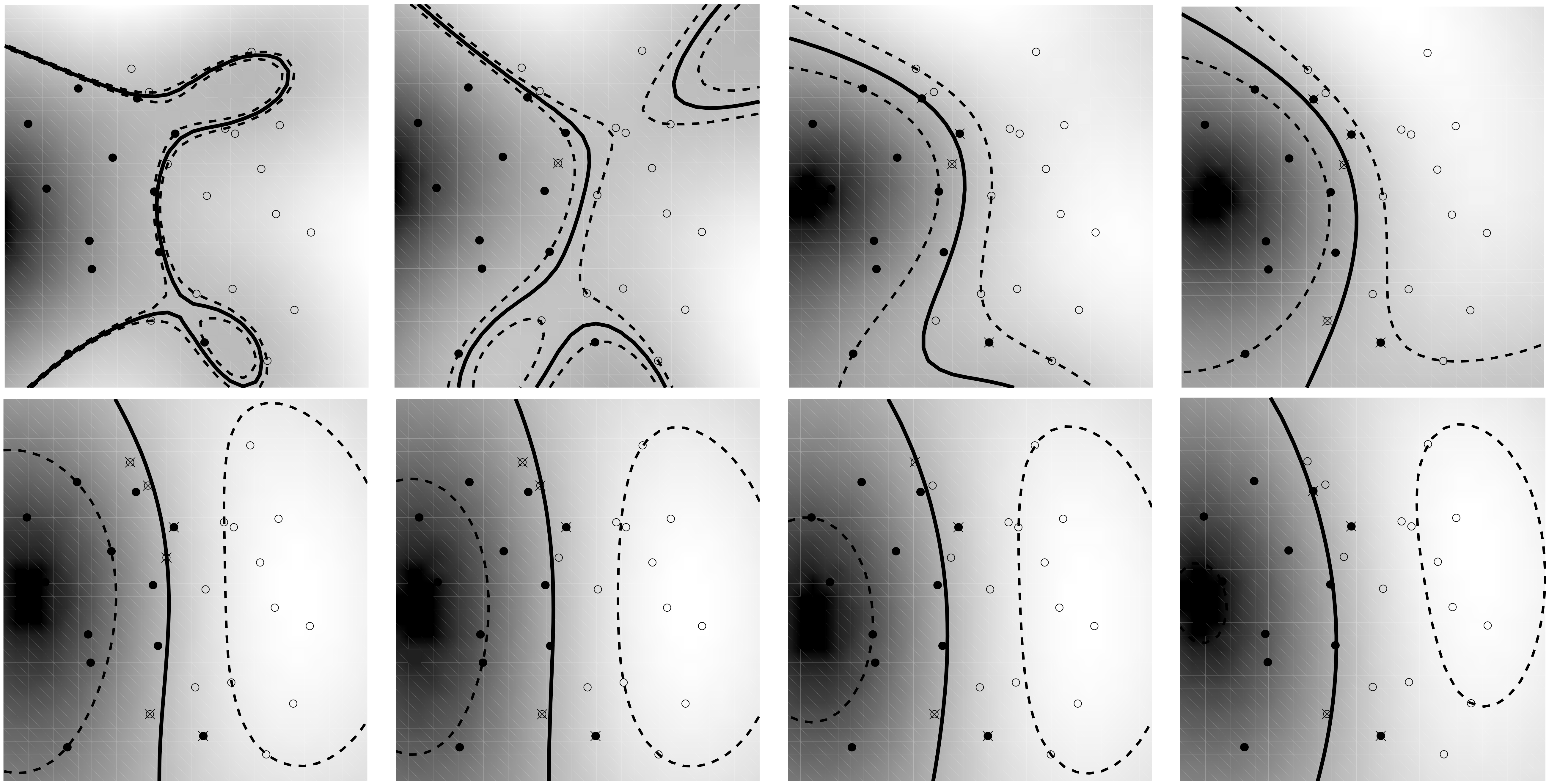
threshold and smoothness requirements

Novelty detection for OCR



- Better estimates since we only optimize in low density regions.
- Specifically tuned for small number of outliers.
- Only estimates of a level-set.
- For $\nu = 1$ we get the Parzen-windows estimator back.

Classification with the ν-trick



changing kernel width and threshold

Structured Estimation (preview)

Large Margin Condition

- Binary classifier
  Correct class chosen with large margin y f(x)
- Multiple classes
  - Score function per class f(x,y)
  - Want that correct class has much larger score than incorrect class

$$f(x, y) - f(x, y') \geq 1 \text{ for all } y' \neq y$$

- Structured loss function (e.g. coal & diamonds)

$$\Delta(y, y')$$

Large Margin Classifiers

- Large Margin without rescaling (convex) (Guestrin, Taskar, Koller)

$$l(x, y, f) = \sup_{y' \in \mathcal{Y}} \left[f(x, y') - f(x, y) + \Delta(y, y') \right]$$

- Large Margin with rescaling (convex) (Tsochantaridis, Hofmann, Joachims, Altun)

$$l(x, y, f) = \sup_{y' \in \mathcal{Y}} \left[f(x, y') - f(x, y) + 1 \right] \Delta(y, y')$$

- Both losses majorize misclassification loss

$$\Delta\left(y, \operatorname*{argmax}_{y'} f(x, y')\right)$$

- Proof by plugging argmax into the definition

Many applications

- Ranking (DCG, NDCG)
- Graph matching (linear assignment)
- ROC and F_β scores
- Sequence annotation (named entities, activity)
- Segmentation
- Natural Language Translation
- Image annotation / scene understanding

- Caution - this loss is generally not consistent!

Extensions

- Invariances
  - Add prior knowledge (e.g. in OCR)
  - Make estimates robust against malicious abuse (e.g. spam filtering)
- Tighter upper bounds
  - Convex bound can be very loose
  - Overweights noisy data
  - Structured version of ramp loss
  - Can be shown to be consistent

More Kernel Algorithms

Kernel PCA

Principal Component Analysis

- Gaussian density model

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

- Estimate variance by empirical average

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

- Good approximation by low-rank model
    - Extract leading eigenvalues of covariance
    - Data might lie in a subspace

Principal Component Analysis

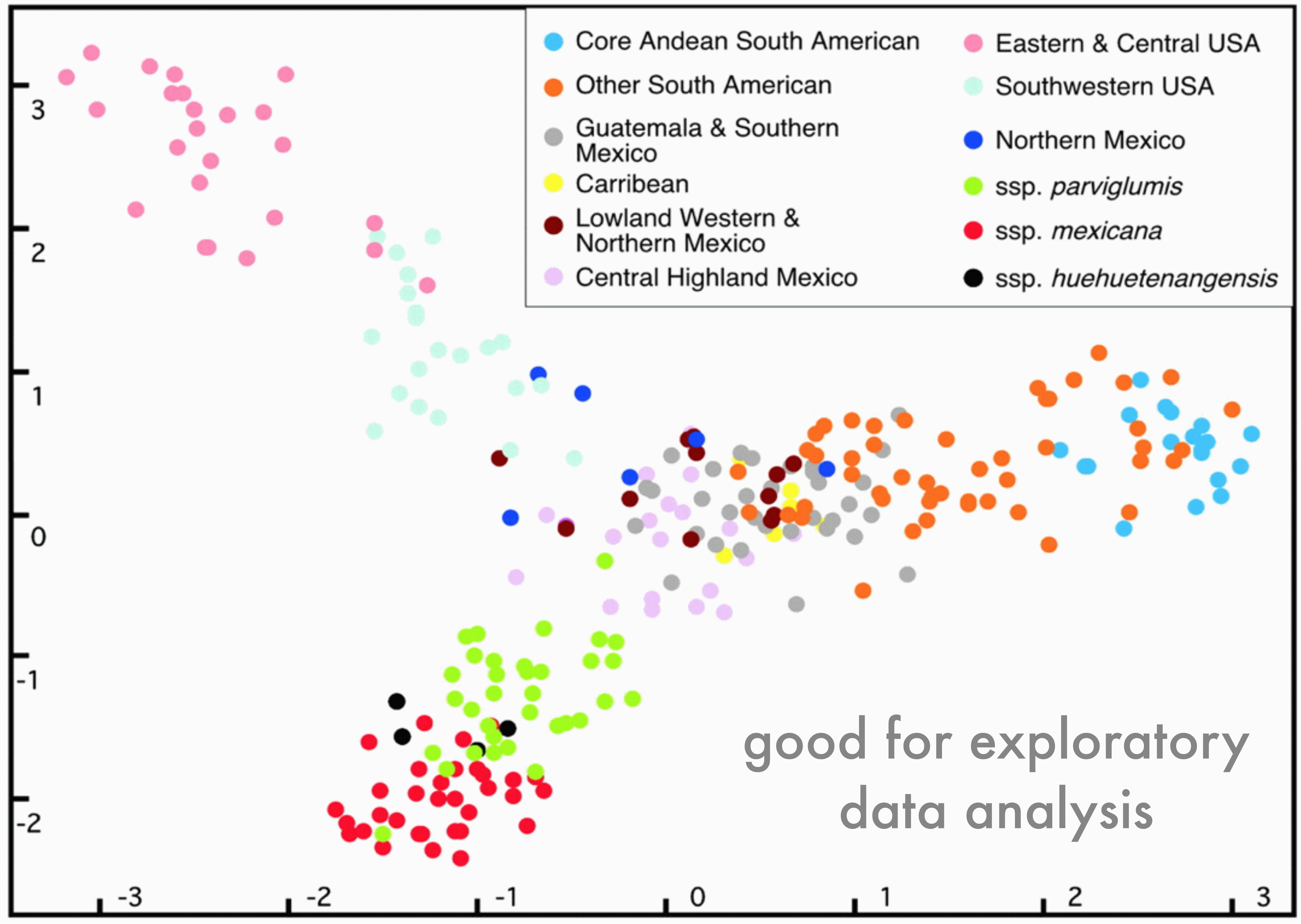
- Generative approximation of data

$$x = \sum_i \sigma_i v_i \alpha_i \text{ where } \alpha_i \sim \mathcal{N}(0,1)$$

- Heuristic
  Good explanation of data implies that we have meaningful dimensions of the data.
- Linear feature extraction

$$g_i(x) = \langle v_i, x \rangle$$

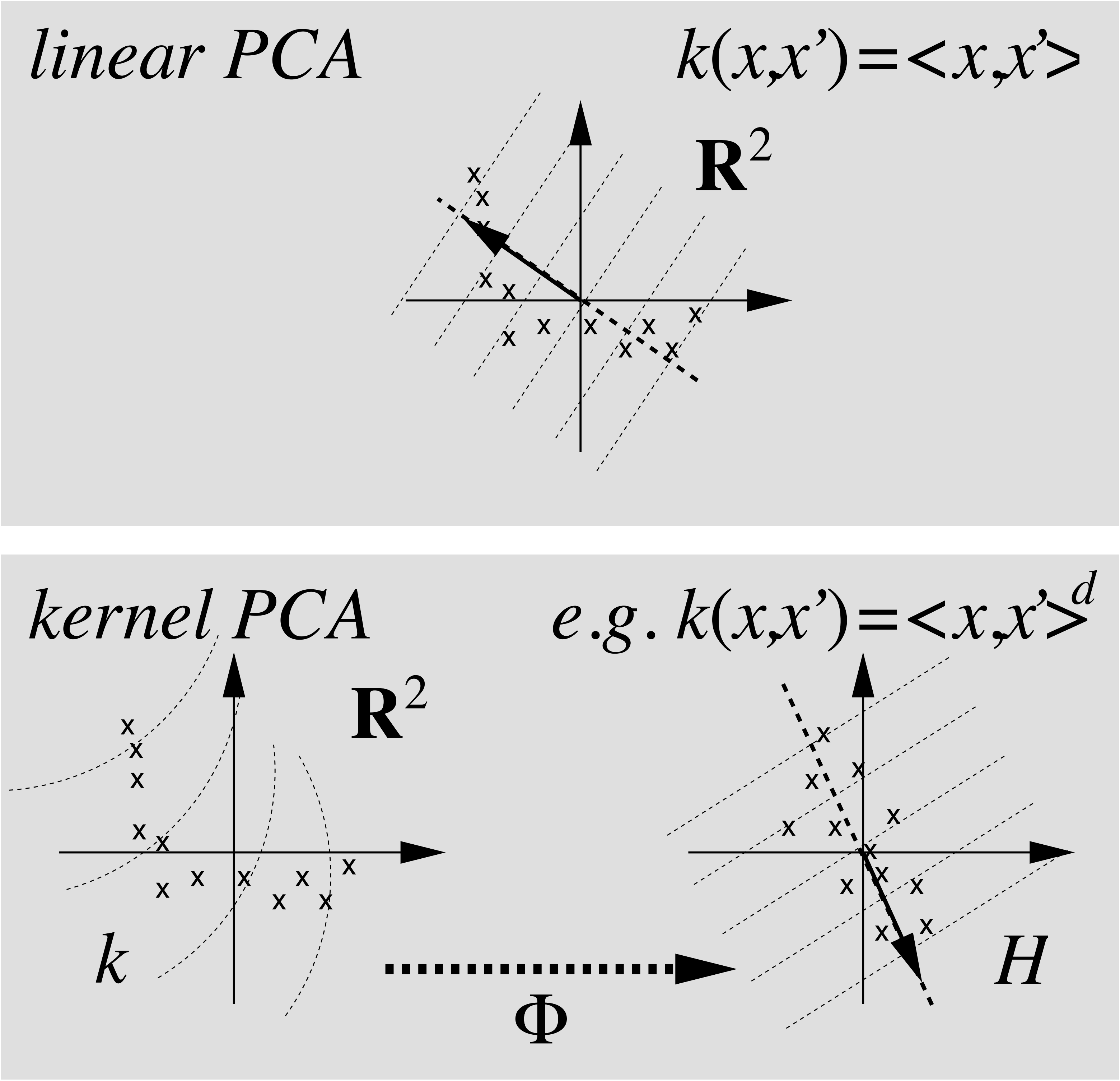
- PCA is reconstruction with smallest l_2 error



good for exploratory data analysis

http://www.plantsciences.ucdavis.edu/gepts/pb143/LEC17/pq0921251003.gif

Kernel PCA



PCA via inner products

- **Eigenvector condition** $\Sigma v = \lambda v$

$$\frac{1}{m}\sum_i \bar{x}_i \bar{x}_i^\top v = \lambda v \text{ for } \bar{x}_i = x_i - \frac{1}{m}\sum_i x_i$$

$$\text{hence } v = \sum_j \alpha_j \bar{x}_j$$

$$\text{using } \bar{x}_l^\top \frac{1}{m}\sum_i \bar{x}_i \bar{x}_i^\top v = \lambda \bar{x}_l^\top v$$

$$\text{yields } \frac{1}{m}\bar{K}\bar{K}\alpha = \lambda \bar{K}\alpha$$

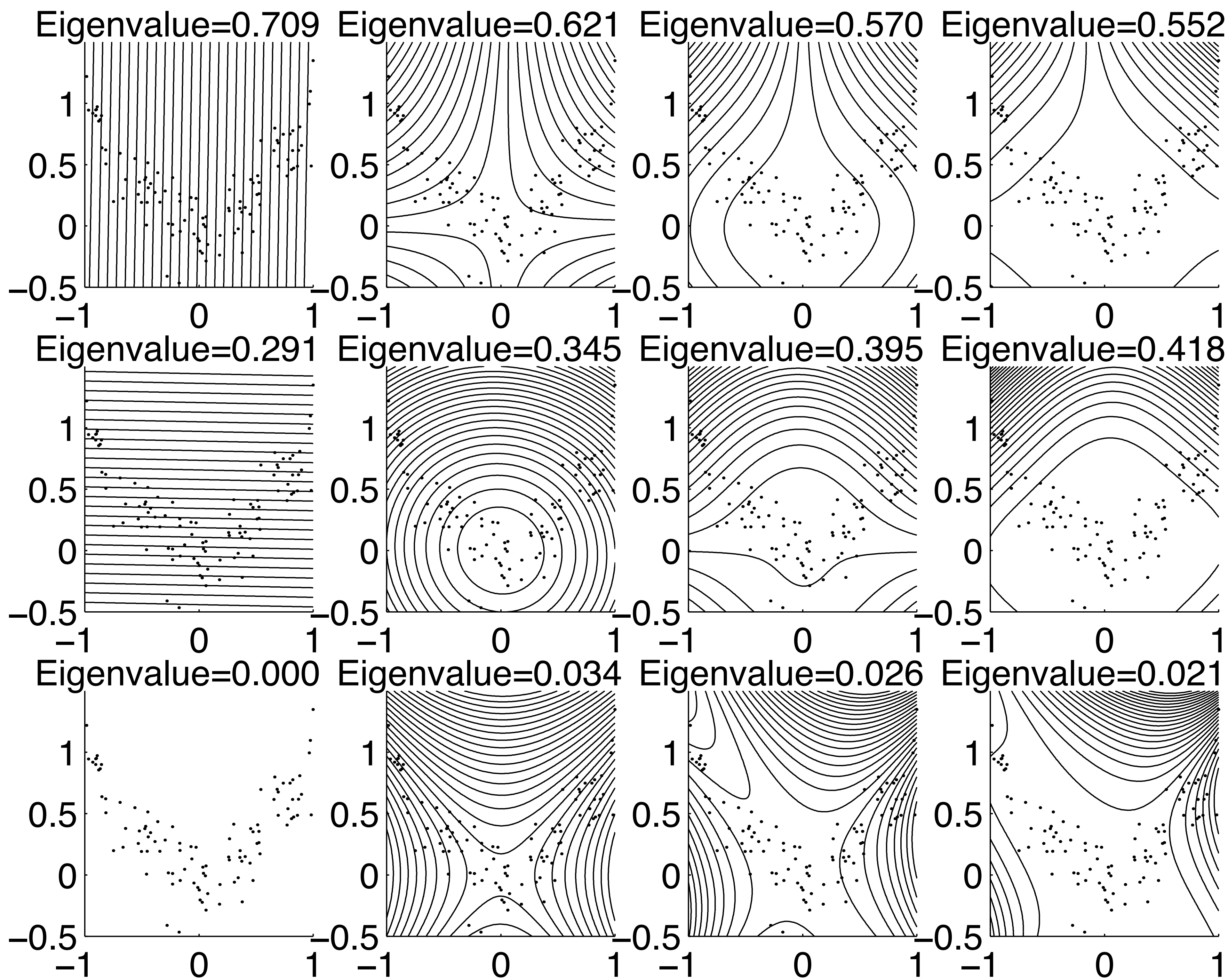
- **Kernel PCA**

$$\frac{1}{m}\bar{K}\alpha = \lambda\alpha \text{ where } \bar{K}_{ij} = \langle \bar{x}_i, \bar{x}_j \rangle$$

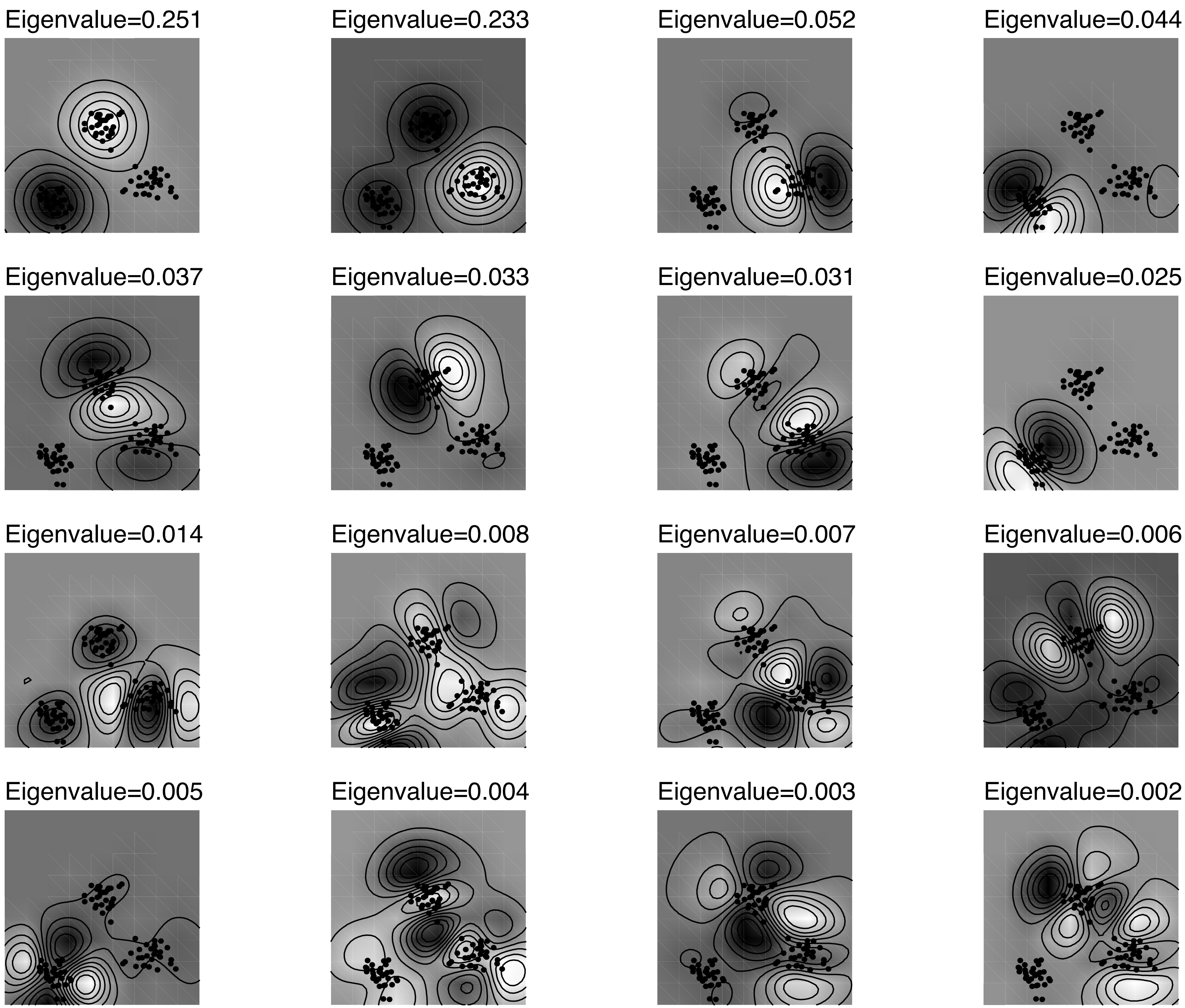
Two dimensional feature extraction

noisy parabola

polynomials of increasing order (1 is PCA)

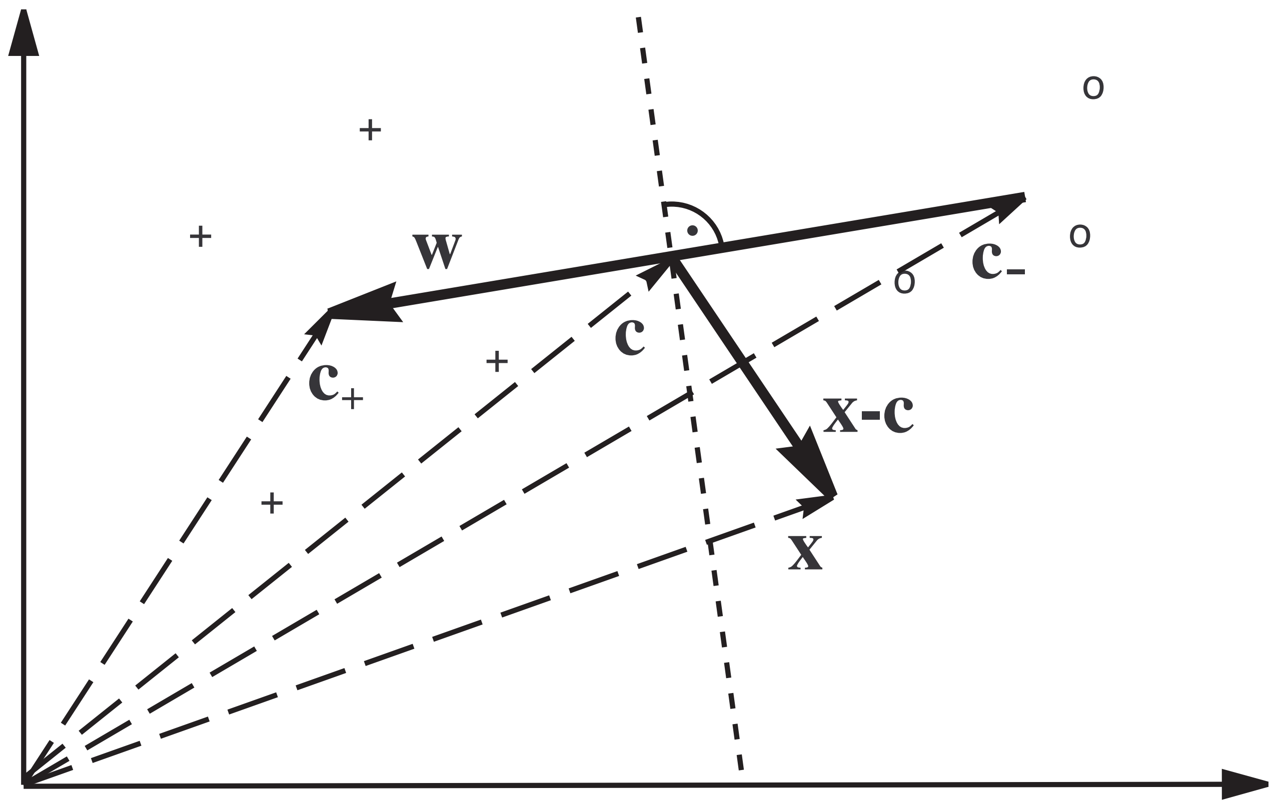


Feature extraction



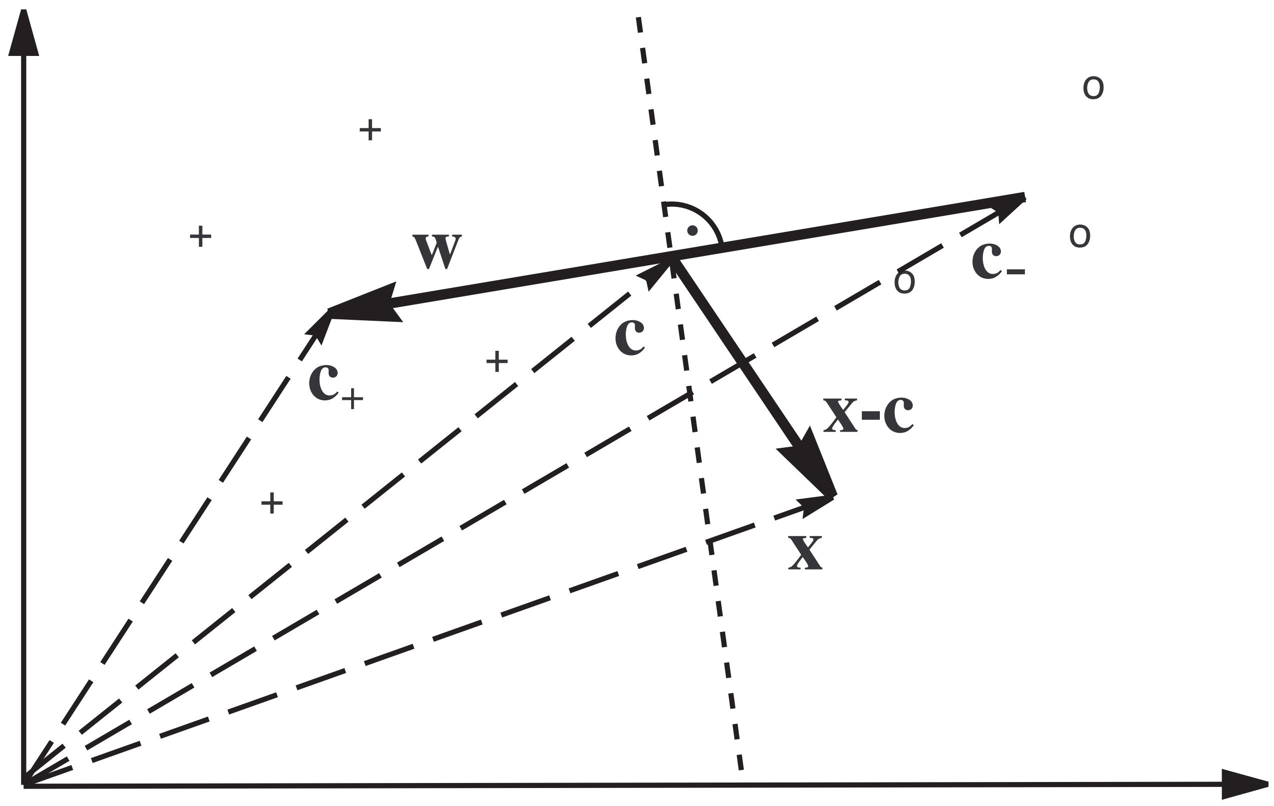
Mean Classifier

'Trivial' classifier



- Represent each class by mean in feature space
- Classify along direction of maximum discrepancy between classes
- Trivial to 'train'

'Trivial' classifier



- **Class mean**

$$\mu_+ = \frac{1}{m_+} \sum_{i:y_i=1} \phi(x_i) \text{ and } \mu_- = \frac{1}{m_-} \sum_{i:y_i=-1} \phi(x_i)$$

- **Classifier**

$$f(x) = \langle \mu_+ - \mu_-, \phi(x) \rangle = \sum_i \frac{y_i}{m_{y_i}} k(x_i, x)$$

like Watson Nadaraya

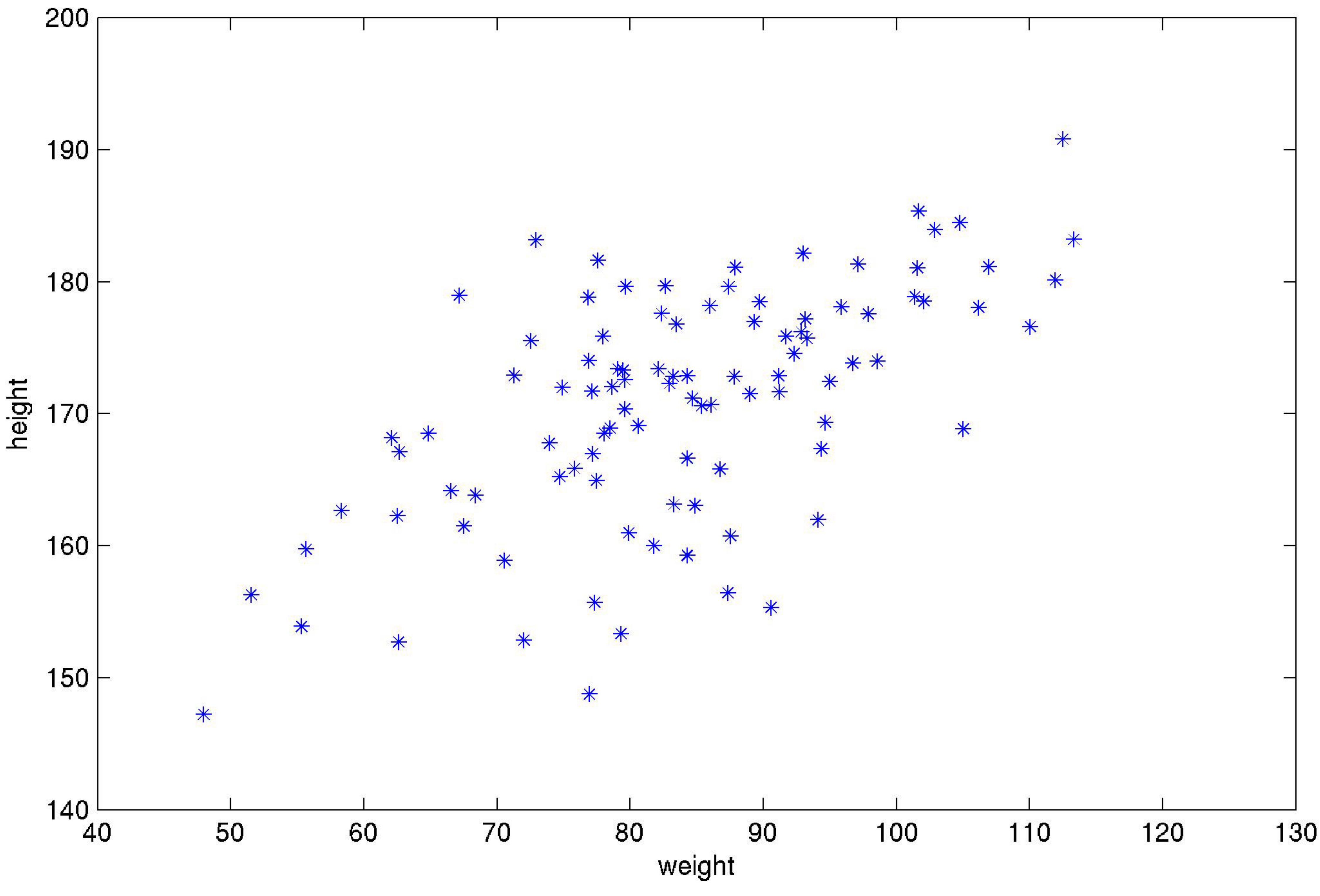
More kernel methods

- Canonical Correlation analysis
- Two sample test
  - Mean in feature space is sufficient to fully represent a distribution
  - Compare them by computing distance
- Independence test
  - Compare joint and product of marginals
- Structured feature extraction
  - Find directions of high significance and low function complexity

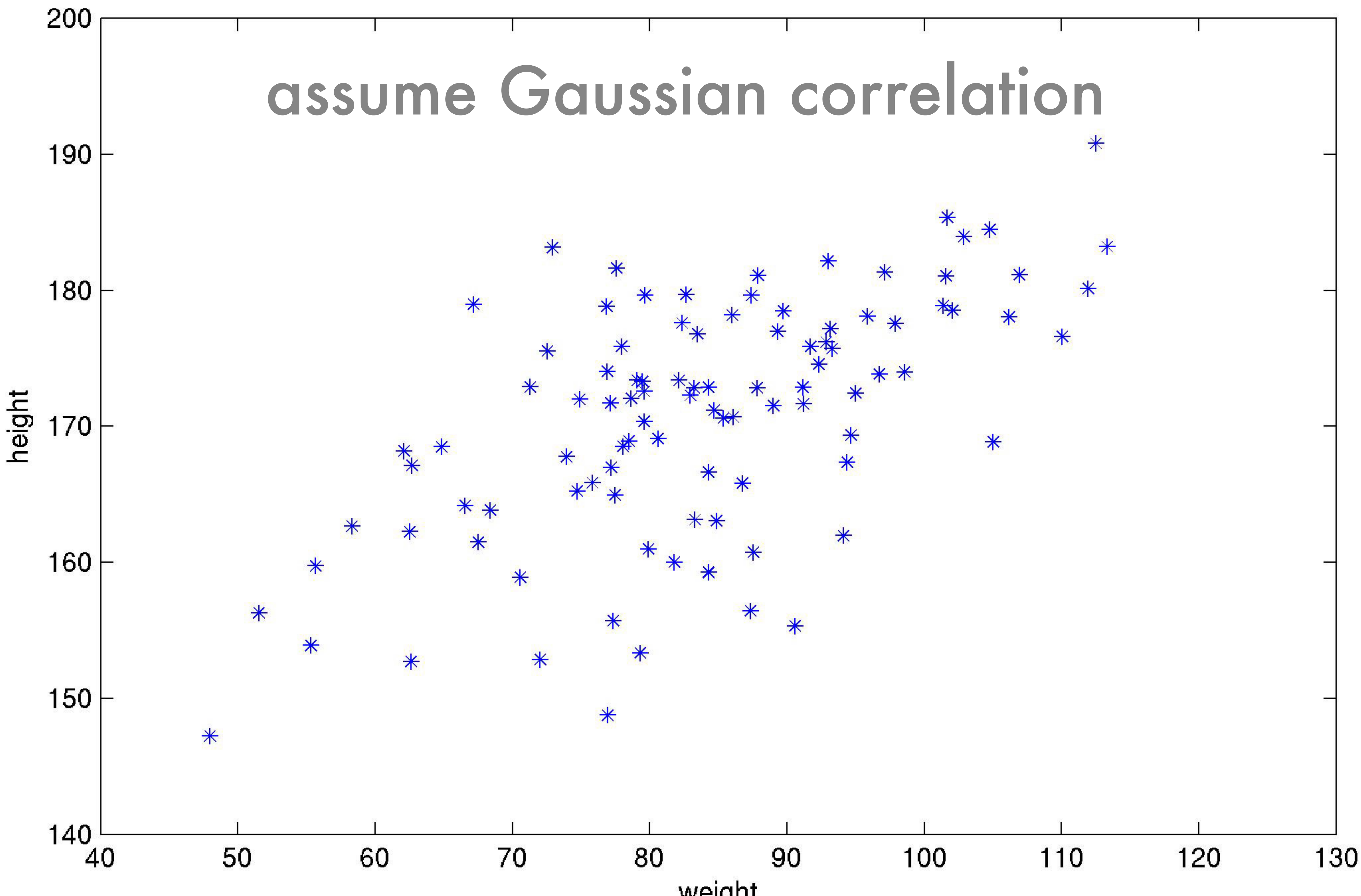
Conditional Models

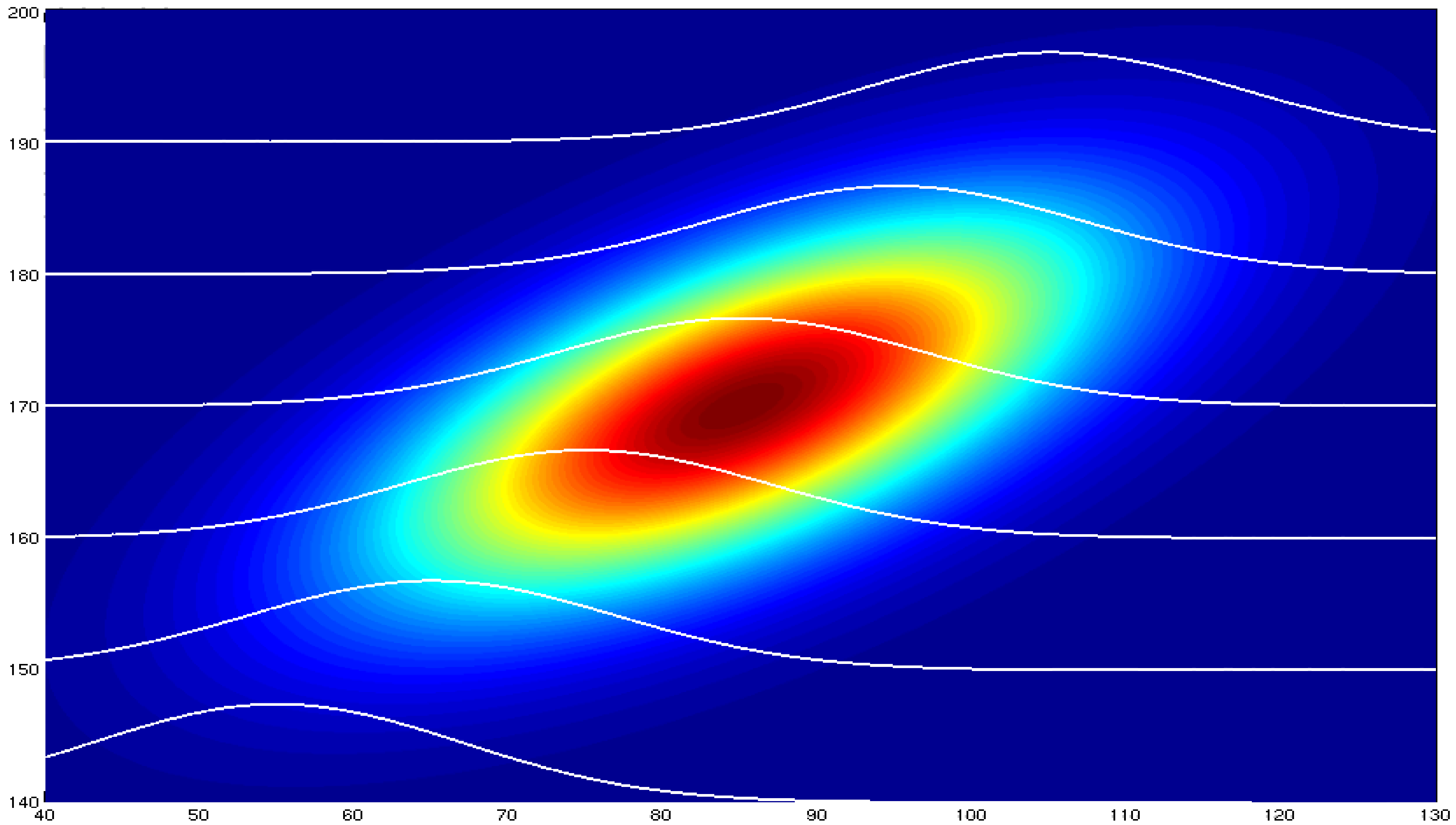
Gaussian Processes

Weight & height

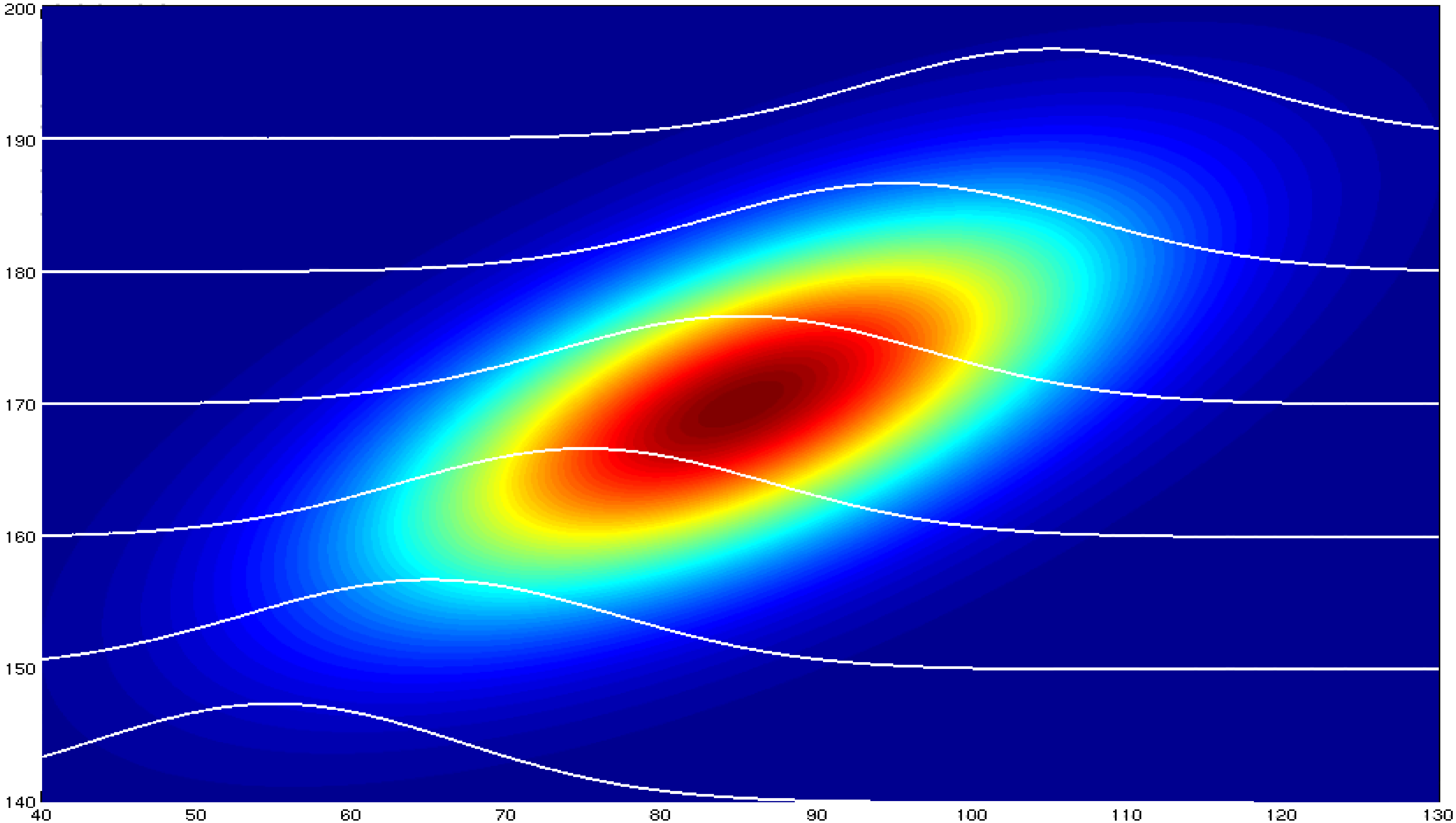


Weight & height





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



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

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

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

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 $\mathcal{X} = \{\mathrm{age}, \mathrm{height}, \mathrm{weight}, \ldots\}$.

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 $\mu = 0$.

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 . . .

The connection

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

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

Conclusion

A small weight vector in “feature space”, as commonly used in SVM amounts to observing t with high $p(t)$.

Log prior $-\log p(t) \Longleftrightarrow$ **Margin** $\|w\|^2$

Will get back to this later again.

Regression

Joint Gaussian Model

- Random variables (t,t') are drawn from GP
- Observe a subset t of them
- Predict the rest 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
- Predictive variance vanishes if K is rank deficient

Some kernels

Observation

Any function k leading to a symmetric matrix with non-negative eigenvalues is a valid covariance function.

Necessary and sufficient condition (Mercer's Theorem)

k needs to be a nonnegative integral kernel.

Examples of kernels $k(x, x')$

Linear	$\langle x, x' \rangle$
Laplacian RBF	$\exp\left(-\lambda \|x - x'\|\right)$
Gaussian RBF	$\exp\left(-\lambda \|x - x'\|^2\right)$
Polynomial	$\left(\langle x, x' \rangle + c\rangle\right)^d, c \geq 0,\ d \in \mathbb{N}$
B-Spline	$B_{2n+1}(x - x')$
Cond. Expectation	$\mathbf{E}_c[p(x\|c)p(x'\|c)]$

Linear 'GP regression'

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

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

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

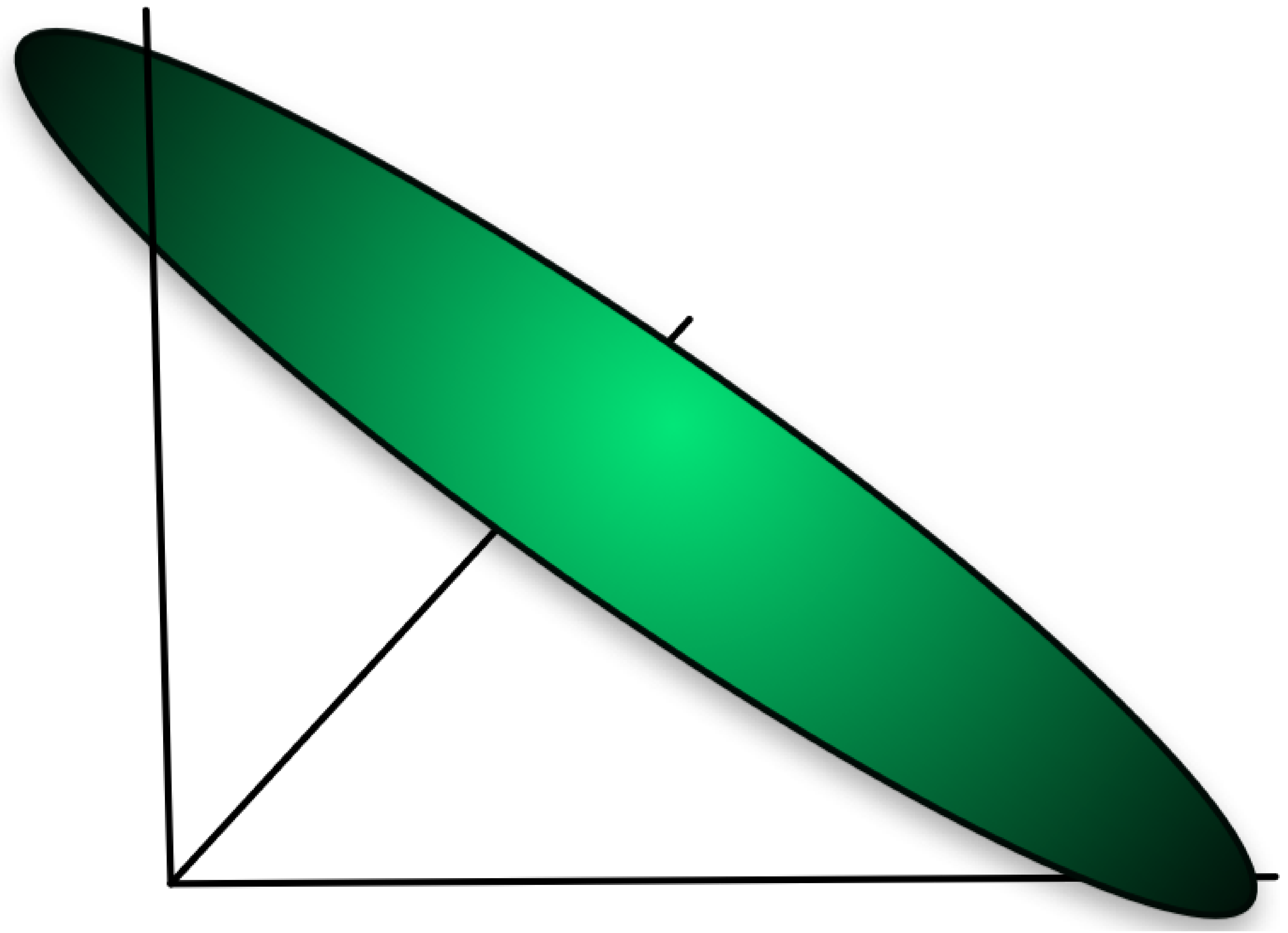
$$\tilde{\mu} = X'^\top \left[X (X^\top X)^{-1} t \right]$$

- $\tilde{\mu}$ 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



Additive Noise

Indirect Model

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

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

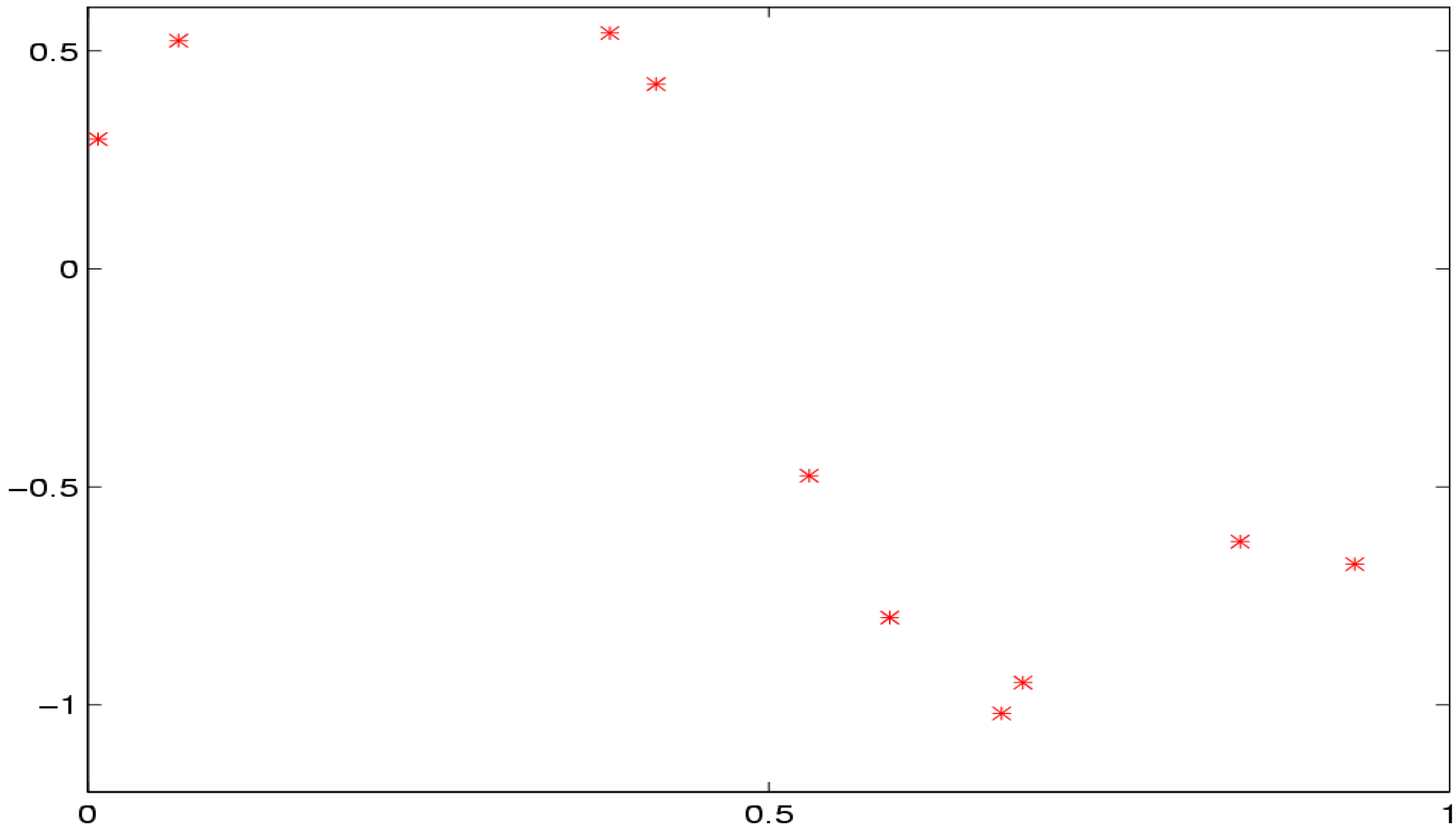
where we can now find a maximum a posteriori solution for t by maximizing the integrand (we will use this later).

Additive Normal Noise

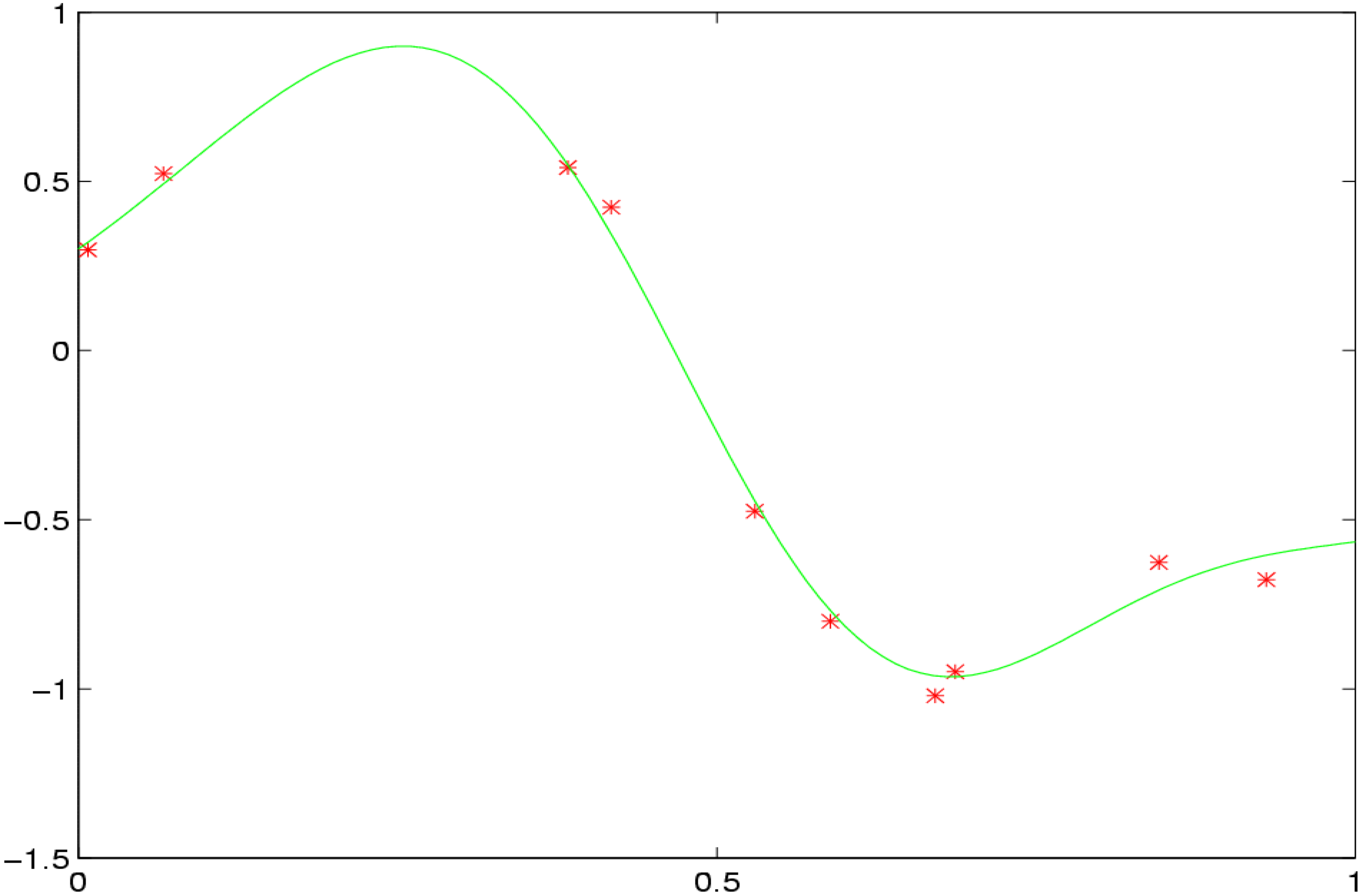
- If $\xi \sim \mathcal{N}(0, \sigma^2)$ then y is the sum of two Gaussian random variables.
- Means and variances **add up**.

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

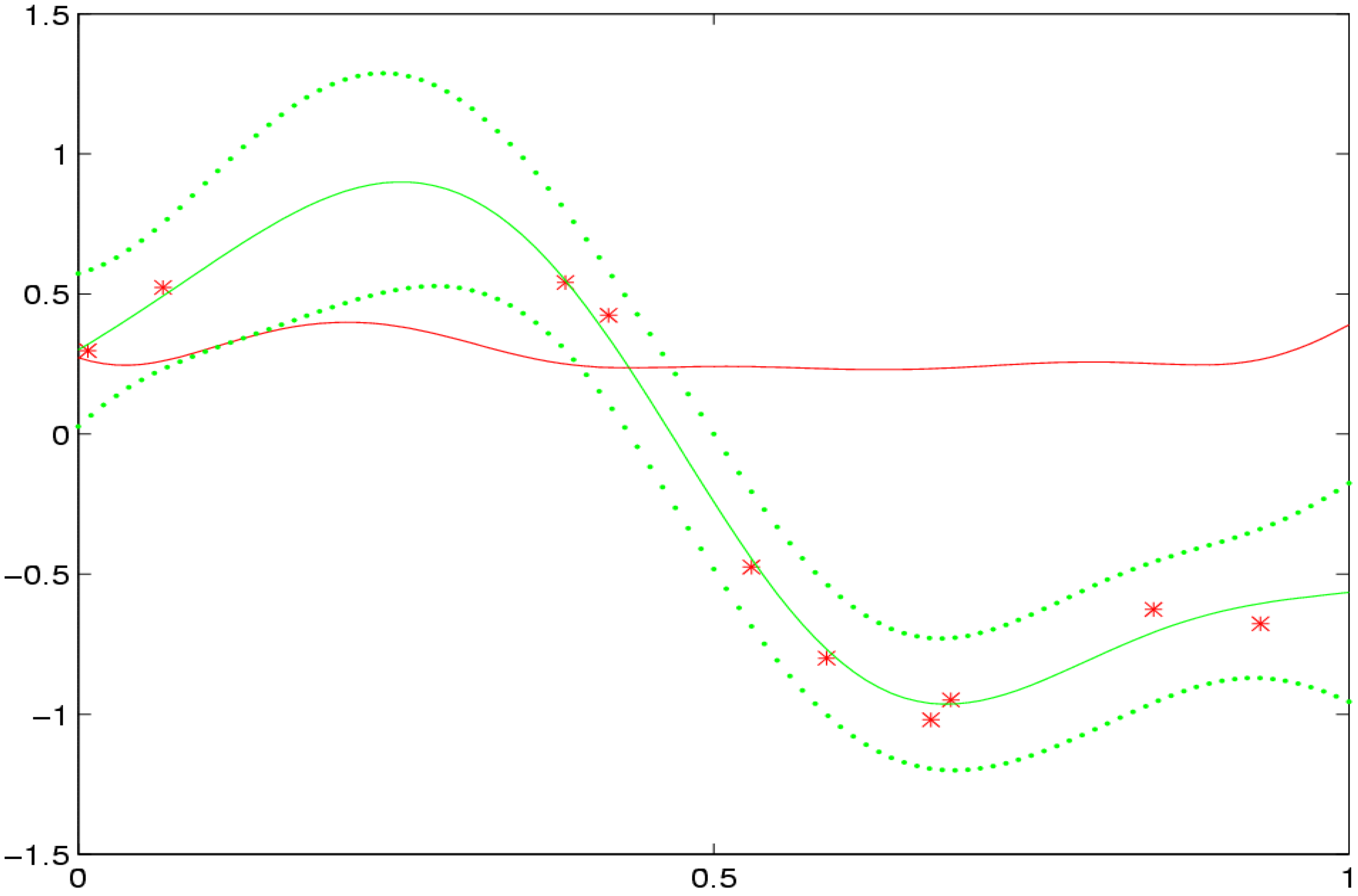
Data



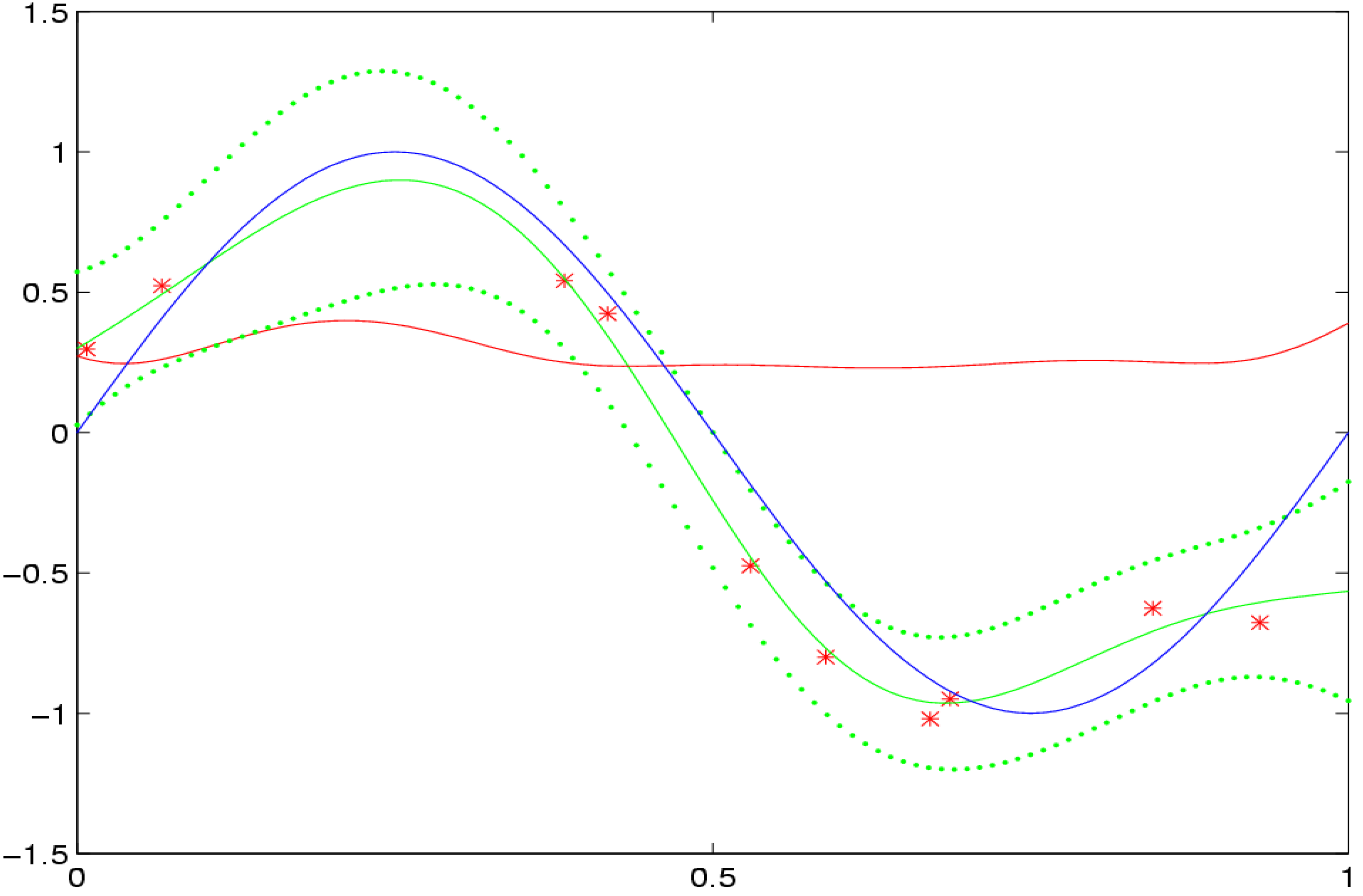
Predictive mean $k(x,X)^\top(K(X,X)+\sigma^2\mathbf{1})^{-1}y$



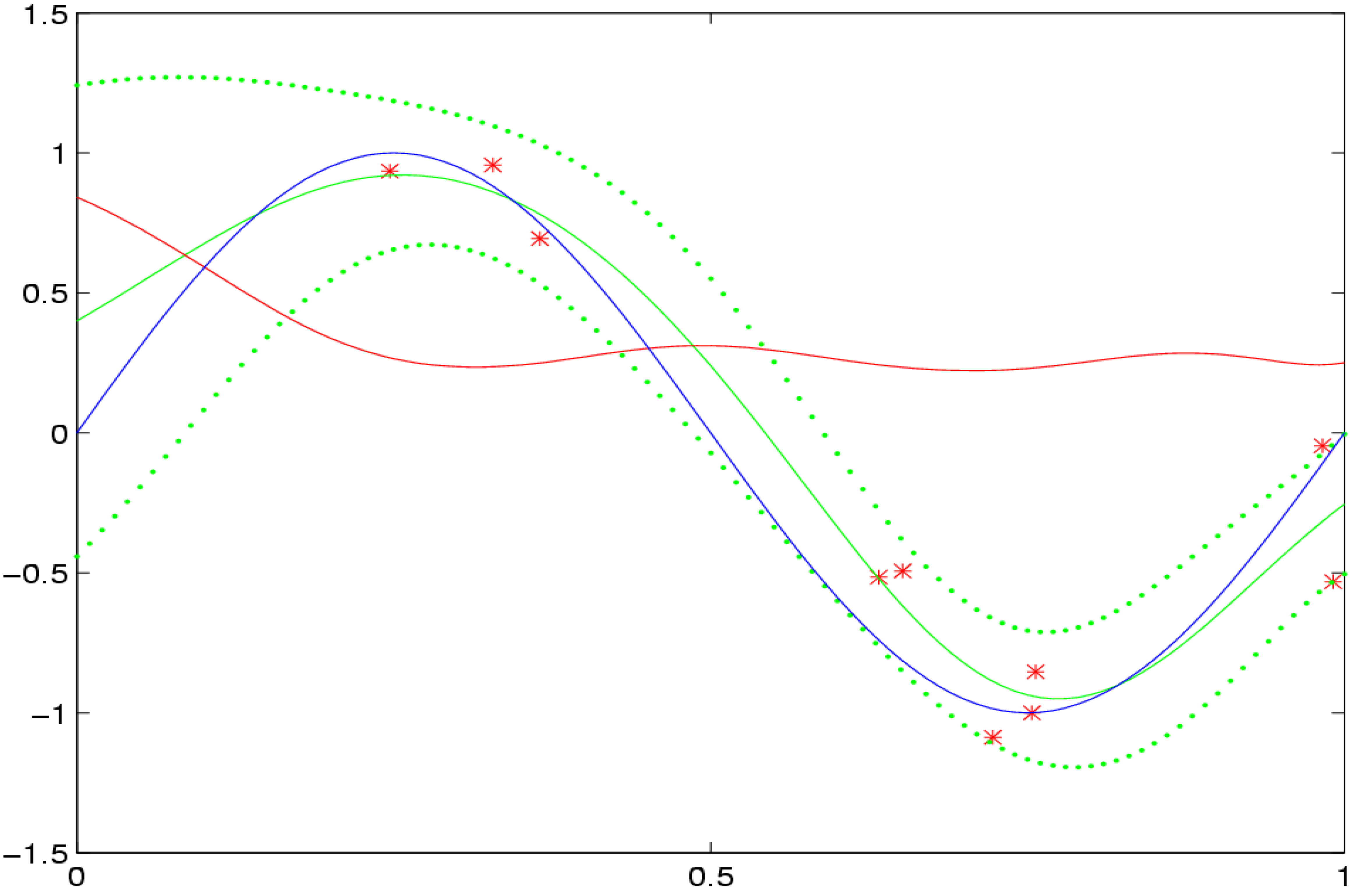
Variance



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

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

Pointwise prediction

$$\begin{aligned} K_{tt} &= K + \sigma^2 \mathbf{1} \\ K_{t't'} &= k(x, x) + \sigma^2 \\ K_{tt'} &= (k(x_1, x), \ldots, k(x_m, x)) \end{aligned}$$

Plug this into the mean and covariance equations.

Gaussian Process Conditional Models

Exponential Families

Exponential Families

- Density function

$$p(x;\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

$$\text{where } g(\theta) = \log \sum_{x'} \exp\left(\langle \phi(x'), \theta \rangle\right)$$

Exponential Families

- Density function

$$p(x;\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

$$\text{where } g(\theta) = \log \sum_{x'} \exp\left(\langle \phi(x'), \theta \rangle\right)$$

- Log partition function generates cumulants

$$\partial_\theta g(\theta) = \mathbf{E}\left[\phi(x)\right]$$

$$\partial_\theta^2 g(\theta) = \mathrm{Var}\left[\phi(x)\right]$$

Exponential Families

- Density function

$$p(x;\theta) = \exp\left(\langle\phi(x),\theta\rangle - g(\theta)\right)$$
$$\text{where } g(\theta) = \log \sum_{x'} \exp\left(\langle\phi(x'),\theta\rangle\right)$$

- Log partition function generates cumulants

$$\partial_\theta g(\theta) = \mathbf{E}\left[\phi(x)\right]$$
$$\partial_\theta^2 g(\theta) = \mathrm{Var}\left[\phi(x)\right]$$

- g is convex (second derivative is p.s.d.)

Conditional Exponential Families

$$p(y|x;\theta) = \exp\left(\langle\phi(x,y),\theta\rangle - g(\theta|x)\right)$$

$$\text{where } g(\theta|x) = \log\sum_{y'}\exp\left(\langle\phi(x,y'),\theta\rangle\right)$$

$$\partial_\theta g(\theta|x) = \mathbf{E}\left[\phi(x,y)|x\right]$$

$$\partial_\theta^2 g(\theta|x) = \mathrm{Var}\left[\phi(x,y)|x\right]$$

Conditional Exponential Families

- Density function

$$p(y|x;\theta) = \exp\left(\langle \phi(x,y), \theta \rangle - g(\theta|x)\right)$$

$$\text{where } g(\theta|x) = \log \sum_{y'} \exp\left(\langle \phi(x,y'), \theta \rangle\right)$$

$$\partial_\theta g(\theta|x) = \mathbf{E}\left[\phi(x,y)|x\right]$$

$$\partial^2_\theta g(\theta|x) = \mathrm{Var}\left[\phi(x,y)|x\right]$$

Conditional Exponential Families

- Density function

$$p(y|x;\theta) = \exp\left(\langle\phi(x,y),\theta\rangle - g(\theta|x)\right)$$

$$\text{where } g(\theta|x) = \log \sum_{y'} \exp\left(\langle\phi(x,y'),\theta\rangle\right)$$

- Log partition function generates cumulants

$$\partial_\theta g(\theta|x) = \mathbf{E}\left[\phi(x,y)|x\right]$$

$$\partial_\theta^2 g(\theta|x) = \mathrm{Var}\left[\phi(x,y)|x\right]$$

Conditional Exponential Families

- Density function

$$p(y|x;\theta) = \exp\left(\langle \phi(x,y), \theta \rangle - g(\theta|x)\right)$$

$$\text{where } g(\theta|x) = \log \sum_{y'} \exp\left(\langle \phi(x,y'), \theta \rangle\right)$$

- Log partition function generates cumulants

$$\partial_\theta g(\theta|x) = \mathbf{E}\left[\phi(x,y)|x\right]$$

$$\partial^2_\theta g(\theta|x) = \mathrm{Var}\left[\phi(x,y)|x\right]$$

- g is convex (second derivative is p.s.d.)

Key Idea

- Gaussian Process indexed by (x,y)
  - Binary y yields classification
  - Set for y yields multiclass
  - Integer y yields Poisson regression
  - Scalar y yields heteroscedastic regression
  - Sequence for y yields CRF
  - ... and lots more ...
- The GP is in the latent variables
  (Regression is special case where we can integrate)

Conditional GP Model

- Data likelihood

$$p(y|x, t(x)) := e^{t(x,y) - g(t(x))}$$

$$\text{where } g(t(x)) = \sum_y e^{t(x,y)}$$

- Prior

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

- Posterior distribution

$$p(t|X, Y) \propto \exp\left(\sum_i t(x_i, y_i) - g(t(x_i)) - \frac{1}{2} t^\top K^{-1} t\right)$$

- Maximize with respect to t for MAP estimate

Logistic Regression

Binomial Model

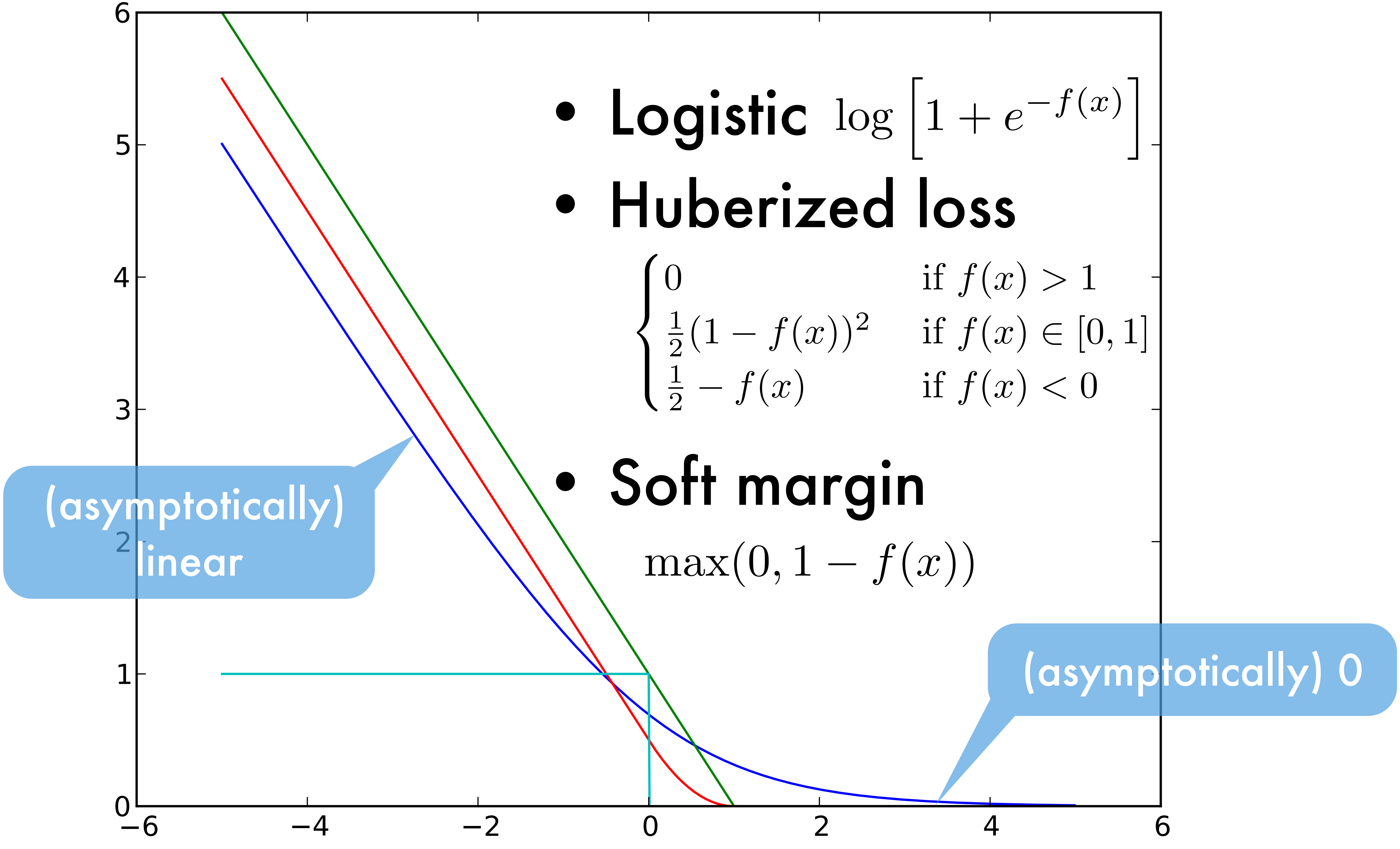
- Binary label space {-1, 1}
- We can center t(x,y) as y t(x) (constant offset doesn't change model)
- Log-likelihood

$$-\log p(y|t) = \log\left[e^{t} + e^{-t}\right] - yt = \log\left[1 + e^{-2yt}\right]$$

- After rescaling by 2 this is the logistic loss
- MAP estimation problem

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

More loss functions



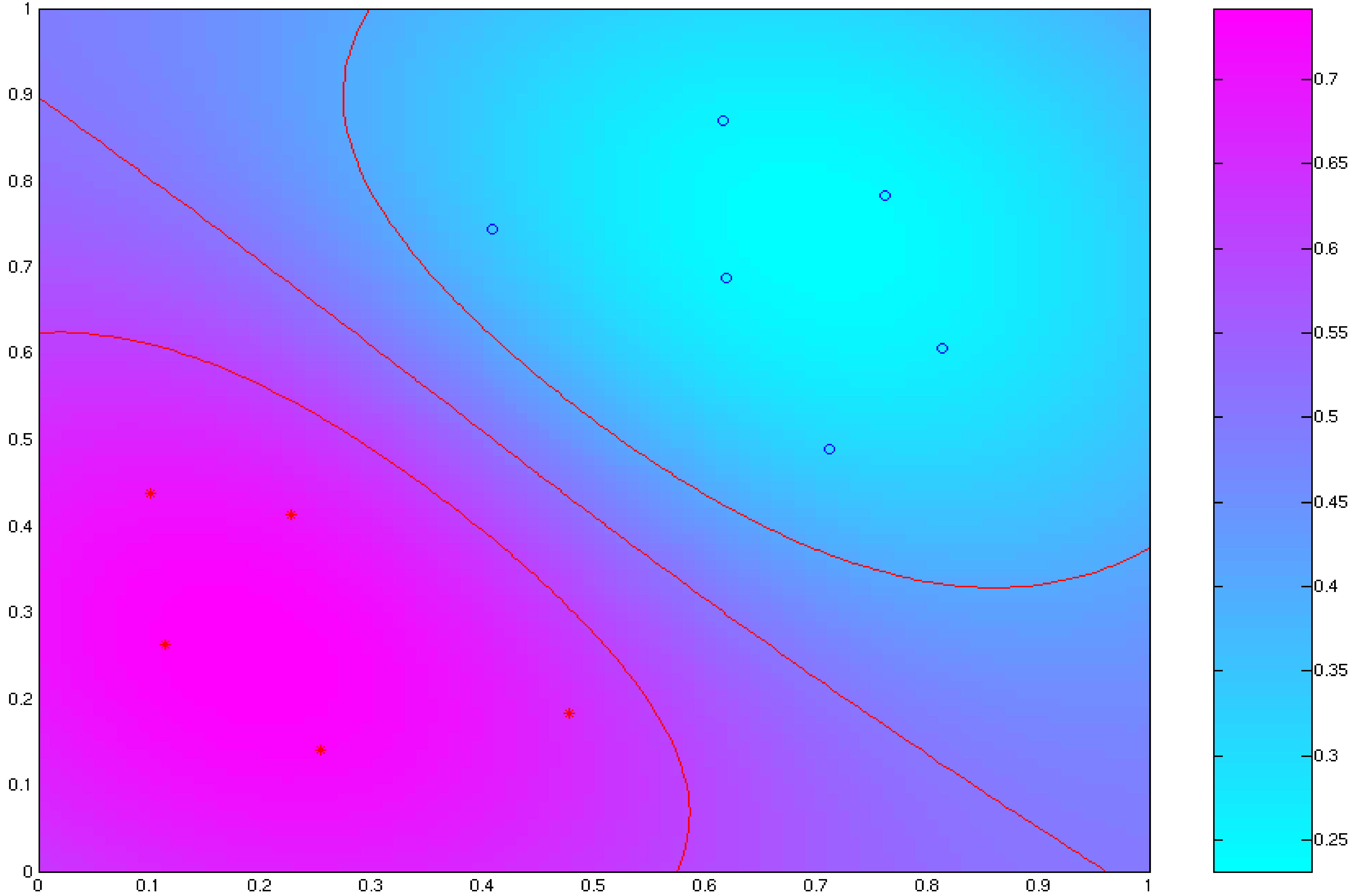
- **Logistic** $\log\left[1 + e^{-f(x)}\right]$
- **Huberized loss**

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

- **Soft margin**

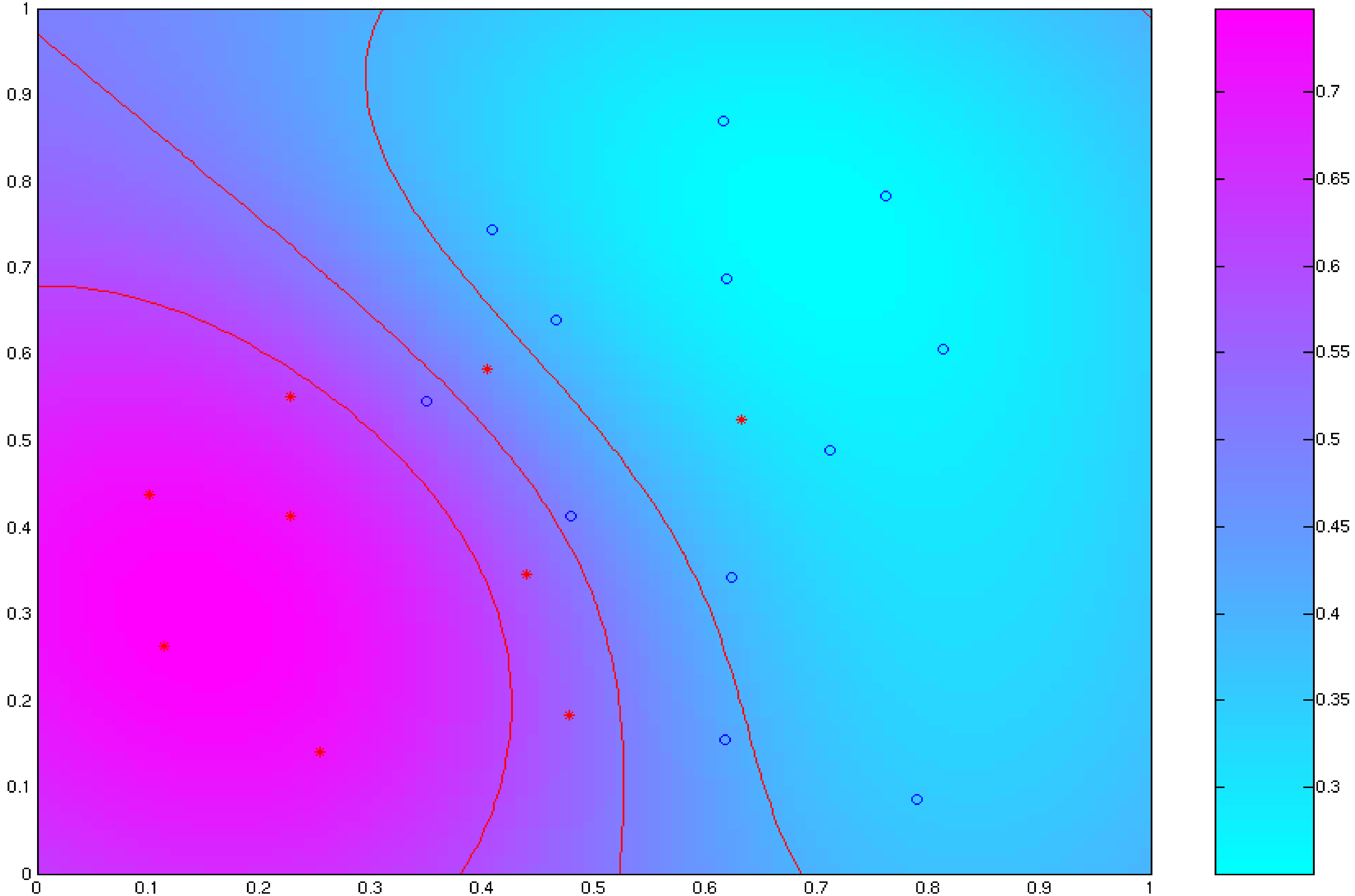
$$\max(0, 1 - f(x))$$

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

Heteroscedastic Estimation

Motivation

- GP Regression has variance estimate independent of observed data
- Assumes that we know variance globally beforehand
- **This is nonsense!**
- Estimate mean and variance jointly
- Easily possible in an exponential family model

Le, Canu, Smola, 2005

Recall - Normal distributions

Engineer's favorite

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \text{ where } x \in \mathbb{R} =: \mathcal{X}$$

Massaging the math

$$p(x) = \exp\Big(\langle \underbrace{(x, -0.5x^2)}_{\phi(x)}, \theta\rangle - \underbrace{\left(\frac{\mu^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2)\right)}_{g(\theta)}\Big)$$

Using the substitution $\theta_2 := \sigma^{-2}$ and $\theta_1 := \mu\sigma^{-2}$ yields

$$g(\theta) = \frac{1}{2}\left[\theta_1^2\theta_2^{-1} + \log 2\pi - \log\theta_2\right]$$

Basic Idea

Sufficient Statistic

We pick $\phi(x, y) = (y\phi_1(x), y^2\phi_2(x))$, that is

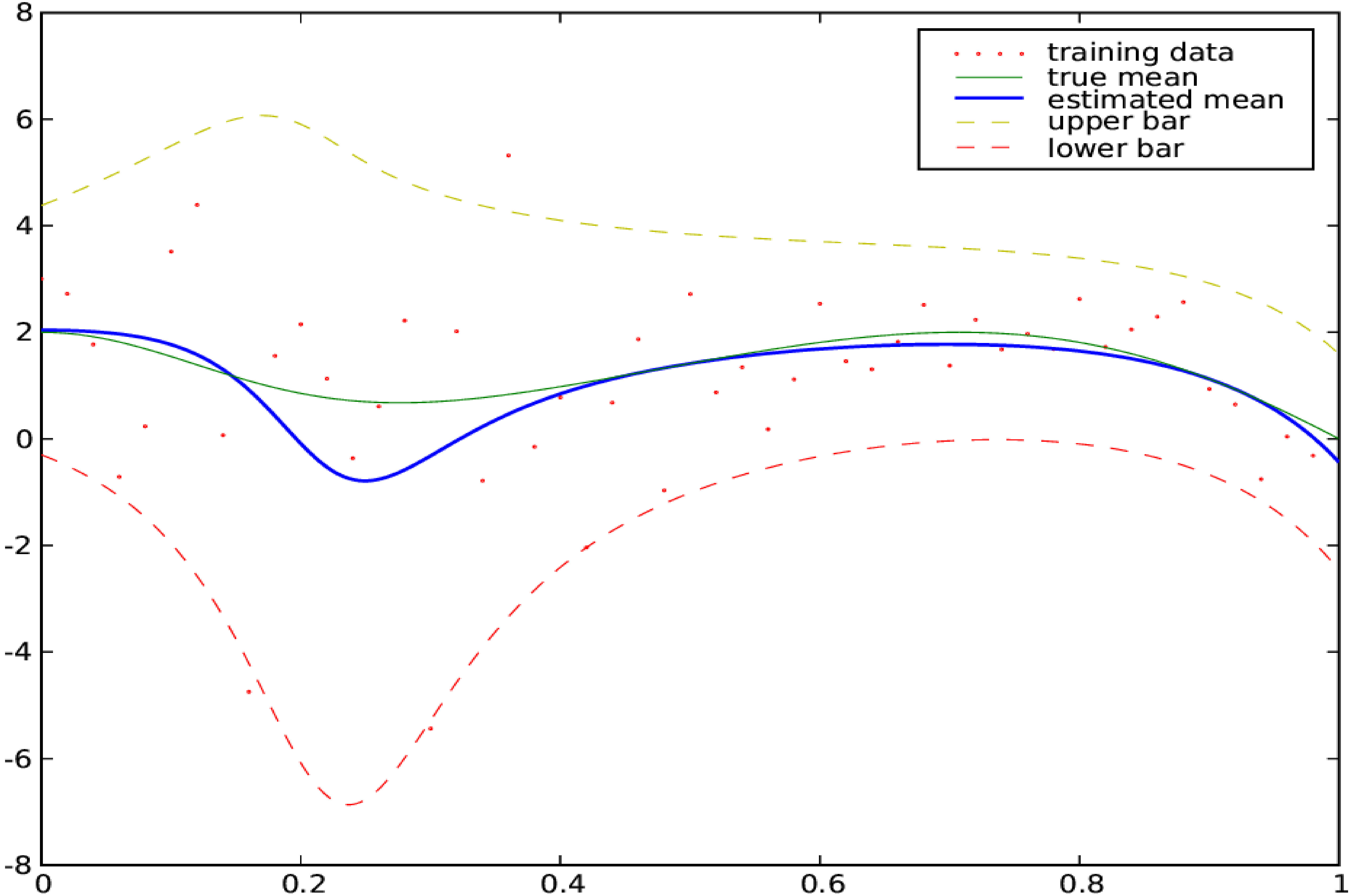
$k((x, y), (x', y')) = k_1(x, x')yy' + k_2(x, x')y^2y'^2$ where $y, y' \in \mathbb{R}$

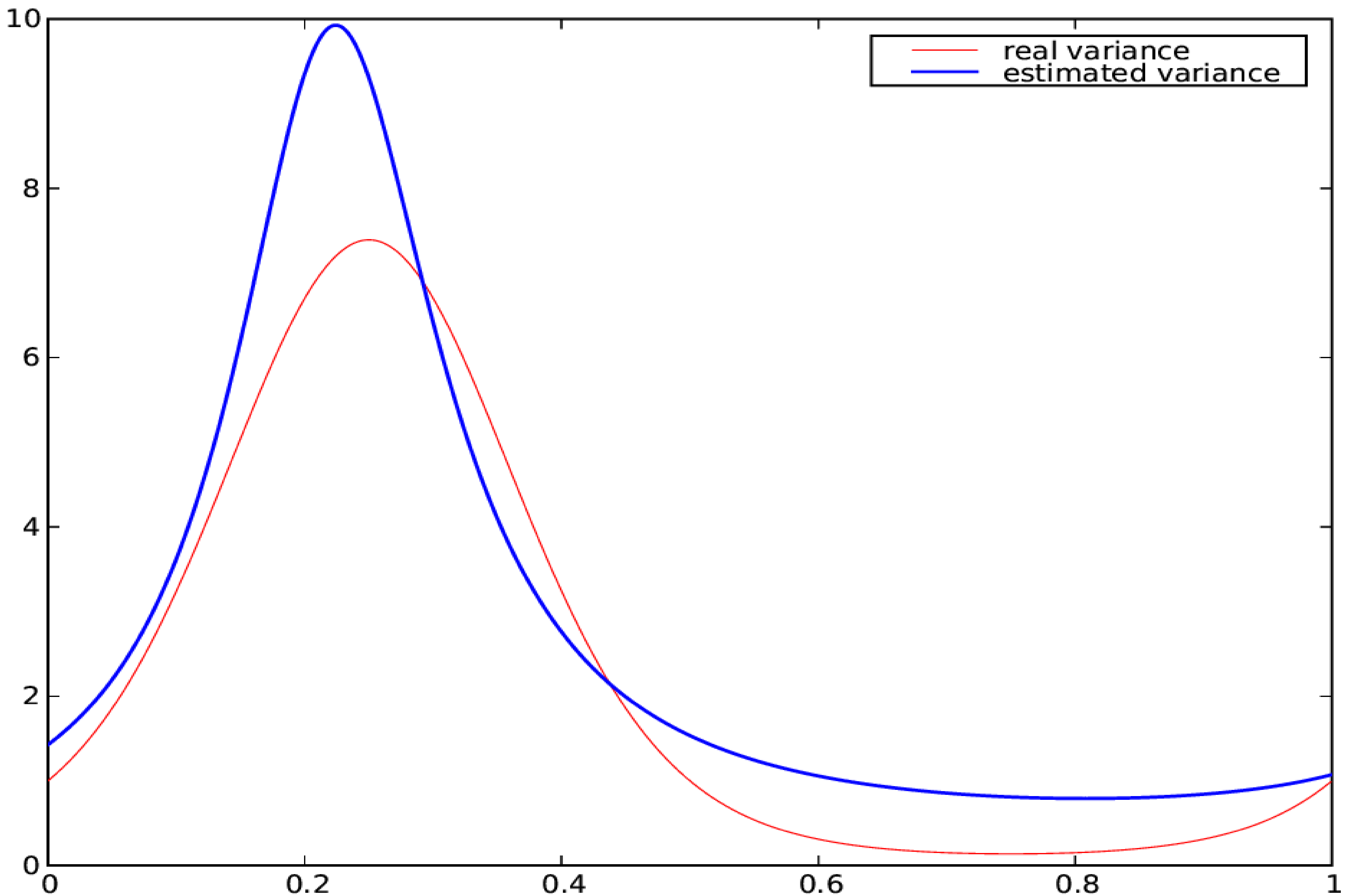
Hence estimate mean and variance **simultaneously**.

Optimization Problem

$$\begin{aligned}
\text{minimize } & \sum_{i=1}^m \left[-\frac{1}{4} \left[\sum_{j=1}^m \alpha_{1j} k_1(x_i, x_j) \right]^\top \left[\sum_{j=1}^m \alpha_{2j} k_2(x_i, x_j) \right]^{-1} \left[\sum_{j=1}^m \alpha_{1j} k_1(x_i, x_j) \right] \right. \\
& \left. -\frac{1}{2} \log\det -2 \left[\sum_{j=1}^m \alpha_{2j} k_2(x_i, x_j) \right] - \sum_{j=1}^m \left[y_i^\top \alpha_{1j} k_1(x_i, x_j) + (y_j^\top \alpha_{2j} y_j) k_2(x_i, x_j) \right] \right] \\
& + \frac{1}{2\sigma^2} \sum_{i,j} \alpha_{1i}^\top \alpha_{1j} k_1(x_i, x_j) + \operatorname{tr} \left[\alpha_{2i} \alpha_{2j}^\top \right] k_2(x_i, x_j). \\
\text{subject to } & 0 \succ \sum_{i=1}^m \alpha_{2i} k(x_i, x_j)
\end{aligned}$$

- The problem is convex
- The log-determinant from the normalization of the Gaussian acts as a **barrrier function**, i.e. a **nice** SDP.


training data
true mean
estimated mean
upper bar
lower bar
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1


real variance
estimated variance
0
2
4
6
8
10
0
0.2
0.4
0.6
0.8
1

Computational Issues

Newton Method with CG Solver

Use Newton method to compute update direction, CG solver instead of inverting Hessian.

Lazy Evaluation

Never build explicit Hessian.

Reduced Rank

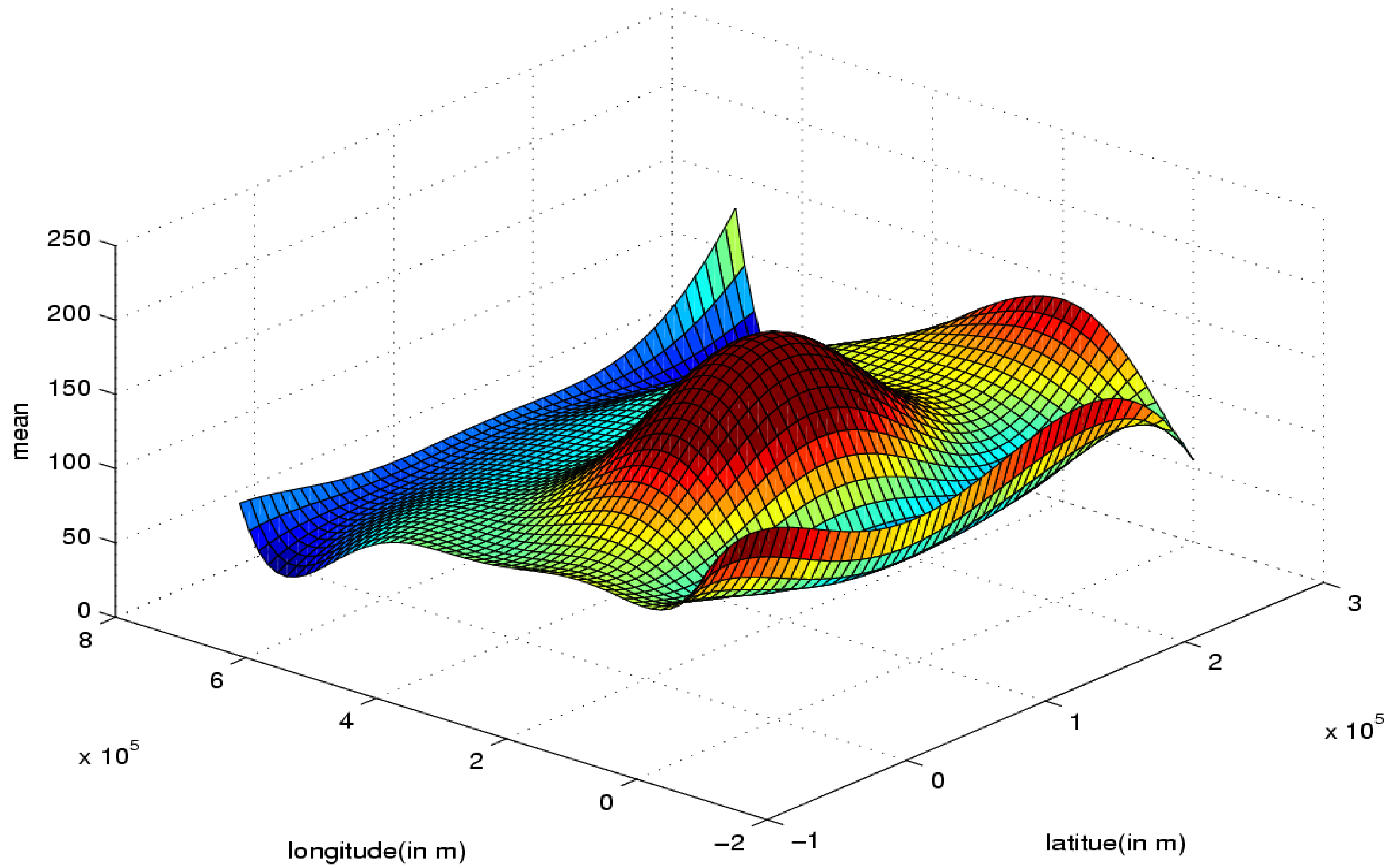
Use incomplete Cholesky factorization for low-rank approximation.

Result

m	100	200	500	1k	2k	5k	10k	20k
Direct Hessian	8	18	90	607	3551	-	-	-
Hessian vector	9	15	38	115	752	-	-	-
Reduced rank	7	7	12	30	54	179	368	727

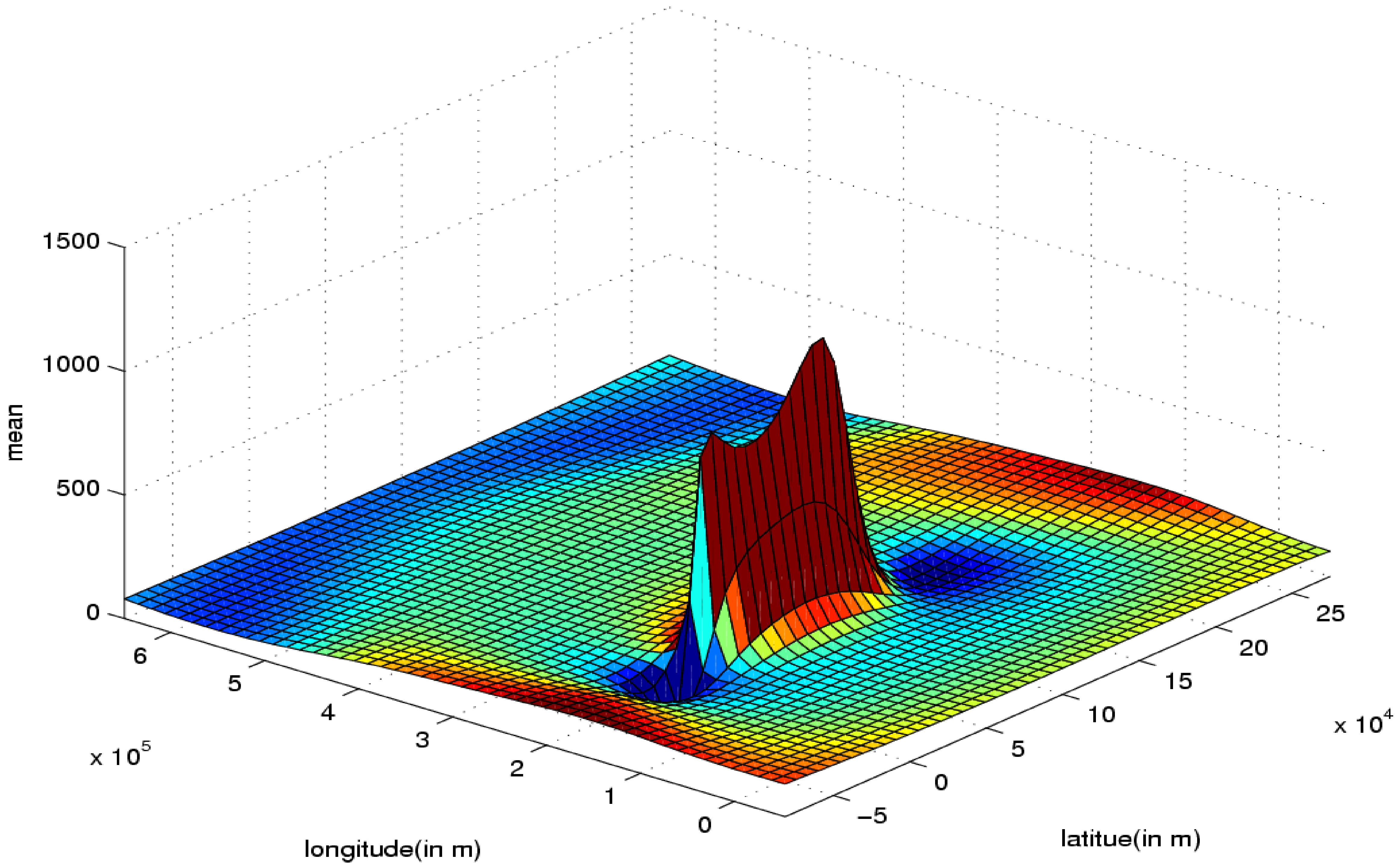
This yields scaling of $O(m^{2.1})$, $O(m^{1.4})$, and $O(m^{0.95})$.

Standard GP

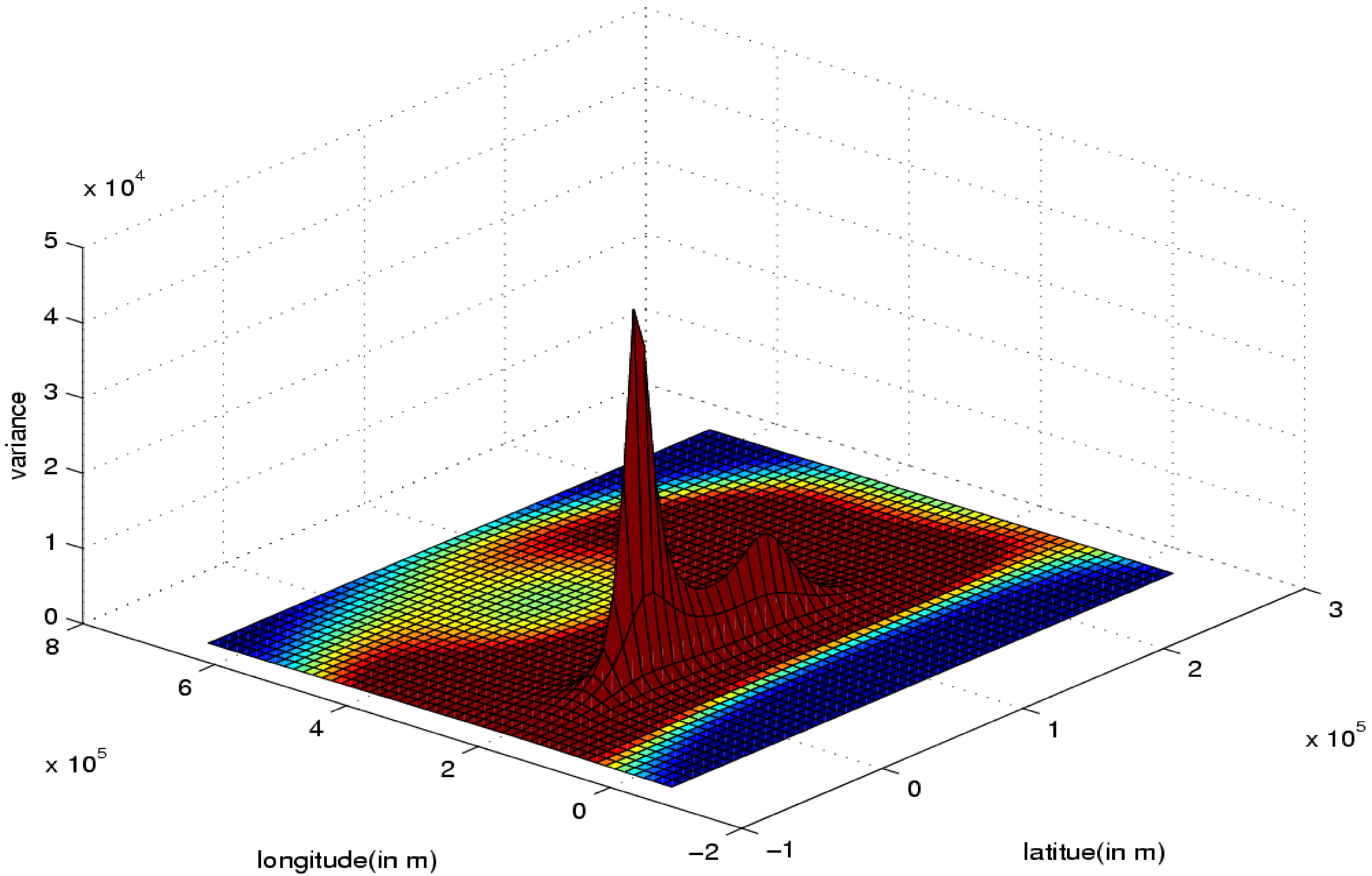


250
200
150
100
50
0
mean
8
6
4
2
0
−2
x 10^5
longitude(in m)
−1
0
1
2
3
x 10^5
latitue(in m)

Heteroscedastic GP mean



Heteroscedastic GP variance



(Generalized) Linear Models

- Kernel trick
  - Simple kernels
  - Kernel PCA
  - Mean Classifier
- Support Vectors
  - Support Vector Machine classification
  - Regression
  - Logistic regression
  - Novelty detection
- Gaussian Process Estimation
  - Regression
  - Classification
  - Heteroscedastic Regression

Further reading

- Ramp loss consistency
  http://books.nips.cc/papers/files/nips24/NIPS2011_1222.pdf
- Ranking and structured estimation
  http://users.cecs.anu.edu.au/~chteo/pub/LeSmoChaTeo09.pdf
- Invariances and convexity
  http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11755
- Ramp loss for structured estimation
  http://users.cecs.anu.edu.au/~chteo/pub/Chaetal09.pdf
- Structured estimation (with margin rescaling)
  http://ttic.uchicago.edu/~altun/pubs/AltHofTso06.pdf
- Structured estimation (without margin rescaling)
  http://www.seas.upenn.edu/~taskar/pubs/icml05.pdf
- Ben Taskar's tutorial
  http://www.seas.upenn.edu/~taskar/nips07tut/nips07tut.ppt

Further reading

- SVM Tutorial (regression)
  http://alex.smola.org/papers/2003/SmoSch03b.pdf
- SVM Tutorial (classification)
  http://www.umiacs.umd.edu/~joseph/support-vector-machines4.pdf
- Introductory chapter of Kernel book
  http://alex.smola.org/teaching/berkeley2012/slides/lwk_chapter1.pdf
- Introductory chapter of structured estimation book
  http://alex.smola.org/teaching/berkeley2012/slides/se_chapter2.pdf
- Kernel PCA
  http://dl.acm.org/citation.cfm?id=295919.295960