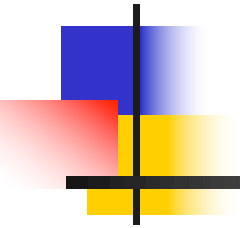
CMSC 510
Regularization Methods for Machine Learning

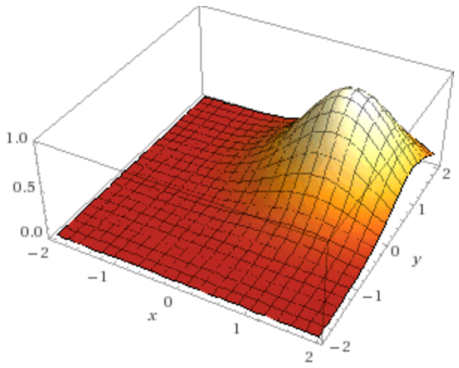


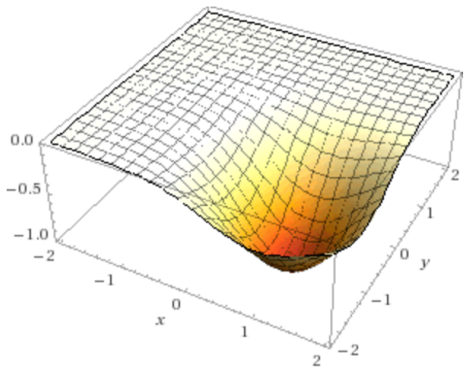
Learning with Kernels

Instructor:
Dr. Tom Arodz

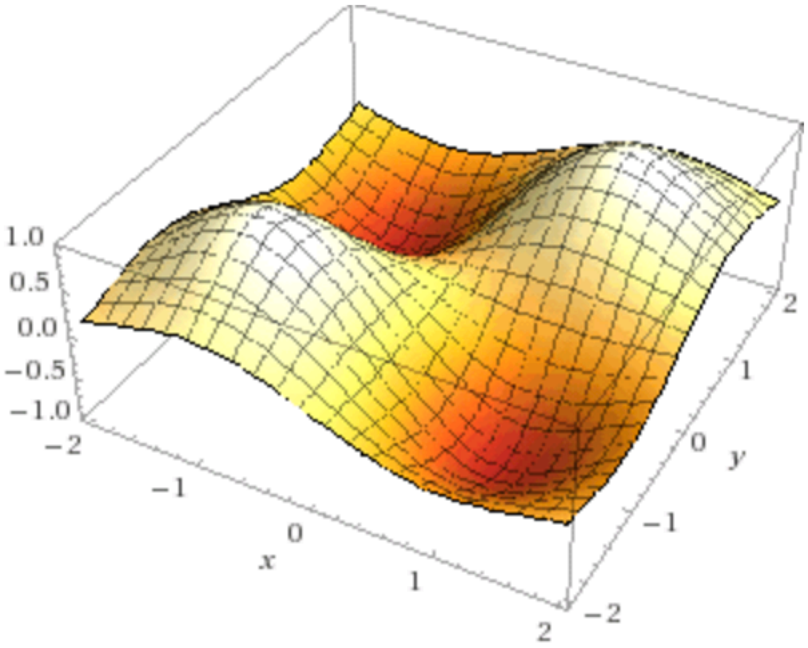
Nonlinear classification

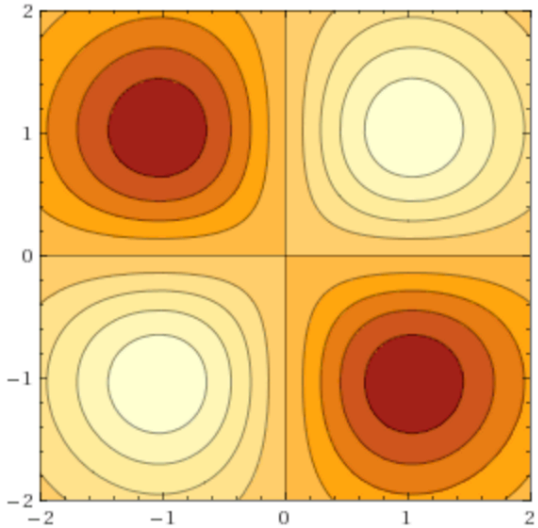
- XOR problem
  - $h(x)=c \exp(-(x-m)^2)$
- Fix 4 Gaussians
  c=+1 or -1
  - Works!





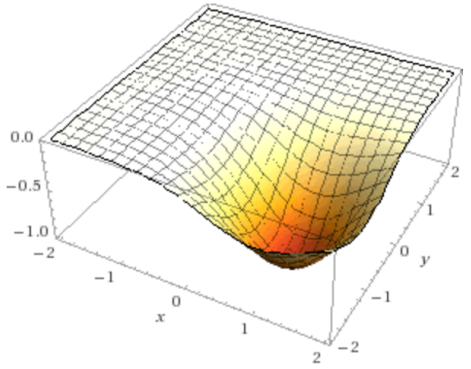
- How can computer find that solution?



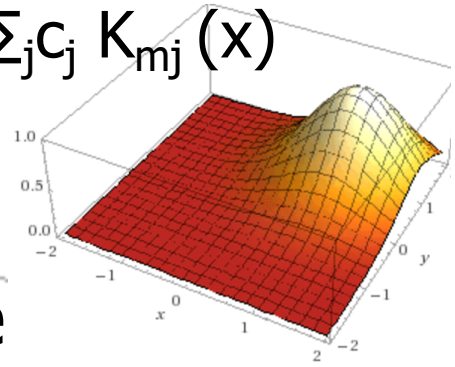


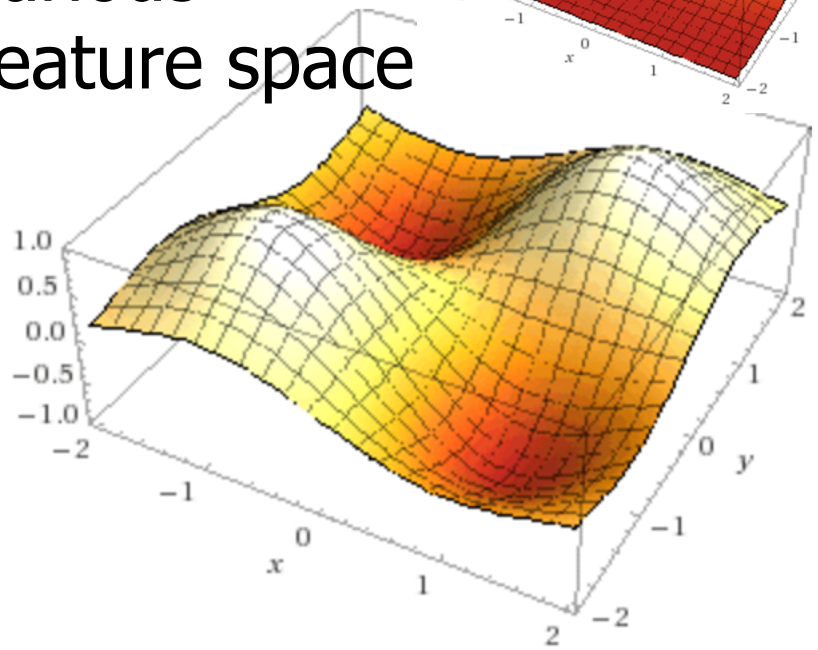
Kernel methods

- $K_{mj}(x) = \exp(-||x-m_j||^2)$



- Classification using Gaussians
  - $h(x)=\sum_j c_j \exp(-||x-m_j||^2) = \sum_j c_j K_{mj}(x)$
- Place a number of Gaussians in various places in the feature space





Key decision:
- Where to place Gaussian centers m_j?
- What c_j to choose?

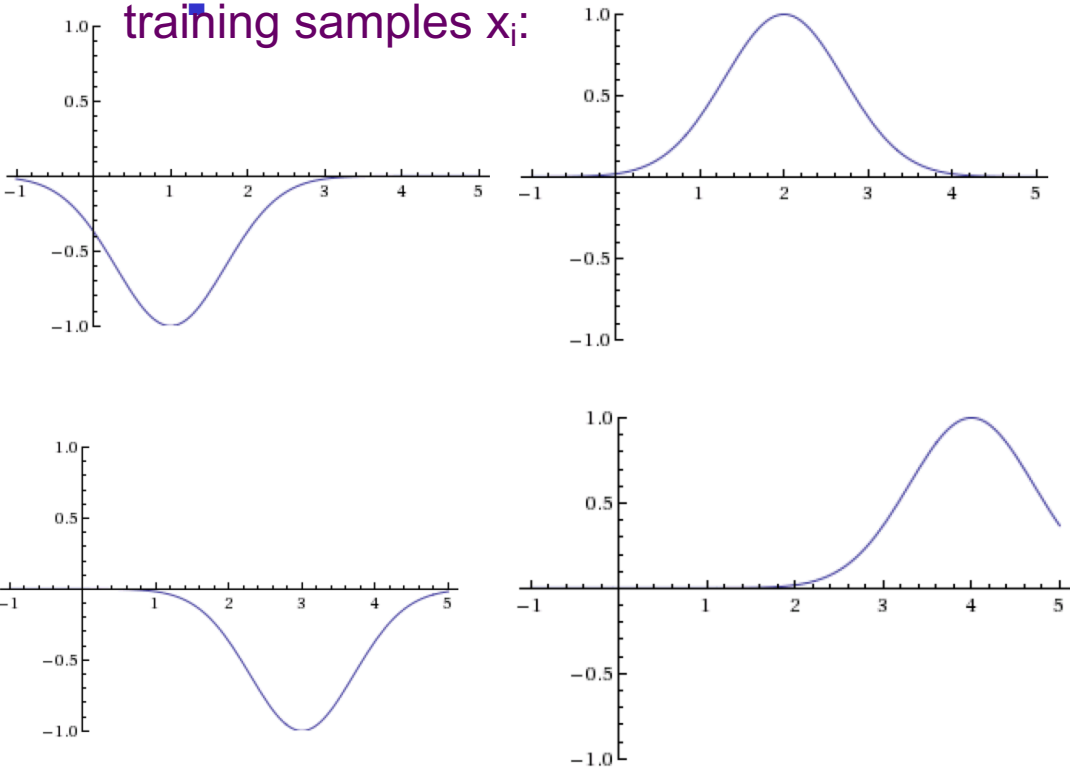
- Space of possible functions:

$$\mathcal{H} := \left\{ h : X \to \mathbb{R} : \quad h(x) = \sum_{i=1}^{n} c_j K_{t_j}(x) = \sum_{i=1}^{n} c_j K(t_j, x) \right\}$$

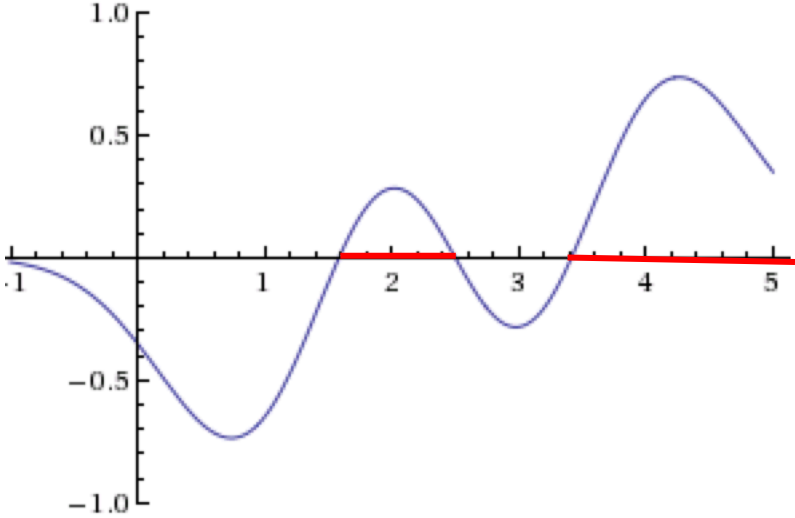
Kernel methods

- Classification using Gaussians
  - $h(x)=\sum_j c_j \exp(-||x-m_j||^2) = \sum_j c_j K_{mj}(x) = \sum_j c_j K(x, m_j)$
  - the function $K(x, m_j)$ used here is called *a kernel*
    - *Gaussian kernel:* $K(x, z) = \exp(-||x - z||^2)$

Four points: $x_1=1$ & $x_3=3$ from class -1; $x_2=2$ & $x_4=4$ from class 1
Individual Gaussians (multiplied by class y_i) centered at (i.e. with means at) training samples x_i:



Possible (not necessarily optimal) decision function $c_1=c_3=-1$ & $c_2=c_4=1$



Gaussian kernel

- Classification with Gaussians:
  $h(x)=\sum_j c_j \exp(-||x-m_j||^2) = \sum_j c_j K_{mj}(x)$
  - Function K(x) is symmetric in this sense:
    - $K_{mj}(x) = \exp(-||x-m_j||^2) = K(m_j,x)$
      $=K(x,m_j) = \exp(-||m_j-x||^2) = K_x(m_j)$

Training:

- Where to place Gaussian centers m_j?
- What c_j to choose?

Naïve approach

- Classification with Gaussians: $h(x)=\sum_j c_j \exp(-||x-m_j||^2) = \sum_j c_j K_{mj}(x)$
  - $K_{mj}(x) = \exp(-||x-m_j||^2) = K(m_j,x)=K(x,m_j) = \exp(-||m_j-x||^2) = K_x(m_j)$

Naïve approach:

- Where to place Gaussian centers m_j?
  - The only special "positions" we know are the positions of samples
  - Let's place Gaussians there, and nowhere else (use sample x_j as m_j)
    - $h(x)=\sum_j c_j \exp(-||x-x_j||^2) = \sum_j c_j K_{xj}(x) = \sum_j c_j K(x_j,x)$

Naïve approach

- Classification with Gaussians: $h(x)=\sum_j c_j \exp(-||x-m_j||^2) = \sum_j c_j K_{mj}(x)$
  - $K_{mj}(x) = \exp(-||x-m_j||^2) = K(m_j,x)=K(x,m_j) = \exp(-||m_j-x||^2) = K_x(m_j)$

Naïve approach:

- What c_j to choose?
  - Just minimize the risk on the training set: $\min_{h\in\mathcal{H}} C\sum_{i=1}^{m} \ell(y_i, h(x_i), b)$
  - If so, we're only evaluating h(x) at training points: $h(x_i)$
    - $h(x_i)=\sum_j c_j \exp(-||x_i-x_j||^2) = \sum_j c_j K_{xj}(x_i) = \sum_j c_j K(x_j,x_i) = \sum_j c_j K[i,j]$
      - We can pre-calculate $K(x_j,x_i)$ and store in a square array $K[i,j] = K(x_j,x_i)$
  - If a point is a "+" sample, no reason to have a -Gaussian there
    - $c_j = a_j y_j$, we expect $a_j >= 0$

Naïve approach

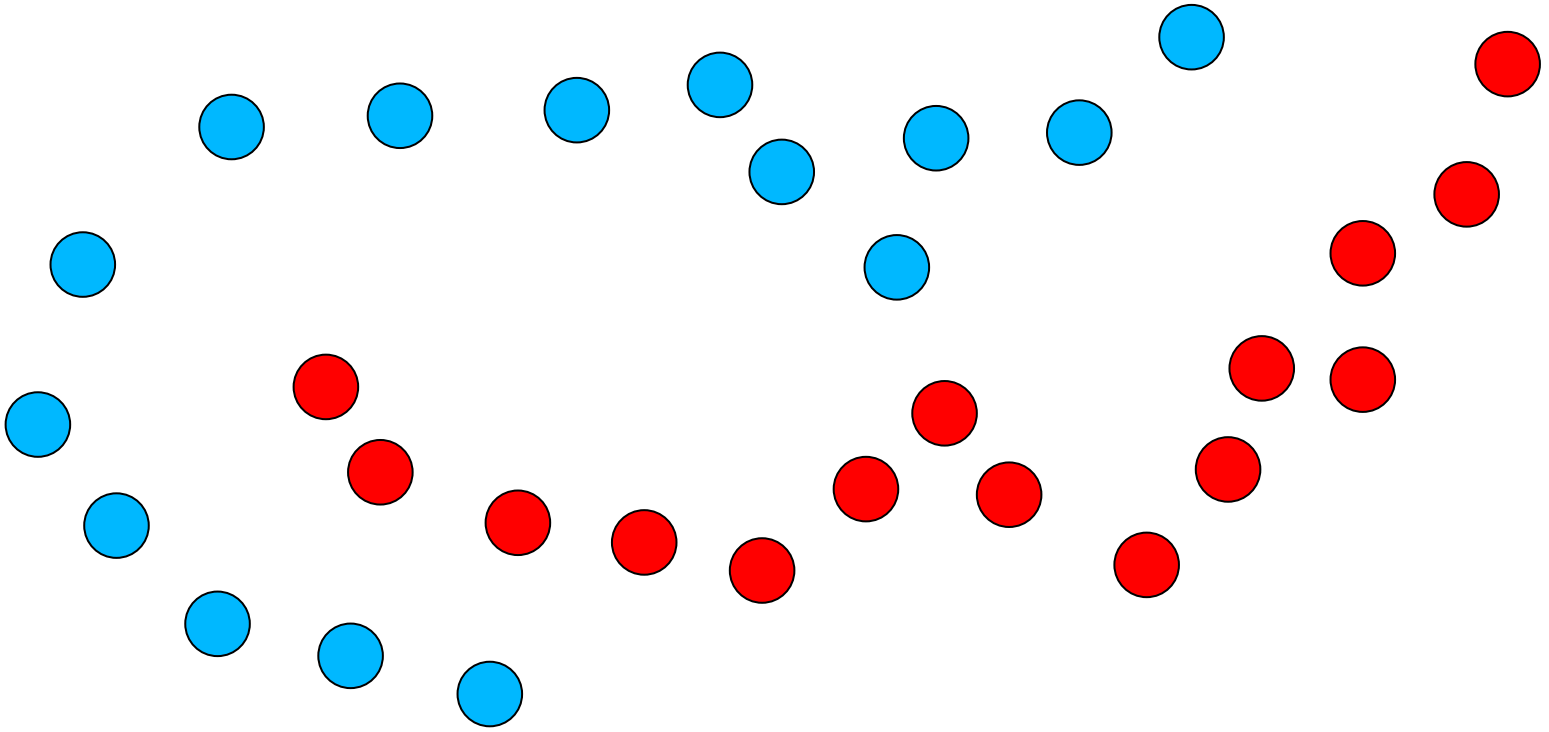
Naïve approach:

- Gaussian centered at training points, with $c_j = a_j y_j$
  - $h(x_i) = \sum_j c_j \exp(-||x_i - x_j||^2) = \sum_j c_j K_{xj}(x_i) = \sum_j c_j K(x_j, x_i) = \sum_j c_j K[i,j]$
- Training:
  - Pre-calculate K[i,j]
  - Train a linear classifier $\sum_j c_j K[i,j]$
    - c_j are like weights w_j
- Use loss ℓ, add bias b, replace $c_j = a_j y_j$ for $a_j >= 0$

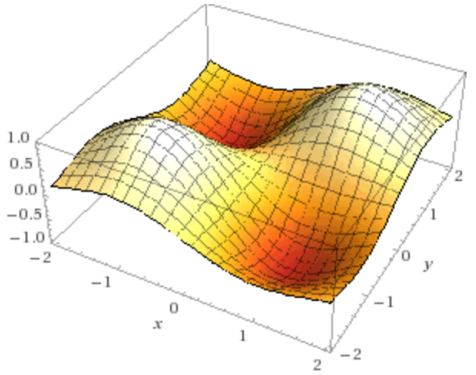
$$\arg\min_{\{\alpha_j\},b} \sum_{i=1}^{m} \ell\left(y_i\left(\sum_{j=1}^{m} \alpha_j y_j K[j,i] + b\right)\right)$$

Naïve approach

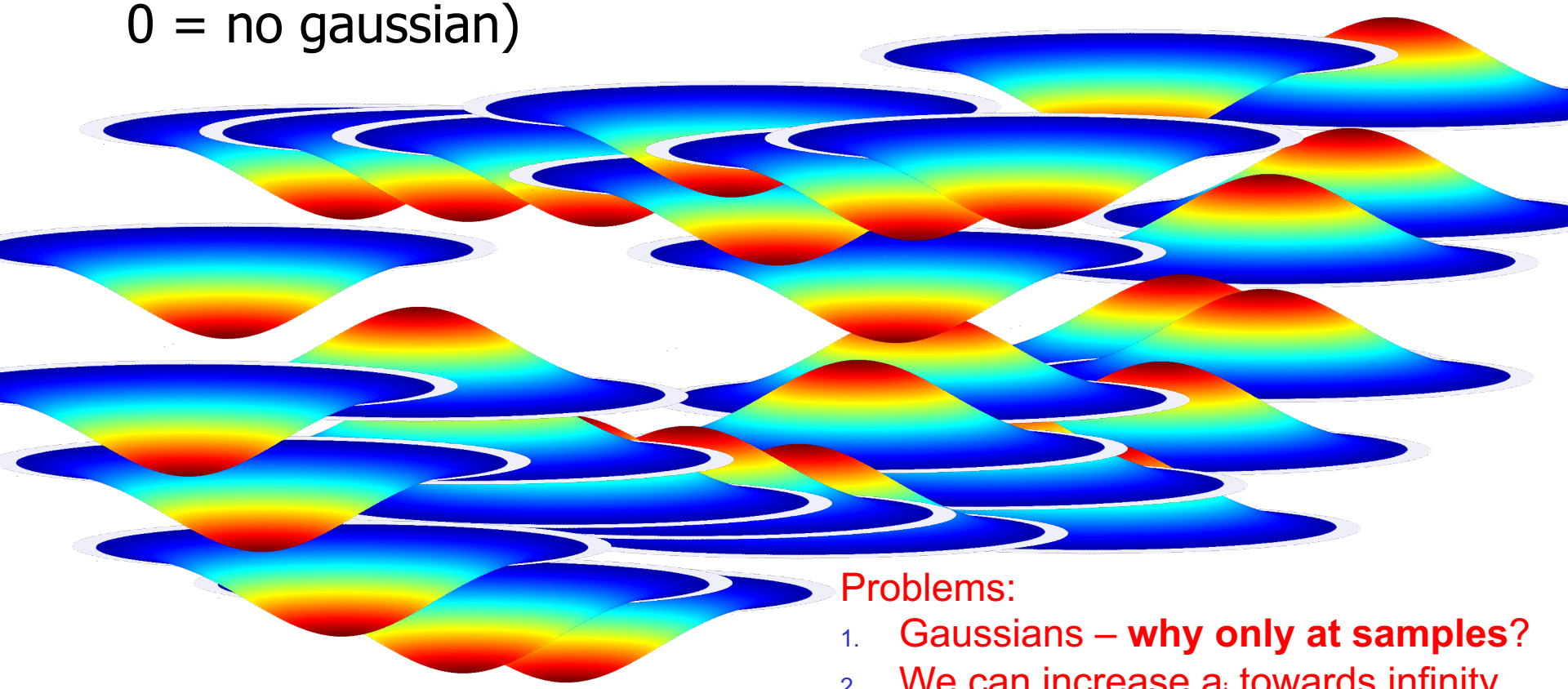
- We have training samples



Naïve approach



- We place a Gaussian at each sample
- And find it's amplitude (positive, negative, or 0 = no gaussian)



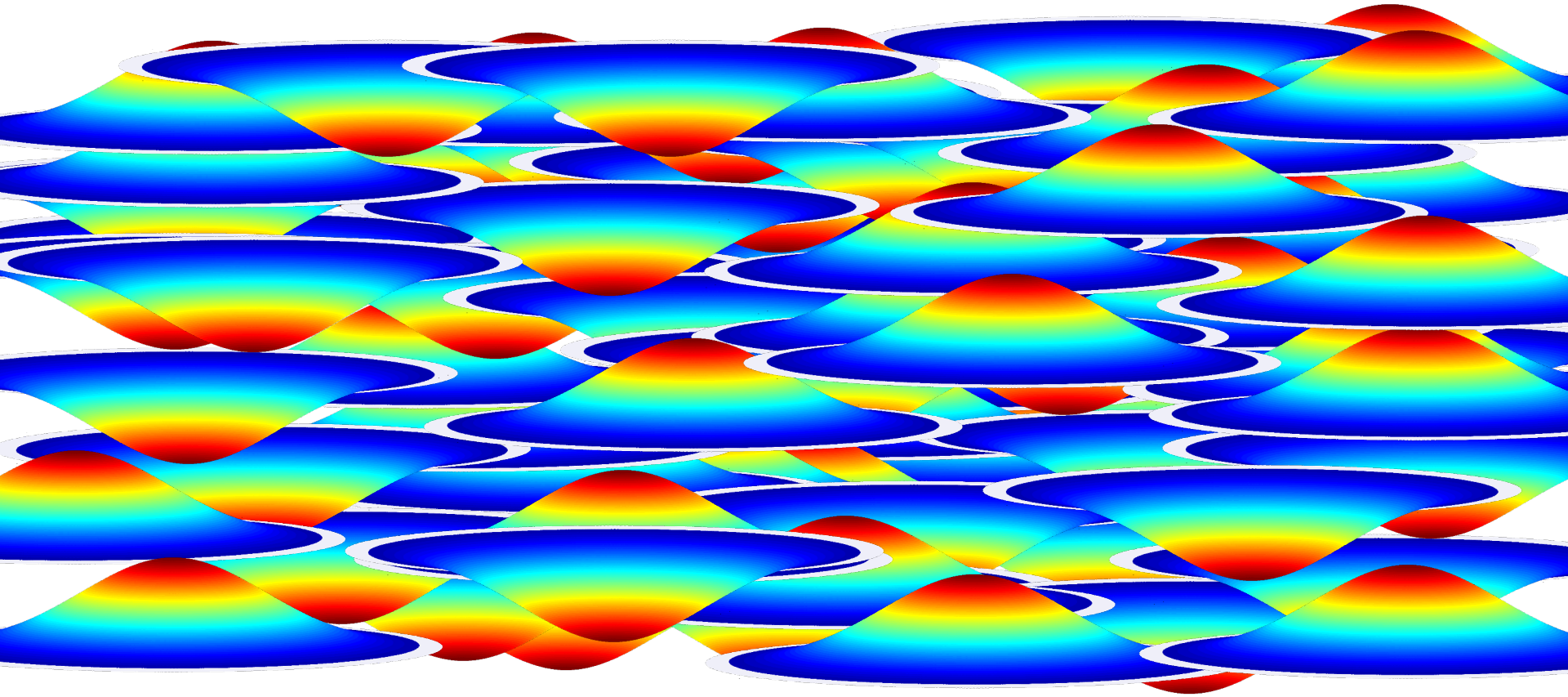
Problems:

1. Gaussians – **why only at samples**?
2. We can increase a_j towards infinity and get higher values of h(x)
   - **Taller Gaussians reduce loss**

$$h(x)=\sum_j a_j\, y_j \exp(-||x-x_j||^2)$$

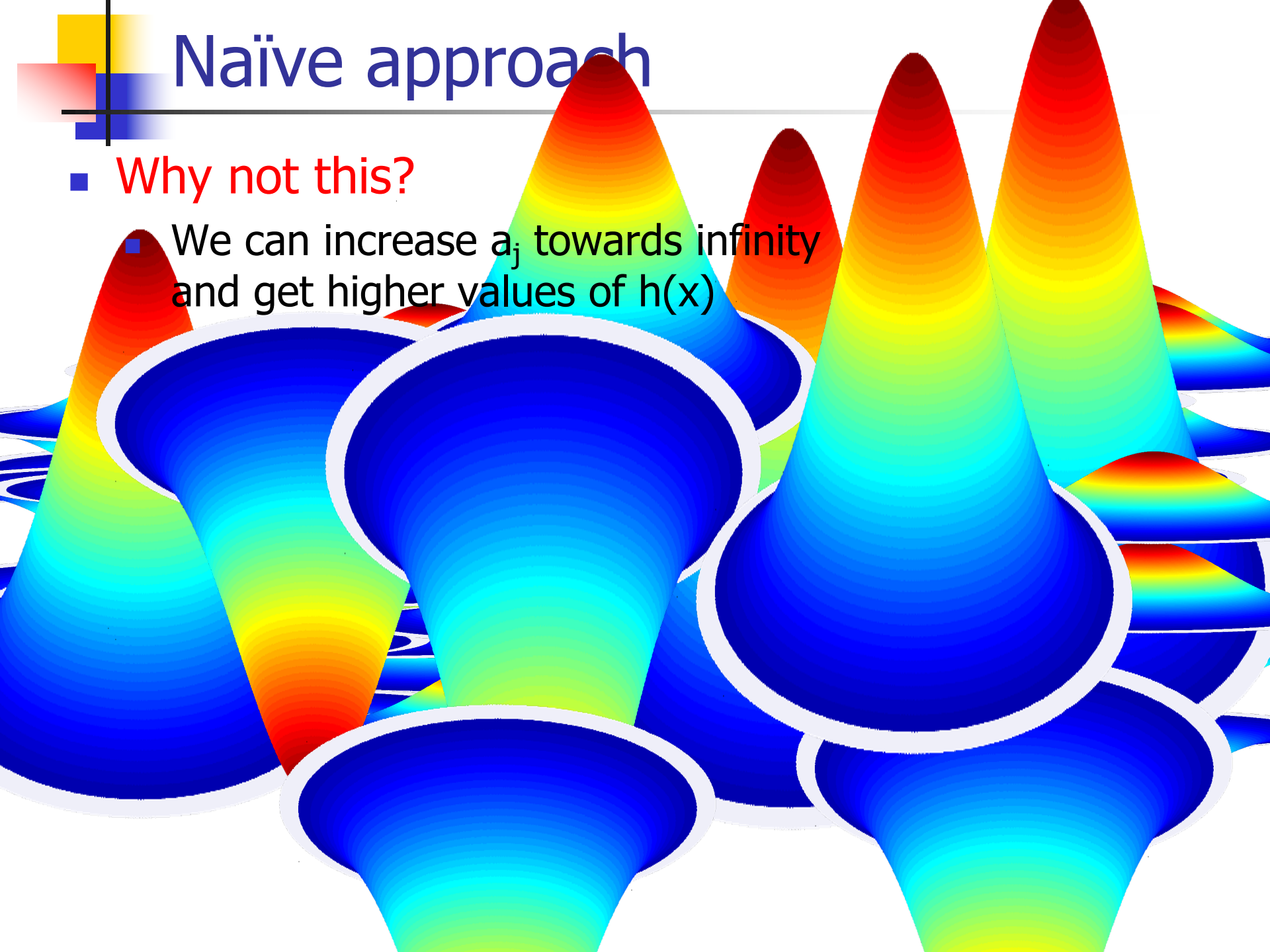
Naïve approach

- Why not this?
  - Gaussians – **why only at samples**?



Naïve approach

- Why not this?
  - We can increase a_j towards infinity and get higher values of h(x)



Better naïve approach

- Classification with Gaussians: $h(x)=\sum_j c_j \exp(-||x-m_j||^2) = \sum_j c_j K_{mj}(x)$
  - $K_{mj}(x) = K(m_j,x) = \exp(-||x-m_j||^2)$

A better naïve approach:

- Where to place Gaussian centers m_j?
  - Let's place Gaussians at samples
    - $h(x)=\sum_j c_j \exp(-||x-x_j||^2) = \sum_j c_j K_{xj}(x) = \sum_j c_j K(x_j,x)$
- What c_j to choose?
  - Minimize the risk on the training set:

$$\arg\min_{\{\alpha_j\},b} \sum_{i=1}^{m} \ell(y_i(\sum_{j=1}^{m} \alpha_j y_j K[j,i] + b))$$

  - But add L_2 (or L_1) penalty on the vector alpha
    - Still not that good: same penalty, no matter where the Gaussian is…

Problems:

- Gaussians – why only at samples?
- We can increase a_j towards infinity (taller Gaussians) and get higher values of h(x)
  - And reduce our loss

Do we need both?