CMSC 510
Regularization Methods for Machine Learning

Reproducing Kernel Hilbert Space

Instructor:
Dr. Tom Arodz

Naïve approach

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

A better naïve approach:

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

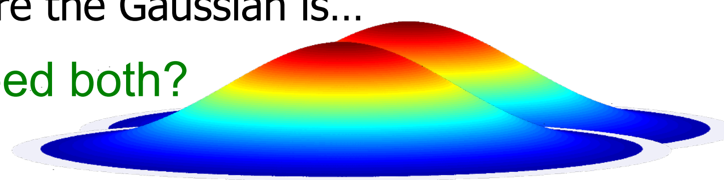
$$\arg\min_{\{\alpha_j\},b} \sum_{i=1}^{m} \ell(y_i(\sum_{j=1}^{m} \alpha_j y_j K[j,i] + b))$$

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

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

Do we need both?



Reproducing Kernel Hilbert Space

- To answer both our problem:
  - Why gaussians only at training points?
  - How to do better regularization?

  we need to define RKHS
- $K_x(z) = \exp(-||z-x||^2)$
- Reproducing Kernel Hilbert Space (for Gaussians)
  - A vector space contains objects (vectors) that we can add and multiply by a real number
    - Gaussians and their linear combinations: we can + and * them
    - Vectors in RKHS: K_x, $K_{x'}$, $0.2*K_x$, $-0.3*K_{x'}$, $7*K_x + 11*K_{x'}$,
  - An inner product between any pair of vectors: $\langle K_x, K_{x'} \rangle$, $\langle 0.2*K_x + .3*K_{x'}, 7*K_x + 11*K_{x'} \rangle = 0.2*7*\langle K_x, K_x \rangle + .3*7*\langle K_{x'}, K_x \rangle + .2*11*\langle K_x, K_{x'} \rangle + .3*11*\langle K_{x'}, K_{x'} \rangle$

Reproducing Kernel Hilbert Space

- Reproducing Kernel Hilbert Space (for Gaussians)
  - A vector space contains objects (vectors) that we can add and multiply by a real number
    - Gaussians and their linear combinations: we can + and * them
    - Vectors in RKHS: K_x , $K_{x'}$, $0.2{*}K_x$, $-0.3{*}K_{x'}$, $7{*}K_x + 11{*}K_{x'}$,
- We also said: $K_x(y) = K(x,y)$ so RKHS is:

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

- With elements like:

$$f(x) = \sum_{j=1}^{n} c_j K_{t_j}(x)$$

$$g(x) = \sum_{j=1}^{n'} c'_j K_{t'_j}(x)$$

Reproducing Kernel Hilbert Space

- Reproducing Kernel Hilbert Space (for Gaussians)
  - An inner product between any pair of vectors: $\langle K_x, K_{x'} \rangle$, $\langle 0.2*K_x + .3*K_{x'}, 7*K_x + 11*K_{x'} \rangle = 0.2*7*\langle K_x, K_x \rangle + .3*7*\langle K_{x'}, K_x \rangle + .2*11*\langle K_x, K_{x'} \rangle + .3*11*\langle K_{x'}, K_{x'} \rangle$
- What should we use as inner product $\langle K_x, K_{x'} \rangle$?
- We will use our kernel: $\langle K_x, K_{x'} \rangle = K(x, x')$

- Then, for: $f(x) = \sum_{j=1}^{n} c_j K_{t_j}(x)$ $\quad g(x) = \sum_{j=1}^{n'} c'_j K_{t'_j}(x)$

we have: $\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{n} \sum_{j=1}^{n'} c_j c'_j K(t_j, t'_j).$

- $K_x(z) = K(x,z) = \exp(-||z-x||^2)$

RKHS - Summary

- For Mercer kernel K(x,z), (e.g. Gaussian) define functions $K_x(z)=K(x,z)$ and construct H as:

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

- Vector space H has inner product **defined as**:

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{n} \sum_{j=1}^{n'} c_j c'_j K(t_j, t'_j).$$

$$f(x) = \sum_{j=1}^{n} c_j K_{t_j}(x)$$

$$g(x) = \sum_{j=1}^{n'} c'_j K_{t'_j}(x)$$

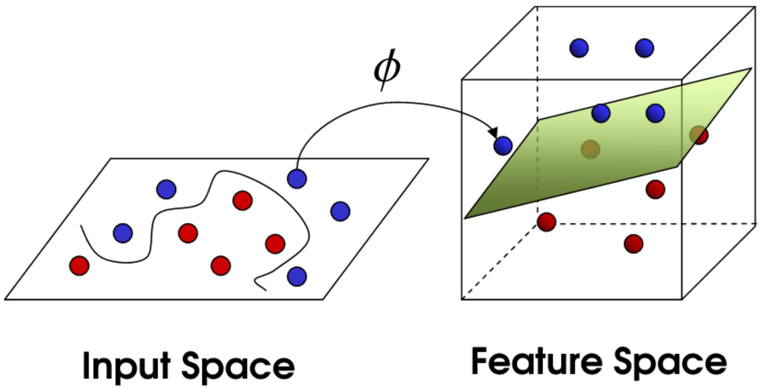
which can be evaluated very easily!

- Gaussian is a valid Mercer kernel:
  - Symmetric
  - Positive semi-definite (matrix K eigenvalues are >=0)

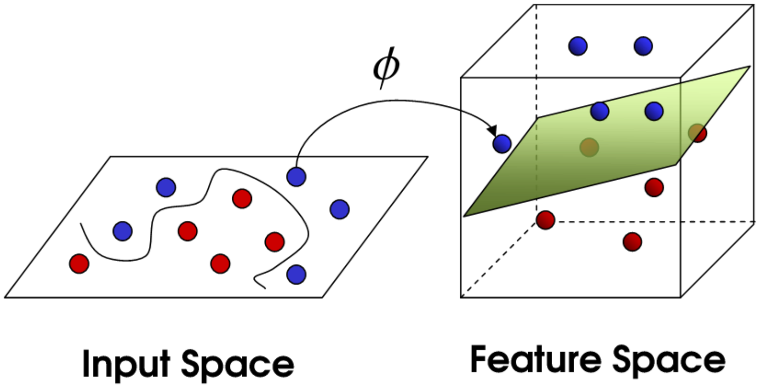
input space => feature space

- We have a classification problem over some samples/vectors x,z, …
  - Vector space with these vectors will now be called "input space", not "feature space"
- Define kernel $K(x,z)=K(z,x)$ e.g. gaussian kernel $K(x,z)=\exp(-||x-z||^2)$
- Define *representers* K_x as functions K_x: $K_x(z)=K(x,z)$
- Define inner product of representers as: $\langle K_x,K_z\rangle_H=K(x,z)$
- We have a valid vector space with representers and their linear cominations in it (the RKHS, aka "feature space" in kernel learning)

- How to connect the two spaces?
  - The input space with x,z etc.
  - The feature space K_x, K_z
- Obvious: define a function $\Phi(x)=K_x$
  - Φ maps x to its representer K_x

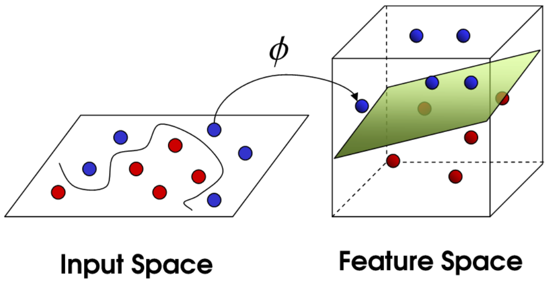


Reproducing Kernel Hilbert Space



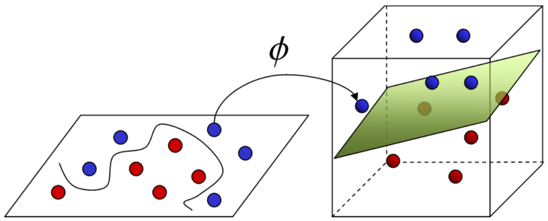
- Let's define a linear classifier in RKHS ("feature space")
- One approach: use the mapping $\Phi(x)=K_x$ to obtain vectors in RKHS
  - $h = \Sigma_j c_j \, \Phi(x_j) = \Sigma_j c_j \, K_{xj}$
  - $h(x) = \Sigma_j c_j \, \Phi(x_j)(x) = \Sigma_j c_j \, K_{xj} \, (x) = \Sigma_j c_j \, K(x_j,x)$ $=\Sigma_j c_j \, \exp(-||x-x_j||^2)$
- Alternative – avoid calculating the mapping Φ, use inner products instead

Reproducing Kernel Hilbert Space



- Alternative – avoid calculating the mapping Φ, use inner products instead
  - What is a linear classifier?
    - We used $h(x)=w^Tx$
    - We can instead write $h(x)=<w,x>$
    - Adding keeps them linear: $h(x)=0.3*<w,x> + 0.2<v,x>$ is also linear
    - What can we use as vector w? any vector of the same dimensionality, including training samples x_1, x_2, …
      - $h(x)=0.3*<x_1,x> + 0.2<x_2,x> + \ldots$ is a linear classifier with $w=0.3x_1+0.2x_2+\ldots$

Reproducing Kernel Hilbert Space



Input Space Feature Space

- Alternative – avoid calculating the mapping Φ, use inner products instead
  - What is a linear classifier?
    - We used $h(x)=w^T x$
    - We can instead write $h(x)=\langle w,x\rangle$
    - Adding keeps them linear: $h(x)=0.3*\langle w,x\rangle + 0.2\langle v,x\rangle$ is also linear
    - What can we use as vector w? any vector of the same dimensionality, including training samples $x_1, x_2, \ldots$
      - $h(x)=0.3*\langle x_1,x\rangle + 0.2\langle x_2,x\rangle + \ldots$ is a linear classifier with $w=0.3x_1+0.2x_2+\ldots$
  - If we have *linear_function*$(x) = \langle x_j,x\rangle$ in the input space, in feature space we have *linear_function*$(\Phi(x))= \langle \Phi(x_j), \Phi(x) \rangle$
  - But $\Phi(x)=K_x$, and also $K(x,z)=K_x(z)$, so we get:
    - $h(x) = \Sigma_j c_j \langle \Phi(x_j), \Phi(x_i) \rangle = \Sigma_j c_j \langle K_{xj},K_x\rangle = \Sigma_j c_j\, K(x_j,x) = \Sigma_j c_j\, K_{xj}(x)$
    - $h(x) = \Sigma_j c_j \exp(-||x_i-x_j||^2)$
    - $h(x)$ is a linear classifier in the feature space, nonlinear in input space

Kernel classifier

- We have an optimization problem where we're looking for the optimal function h(x)

$$\min_{h \in \mathcal{H}} \quad C \sum_{i=1}^{m} \ell(y_i, h(x_i), b) + \frac{1}{2} ||h||_{\mathcal{H}}^2$$

- We use h: $h(x_i) = \sum_{j=1}^{m} c_j K_{x_j}(x_i)$
  where $K_t(x) = \langle K_t, K_x \rangle_H = K(t,x)$ based on kernel K
  - Symmetric, positive definite K $K[i,j] := K(x_i, x_j)$
  - We know how to get norm: from inner product

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{n} \sum_{j=1}^{n'} c_j c'_j K(t_j, t'_j). \qquad ||h||_{\mathcal{H}} = \sqrt{\langle h, h \rangle_{\mathcal{H}}}$$

- We can solve using vectors (we used $c_j = \alpha_j y_j$)

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

Principled (math) approach

- We have an optimization problem where we're looking for the optimal function h(x)

$$\min_{h\in\mathcal{H}} \quad C\sum_{i=1}^{m} \ell(y_i, h(x_i), b) + \frac{1}{2}||h||^2_{\mathcal{H}}$$

$$||h||_{\mathcal{H}} = \sqrt{\langle h, h\rangle_{\mathcal{H}}}$$

$$\langle f, g\rangle_{\mathcal{H}} = \sum_{j=1}\sum_{j=1} c_j c'_j K(t_j, t'_j).$$

- Let's focus on $||h||^2$

$$\min_{h\in\mathcal{H},b} \quad C\sum_{i=1}^{m} \ell(x_i, y_i, h(x_i), b) + \frac{1}{2}||h||^2_{\mathcal{H}}$$

$$= \min_{c\in\mathbb{R}^m,b} C\sum_{i=1}^{m} \ell(x_i, y_i, \sum_{j=1}^{m} c_j K_{x_j}(x_i), b) + \frac{1}{2}||\sum_{j=1}^{m} c_j K_{x_j}||^2_{\mathcal{H}}$$

$$= \min_{c\in\mathbb{R}^m,b} C\sum_{i=1}^{m} \ell(x_i, y_i, \sum_{j=1}^{m} c_j K(x_j, x_i), b) + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j K(x_i, x_j)$$

$K(x_j,x_i) = \exp(-||x_i-x_j||^2)$

High $K(x_i,x_j)$, so:
low c_ic_j highly preferred

Low $K(x_i,x_j)$, so:
not much incentive for low c_ic_j