Kernel Methods for Statistical Learning

Jakob Verbeek

jakob.verbeek@inria.fr

November 4, 2014

http://lear.inrialpes.fr/people/mairal/teaching/2014-2015/MSIAM

Homework + Project

- Homework 1: due Thursday November 6, 2014.
- Homework 2: available on website
- Homework 2 and 3: only required for UJF students
  - Since course has less credits for ENSIMAG students

- Project:
  - Select a paper from the website, or find your own.
  - Due January 5th 2015.
  - Possible to do in teams of 2 students
  - Multiple people/groups can pick the same paper.

Summary of previous lecture

- Definition and properties of reproducing kernel Hilbert spaces.

Reproducing kernel Hilbert spaces

- Let X be a set and H a class of functions over X

$$H \subset R^X$$

forming a Hilbert space with inner product $\langle .,. \rangle_H$

- The function $K: X^2 \rightarrow R$ is called a reproducing kernel of H if

  - H contains all functions of the form

$$\forall x \in X: \; k_x: \; t \rightarrow K(x,t)$$

  - For every x in X, and f in H, the reproducing property holds:

$$f(x) = \langle f, k_x \rangle_H$$

- If a reproducing kernel exists, then H is called a reproducing kernel Hilbert space.

Reproducing kernel Hilbert spaces: example 1

- Let H be the class of linear functions over a real vector space X

$$H=\{f_w \ : \ f_w(x)=w^T x\}$$

forming a Hilbert space with inner product $\langle f_w, f_v \rangle_H = w^T v$

- The function $K: \ X^2 \rightarrow R$ is called a reproducing kernel of H if

  - H contains all functions of the form

$$\forall x \in X: \ K_x: \ t \rightarrow K(x,t)$$

  - For every x in X, and f in H, the reproducing property holds:

$$f(x)=\langle f, K_x \rangle_H$$

- If a reproducing kernel exists, then H is called a reproducing kernel Hilbert space.
  - It this the case ?
  - Yes, for the linear kernel: $K(x,t)=x^T t$

Reproducing kernel Hilbert spaces: example 2

- Let H be the class of all real functions over a finite set X of size n

  $$H = \{ f \in R^n \}$$

  forming a Hilbert space with inner product $\langle f_1, f_2 \rangle_H = \sum_{x \in X} f_1(x) f_2(x)$
- The function $K: \; X^2 \rightarrow R$ is called a reproducing kernel of H if
  - H contains all functions of the form

    $$\forall x \in X: \; K_x: \; t \rightarrow K(x,t)$$
  - For every x in X, and f in H, the reproducing property holds:

    $$f(x) = \langle f, K_x \rangle_H$$

- If a reproducing kernel exists, then H is called a reproducing kernel Hilbert space.
  - It this the case ?
  - Yes, for the identity kernel: $K(x,t) = [x = t]$
    - As before [z]=1 if the expression z is true, zero otherwise.

Reproducing kernel Hilbert spaces: results

- Theorem: If H is an RKHS, then it has a unique reproducing kernel. Conversely, a function K can be the reproducing kernel of at most one RKHS.
- Therefore, we can talk of “the” kernel of an RKHS, and “the” RKHS of a kernel.
- Theorem: A function $K: X^2 \rightarrow R$ is positive definite if and only if it is a reproducing kernel.
- Theorem (Aronszajn,1950): K is a positive definite kernel on the set X if and only if there exists a Hilbert space H and a mapping $\Phi: X \rightarrow H$ such that for any x and x' in X: $K(x,x') = \langle \varphi(x), \varphi(x') \rangle_H$

Proof of Aronzsajn's theorem: first direction

- Suppose K is positive definite over a set X,
  then it is the reproducing kernel of a Hilbert space $H \subset R^X$

- Define the mapping $\Phi: X \to H$ as $\forall x \in X: \varphi(x) = k_x = k(x,.)$

- Then by the reproducing property $f(x) = \langle f, k_x \rangle_H$
  we have:

$$\forall (x,y) \in X^2$$

$$\langle \varphi(x), \varphi(y) \rangle_H = \langle k_x, k_y \rangle_H = k(x,y).$$

Proof of Aronzsajn's theorem: second direction

- Suppose there exists a Hilbert space $H \subset R^X$ and a mapping $\Phi: X \to H$ such that for any x and x' in X: $\langle \varphi(x), \varphi(y) \rangle_H = k(x, y)$.

- Then we have that k is positive definite since

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle_H$$

$$= \sum_{i=1}^{n} \alpha_i \langle \varphi(x_i), \sum_j \alpha_j \varphi(x_j) \rangle_H$$

$$= \langle \sum_{i=1}^{n} \alpha_i \varphi(x_i), \sum_j \alpha_j \varphi(x_j) \rangle_H$$

$$= \| \sum_{i=1}^{n} \alpha_i \varphi(x_i) \|_H \geq 0$$

Plan for this lecture

- **Kernel trick**
  - Distance between points.
  - Distance between sets and points.
  - Data centering.
- Considerations on the RKHS norm
  - Rademacher complexity for RKHS balls.
  - Function smoothness in RKHS.
  - Representer theorem.
- Supervised kernel methods
  - Ridge regression.
  - SVM.
  - Logistic discriminant.
- Fisher kernels

The kernel trick

- Choosing a p.d. kernel K on a set X amounts to embedding the data in a Hilbert space: there exists a Hilbert space H and a mapping
  $\Phi : X \to H$
  such that for all x and x' in X
  $$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{H.}$$

- This mapping might not be explicitly given, nor convenient to work with in practice, e.g. for very large or even infinite dimensions.

- The “trick” is to work implicitly in the feature space H by means of kernel evaluations.

The kernel trick

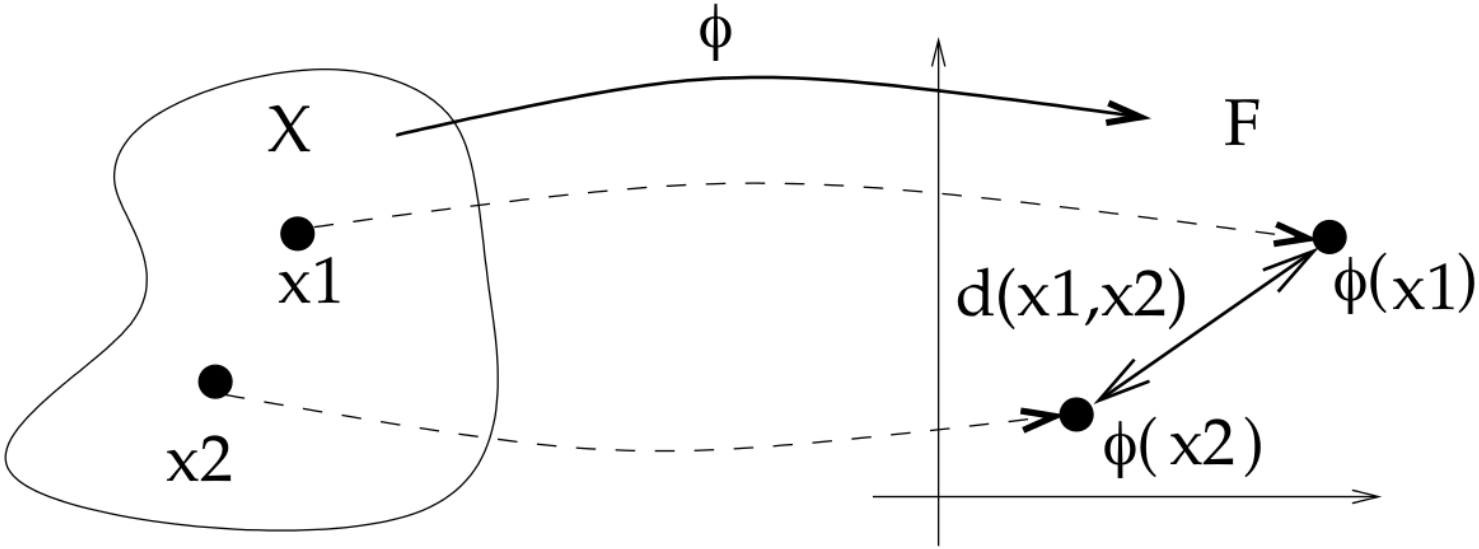
- Any algorithm to process finite dimensional vectors that can be expressed only in terms of pairwise inner products can be applied to potentially infinite-dimensional vectors in the feature space of a p.d. kernel by replacing each inner product evaluation by a kernel evaluation.

- This statement is trivially true, since the kernel computes the inner product in the associated RKHS.

- The practical implications of this “trick” are important.

- Vectors in the feature space are only manipulated implicitly, through pairwise inner products, there is no need to explicitly represent any data in the feature space.

Example 1: computing distances in the feature space



$$
\begin{aligned}
d_k(x,x') &= \|\varphi(x)-\varphi(x')\|_H^2 \\
&= \langle \varphi(x)-\varphi(x'), \varphi(x)-\varphi(x') \rangle_H \\
&= \langle \varphi(x), \varphi(x) \rangle_H + \langle \varphi(x'), \varphi(x') \rangle_H - 2\langle \varphi(x), \varphi(x') \rangle_H \\
&= k(x,x) + k(x',x') - 2k(x,x')
\end{aligned}
$$

Distance for the Gaussian kernel

- The Gaussian kernel with bandwidth sigma is given by

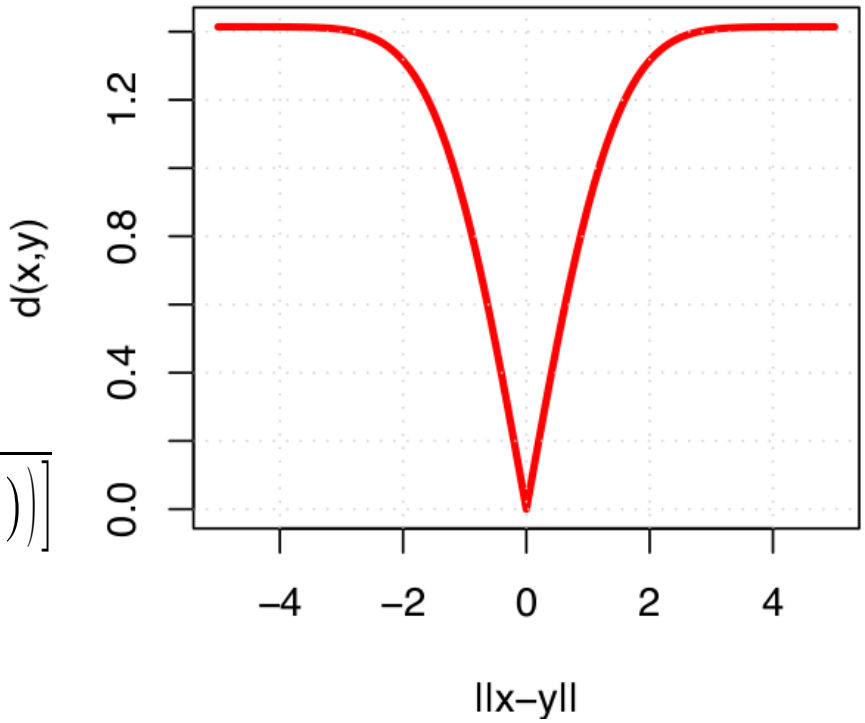
$$k(x,x') = \exp\left(-\|x-x'\|_2 / (2\sigma^2)\right)$$

- In the feature space, all points are embedded on the unit sphere since

$$k(x,x) = \|\varphi(x)\|_H^2 = 1$$

- The distance in the feature space between x and x' is given by

$$d_k(x,x') = \sqrt{2\left[1-\exp\left(-\|x-x'\|^2/(2\sigma^2)\right)\right]}$$



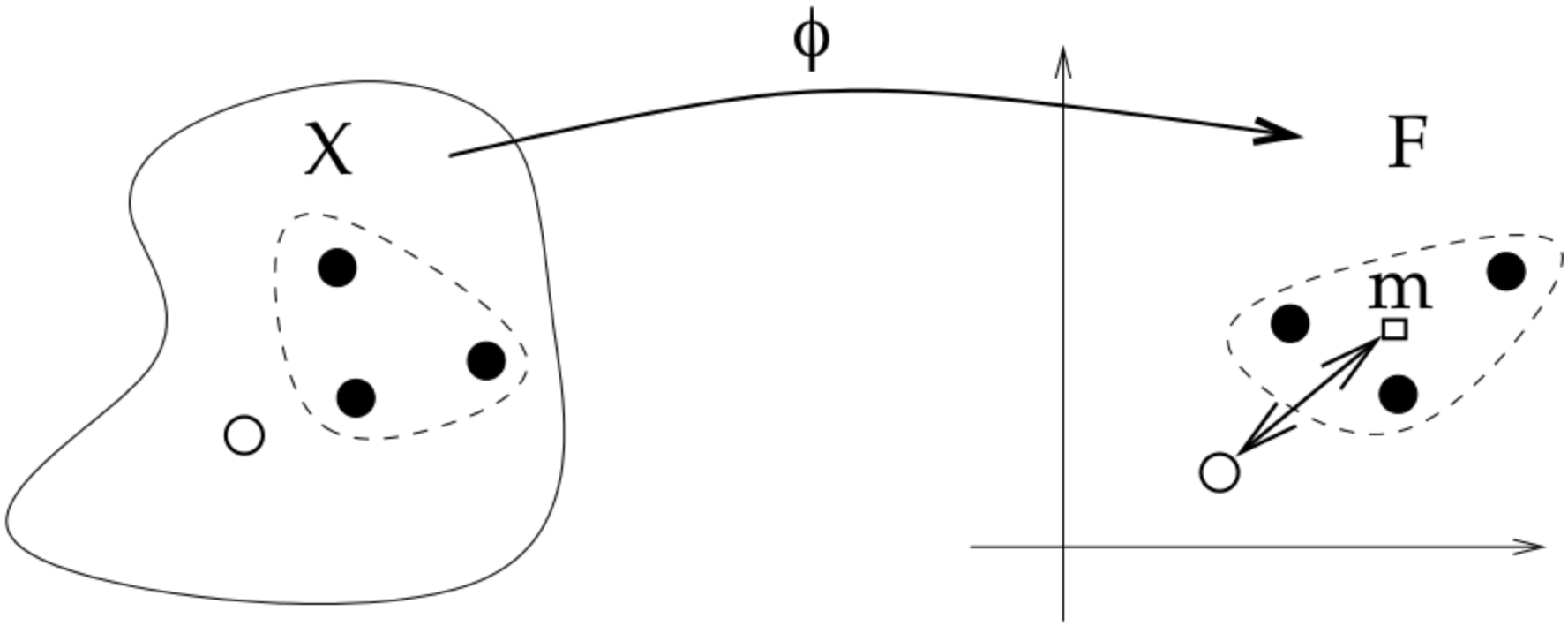
Example 2: distance between a point and a set

- Let S be a finite set of points in X: $S=(x_{1,}\ldots,x_n)$
- How to define and compute the similarity between any point x in X and the set S?

- The following is a simple approach:
  - Map all points to the feature space
  - Summarize S by the barycenter of the points $m=\frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)$
  - Define the distance between x and S as

$$d_k(x,S)=\|\varphi(x)-m\|_H$$

Example 2: distance between a point and a set



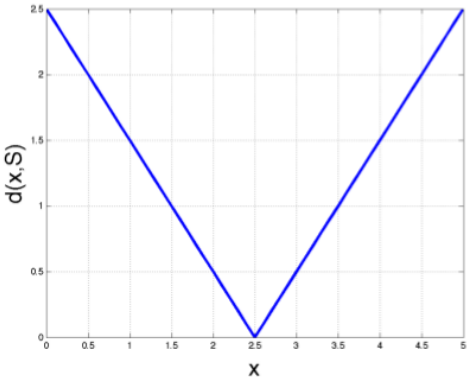
$$d_k(x,S) = \|\varphi(x) - m\|_H$$

$$= \|\varphi(x) - \frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)\|_H$$

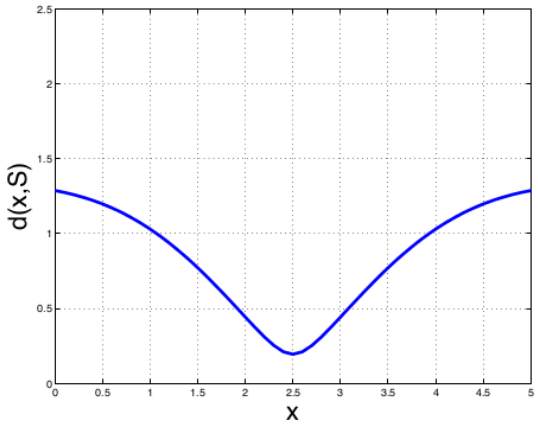
$$= \sqrt{k(x,x) - \frac{2}{n}\sum_{i=1}^{n} k(x,x_i) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i,x_j)}$$

Uni-dimensional illustration

- Let S = {2,3}, plot f(x) = d(x,S).

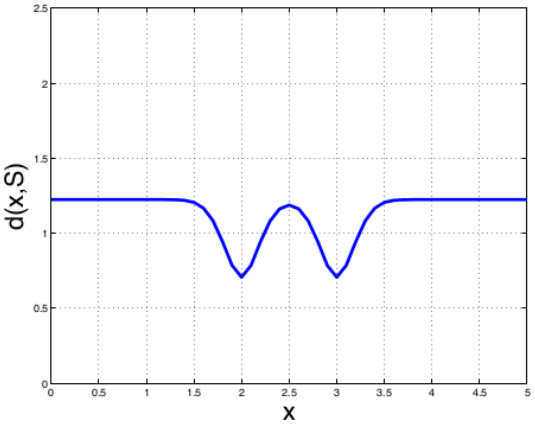


Linear kernel



Gaussian kernel,

with $\sigma = 1$

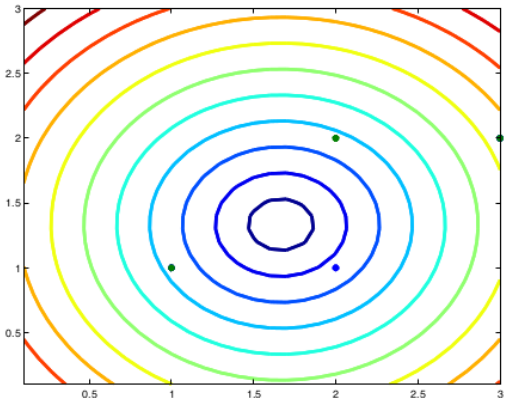


Gaussian kernel,

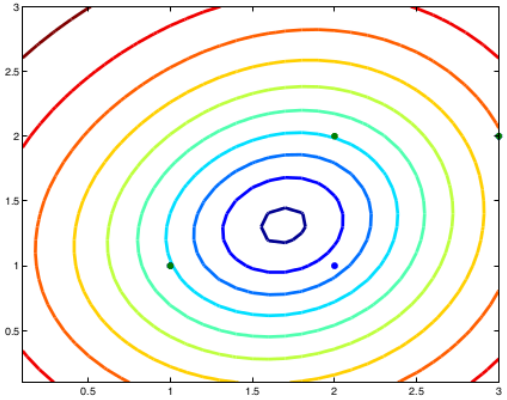
with $\sigma = 0.2$

2D illustration

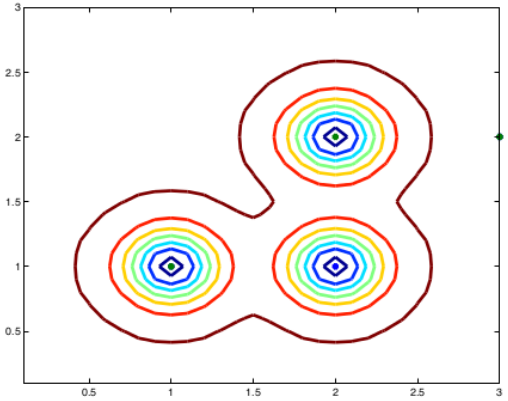
- Let S = { (1,1)', (1,2)', (2,2)' }, plot f(x) = d(x,S).



Linear kernel



Gaussian kernel, with $\sigma = 1$



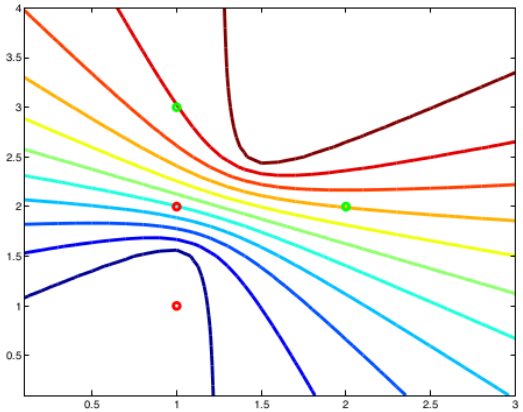
Gaussian kernel, with $\sigma = 0.2$

Application to discrimination

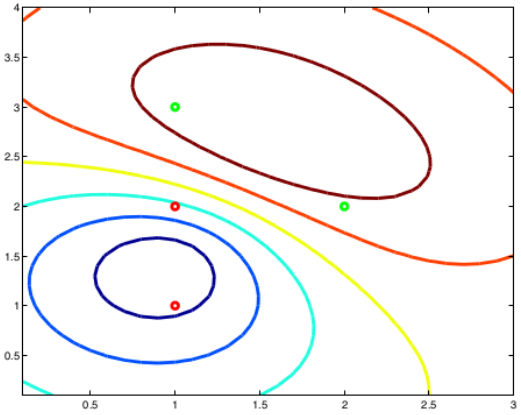
- Consider a set of points from positive class P = { (1,1)', (1,2)' }
- And a set of points from the negative class N={ (1,3)', (2,2)' }
- Plot $f(x) = d_k(x,P)^2 - d_k(x,N)^2$

$$= \|\varphi(x) - m_P\|_H^2 - \|\varphi(x) - m_N\|_H^2$$

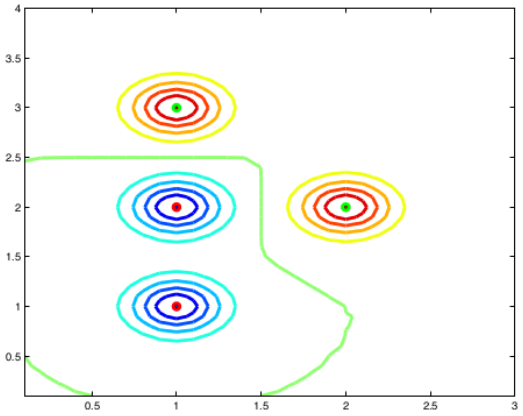
$$= \frac{2}{n}\sum_{x_i \in N} k(x, x_i) - \frac{2}{n}\sum_{x_i \in P} k(x, x_i) + \text{constant}$$



Linear kernel



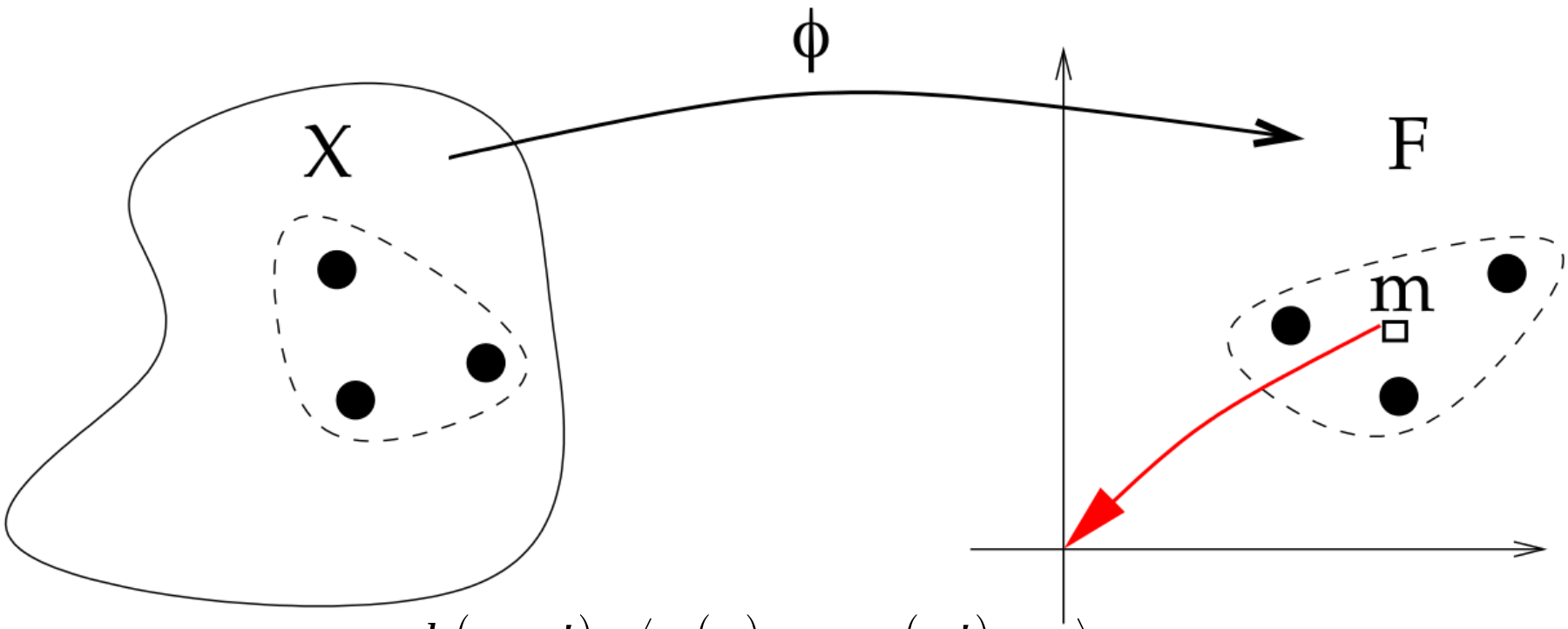
Gaussian kernel, with $\sigma = 1$



Gaussian kernel, with $\sigma = 0.2$

Example 3: centering data in feature space

- Let S be a set of n points in X.
- Let K be the kernel matrix generated by the p.d. kernel k(.,.).
- Let m be the barycenter in the feature space of the points in S.
- How to compute the kernel matrix when the points are centered on m?



$$h(x,x') = \langle \varphi(x) - m, \varphi(x') - m \rangle_H$$

Example 3: centering data in feature space

- Substitution of the barycenter gives

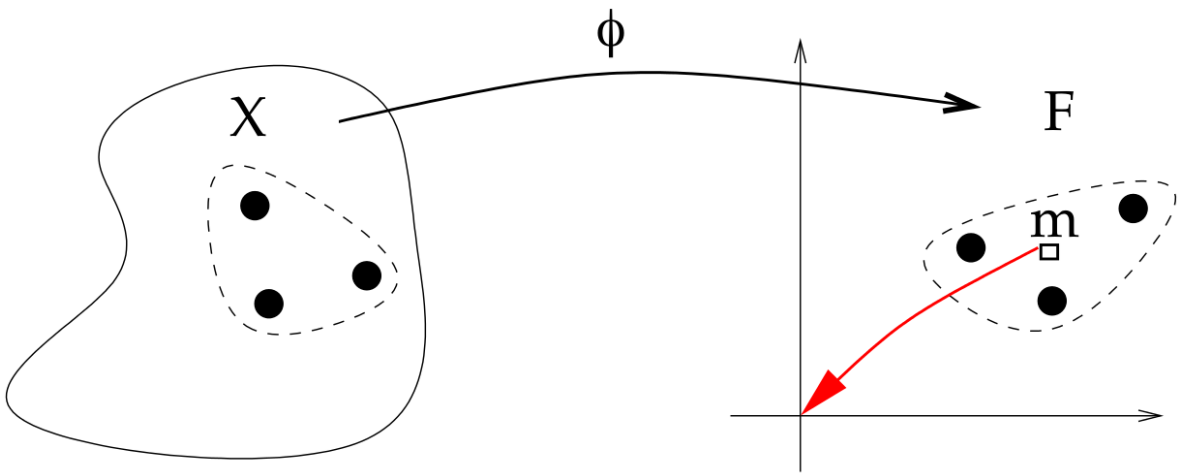
$$h(x_i,x_j)=\langle \varphi(x_i)-m,\varphi(x_j)-m\rangle_H$$
$$=\langle \varphi(x_i),\varphi(x_j)\rangle_H-\langle m,\varphi(x_i)+\varphi(x_j)\rangle_H+\langle m,m\rangle_H$$
$$=k(x_i,x_j)-\frac{1}{n}\sum_{k=1}^{n}\left(k(x_i,x_k)+k(x_k,x_j)\right)+\frac{1}{n^2}\sum_{k,l=1}^{n}k(x_k,x_l)$$

- Or, in matrix notation we get

$$H=K-KU-UK+UKU=(I-U)K(I-U)$$

where for all i,j:

$$U_{i,j}=1/n$$



Plan for this lecture

- Kernel trick
  - Distance between points.
  - Distance between sets and points.
  - Data centering.
- **Considerations on the RKHS norm**
  - Rademacher complexity for RKHS balls.
  - Function smoothness in RKHS.
  - Representer theorem.
- Supervised kernel methods
  - Ridge regression.
  - SVM.
  - Logistic discriminant.
- Fisher kernels

Recap of Rademacher complexity

- Definition of Rademacher complexity of a function class H over X
  - Let $\sigma_i,\ \ i=1,\ldots,n$ be i.i.d. variables with $p(\sigma_i=+1)=p(\sigma_i=-1)=1/2$
  - Let $x_i \in X,\ \ i=1,\ldots,n$ be i.i.d. variables

$$Rad_n(H)=\boldsymbol{E}_{X,\sigma}\left[\sup_{f\in F_B}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right]$$

- Intuitively measures how well functions in H can align to noise.
- Rademacher complexity bounds the expected estimation error in the expected risk

$$\boldsymbol{E}_{(x_i,y_i)_{1=1,\ldots,n}}\left[R(\hat{f})-R^*\right]=\left(min_{g\in H}R(g)-R^*\right)+\left(R(\hat{f})-min_{g\in H}R(g)\right)$$
$$\leq\left(min_{g\in H}R(g)-R^*\right)+4\,Rad_n(H)$$

  - Average taken over training sets that generate our estimator

Rademacher Complexity in RKHS balls

- Suppose X is endowed with a positive definite kernel k, and associated RKHS H.
- Consider the class of functions f in H in a ball of radius B in H.

$$F_B = \{ f \in H \ : \ \|f\|_H \leq B \}$$

- Rademacher complexity of this class can be upper bounded as

$$Rad_n(F_B) \leq \frac{2B\sqrt{\boldsymbol{E}\, k(x,x)}}{\sqrt{n}}$$

- Therefore, by plugging this into the general Rademacher risk bound, the Bayes regret of an estimator in this class can be bounded by

$$\boldsymbol{E}_{(x_i,y_i),i=1,\ldots,n}\left[R(\hat{f}) - R^*\right] \leq \left(min_{g \in F_B} R(g) - R^*\right) + 4\frac{2B\sqrt{\boldsymbol{E}\, k(x,x)}}{\sqrt{n}}$$

  - First term is the bias term that decreases with B
  - Second term is variance term that increases (linearly) with B.

Rademacher Complexity in RKHS balls, proof (1/2)

$$Rad_n(F_B) = \boldsymbol{E}_{X,\sigma}\left[\sup_{f\in F_B}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right]$$

$$= \boldsymbol{E}_{X,\sigma}\left[\sup_{f\in F_B}\left|\left\langle f, \frac{2}{n}\sum_{i=1}^{n}\sigma_i k_{x_i}\right\rangle\right|\right] \qquad \text{By RKHS}$$

$$\leq \boldsymbol{E}_{X,\sigma}\left[\sup_{f\in F_B}\|f\|\times\left\|\frac{2}{n}\sum_{i=1}^{n}\sigma_i k_{x_i}\right\|\right] \qquad \text{By Cauchy-Schwarz}$$

$$= \boldsymbol{E}_{X,\sigma}\left[B\left\|\frac{2}{n}\sum_{i=1}^{n}\sigma_i k_{x_i}\right\|\right]$$

$$= \frac{2B}{n}\boldsymbol{E}_{X,\sigma}\left[\sqrt{\left\|\sum_{i=1}^{n}\sigma_i k_{x_i}\right\|_H^2}\right]$$

$$\leq \frac{2B}{n}\sqrt{\boldsymbol{E}_{X,\sigma}\left[\left\|\sum_{i=1}^{n}\sigma_i k_{x_i}\right\|_H^2\right]} \qquad \text{By Jensen's inquality}$$

$$= \frac{2B}{n}\sqrt{\boldsymbol{E}_{X,\sigma}\left[\sum_{i,j=1}^{n}\sigma_i\sigma_j k(x_i,x_j)\right]}$$

Rademacher Complexity in RKHS balls, proof (2/2)

- But for i.i.d. and uniform $\sigma_i \in \{-1, +1\}$

  we have that $\boldsymbol{E}[\sigma_i \sigma_j]$ is one if i=j and zero otherwise. Therefore:

$$Rad_n(F_B) \leq \frac{2B}{n} \sqrt{\boldsymbol{E}_{X,\sigma}\left[\sum_{i,j=1}^{n} \sigma_i \sigma_j k(x_i, x_j)\right]}$$

$$= \frac{2B}{n} \sqrt{\boldsymbol{E}_{X}\left[\sum_{i,j=1}^{n} \boldsymbol{E}_{\sigma}[\sigma_i \sigma_j] k(x_i, x_j)\right]}$$

$$= \frac{2B}{n} \sqrt{\boldsymbol{E}_{X}\left[\sum_{i=1}^{n} k(x_i, x_i)\right]}$$

$$= \frac{2B \sqrt{\boldsymbol{E}_{X}[k(x, x)]}}{\sqrt{n}}$$

Rademacher Complexity in RKHS balls

- Consider the class of functions f in H in a ball of radius B in H.

$$F_B = \{ f \in H \ : \ \|f\|_H \leq B \}$$

- The Rademacher complexity of this class can be upper bounded as

$$Rad_n(F_B) \leq \frac{2B\sqrt{\boldsymbol{E}\, k(x,x)}}{\sqrt{n}}$$

- Therefore, by plugging this into the general Rademacher risk bound, the Bayes regret of an estimator in this class can be bounded by

$$\boldsymbol{E}_{(x_i,y_i),i=1,\ldots,n}\left[R(\hat{f}) - R^*\right] \leq \left(min_{g \in F_B} R(g) - R^*\right) + 4\frac{2B\sqrt{\boldsymbol{E}\, k(x,x)}}{\sqrt{n}}$$

- For different choices of B find estimator by minimizing empirical risk

$$\hat{f} = \text{argmin}_{f \in F_B} \ \frac{1}{n}\sum_{i=1}^{n} L(y, f(x_i))$$

  - Or equivalently for different lambda

$$\hat{f} = \text{argmin}_{f \in H} \ \lambda \frac{1}{2}\|f\|_H^2 + \frac{1}{n}\sum_{i=1}^{n} L(y, f(x_i))$$

Smoothness of functions in RKHS

- Let f be a function in a RKHS H with associated kernel k over X
- Consider the difference in function evaluations for two points x and x' in X

$$\begin{aligned} |f(x)-f(x')| &= |\langle f, \varphi(x)\rangle - \langle f, \varphi(x')\rangle| \\ &= |\langle f, \varphi(x)-\varphi(x')\rangle| \\ &\leq \|f\|_H \times \|\varphi(x)-\varphi(x')\|_H \end{aligned}$$

- The RKHS norm of f gives the Lipschitz constant of f, for the metric

$$d_k(x,x') = \|\varphi(x)-\varphi(x')\|_H$$

- In particular for $f = \sum_{i=1}^{n} \alpha_i k(x_i,.)$ we have

$$\|f\|_H^2 = \left\langle \sum_{i=1}^{n} \alpha_i k(x_i,.), \sum_{i=1}^{n} \alpha_i k(x_i,.) \right\rangle_H = \sum_{i,j} \alpha_i \alpha_j k(x_i,x_j) = \alpha^T K \alpha$$

Representer Theorem

- Let k be a positive definite kernel over X, and let H be the RKHS associated with k. Let
  - $S=\{x_1,\ldots,x_n\}$ with $x_1,\ldots,x_n\in X$
  - $\Psi:\ R^{n+1}$ → R Be a function that is strictly increasing in its last variable

- Then the solution f* of the problem

$$min_{f\in H}\ \Psi\left(f(x_1),\ldots,f(x_n),\|f\|_H\right)$$

has the form

$$f^*=\sum_{i=1}^{n}\alpha_i k(x_i,.)$$

Proof Representer Theorem (1/2)

- Let H^S be the subspace of H spanned by $k(.,x_i), x_i \in S$

$$H^S = \{ f \in H \ : \ f(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i), (\alpha_1, \ldots, \alpha_n) \in R^n \}$$

- Since H^S is a finite dimensional subspace of H we can decompose any function in H with respect to this subspace by orthogonal projection $f = f_S + f_\perp$

  with $f_S \in H^S$ and $f_\perp \perp H^S$

Proof Representer Theorem (2/2)

- Since H is a RKHS it holds that $\forall_{i=1,\ldots,n}: \; f_{\perp}(x_i) = \langle f_{\perp}, k(.,x_i) \rangle = 0$ and therefore

$$\forall_{i=1,\ldots,n}: \; f(x_i) = f_S(x_i)$$

- By Pythagoras theorem in H we have that $\|f\|_H^2 = \|f_S\|_H^2 + \|f_{\perp}\|_H^2$

- Therefore, we have that

$$\Psi(f(x_1),\ldots,f(x_n),\|f\|) \geq \Psi(f_S(x_1),\ldots,f_S(x_n),\|f_S\|)$$

with equality if and only if $\|f_{\perp}\|_H = 0$

- Hence f* is necessarily in H^S

Representer Theorem

- The representer theorem has an important consequence for us.
- Consider any penalized empirical risk minimization method, where the penalty is in terms of the RKHS norm of f:

$$\hat{f} = \operatorname{argmin}_{f \in H} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \Omega(\|f\|_H)$$

- Then the solution has the form

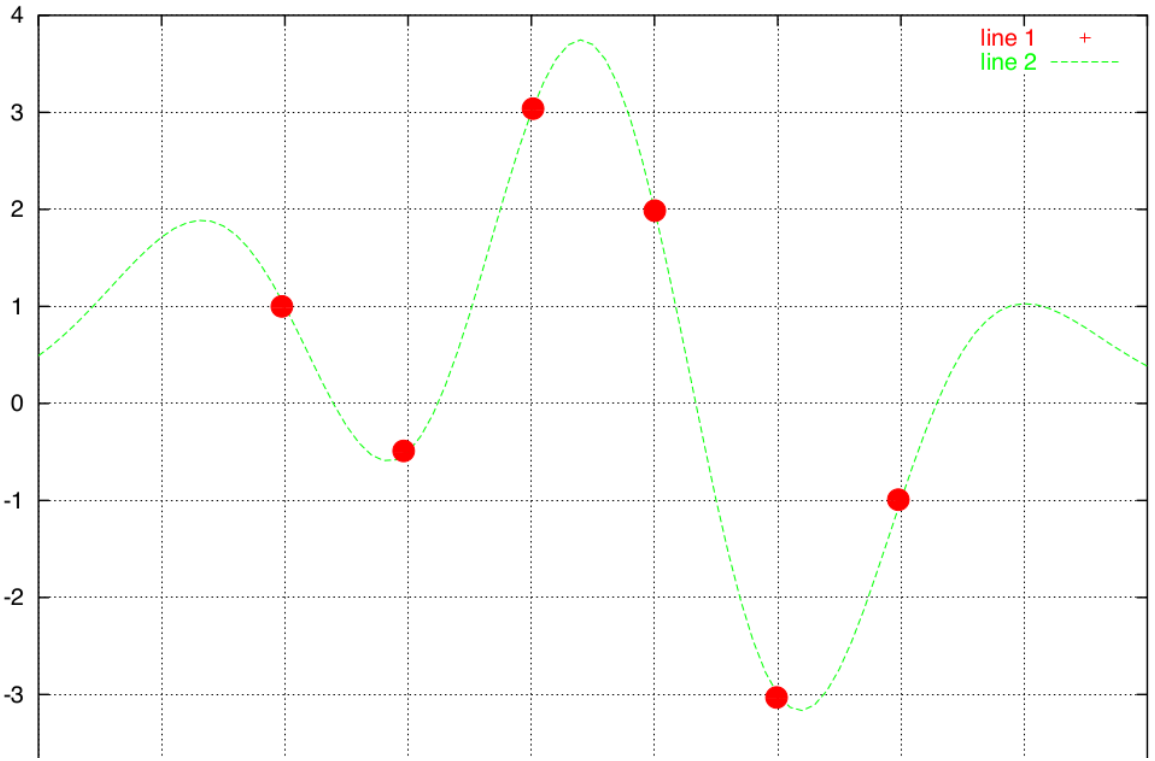
$$\hat{f} = \sum_{i=1}^{n} \alpha_i k(x_i, .)$$

Plan for this lecture

- Kernel trick
  - Distance between points.
  - Distance between sets and points.
  - Data centering.
- Considerations on the RKHS norm
  - Rademacher complexity for RKHS balls.
  - Function smoothness in RKHS.
  - Representer theorem.
- **Supervised kernel methods**
  - Ridge regression.
  - SVM.
  - Logistic discriminant.
- Fisher kernels

Regression

- Let S be a set of n points in X: $S=(x_1,\ldots,x_n)$
- With each element we have an associated target value in R $(y_1,\cdots,y_n)$
- Our goal is to find a function f to predict y by f(x), $f : X \rightarrow R$



Penalized least-squares regression

- Let us use the L2 loss to quantify the error of f with respect to y:

$$L(f(x), y) = (y - f(x))^2$$

- Fix a set of functions H that is the RKHS of a p.d. kernel k on X.
- We estimate f by minimizing the penalized empirical risk:

$$\hat{f} = \operatorname{argmin}_{f \in H} \left\{ \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \|f\|_H^2 \right\}$$

- This regularization has two effects:
  - It prevents overfitting by penalizing non-smooth f, and bounds the Rademacher complexity
  - By the representer theorem, it simplifies the solution to functions that are given by a linear combination of kernel evaluations:

$$\hat{f}(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$$

Dual formulation

$$\hat{f}(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$$

- Let us now redefine the minimization problem in terms of the alpha's.
- Let K be the kernel matrix for the points in S, and $\alpha = (\alpha_1, \ldots, \alpha_n) \in R^n$
- Then we can write

$$\left(\hat{f}(x_1), \ldots, \hat{f}(x_n)\right)^T = K\alpha$$

- Moreover, the squared norm of f can be expressed as

$$\|\hat{f}\|_H^2 = \alpha^T K \alpha$$

- Therefore, the problem is equivalent to

$$\operatorname{argmin}_{\alpha \in R^n} \left\{ \frac{1}{n} (K\alpha - y)^T (K\alpha - y) + \lambda \alpha^T K \alpha \right\}$$

Dual formulation

$$\operatorname{argmin}_{\alpha \in R^n} \left\{ \frac{1}{n} (K\alpha - y)^T (K\alpha - y) + \lambda \alpha^T K \alpha \right\}$$

- Since this is a convex and differentiable function of alpha, its minimum can be found by setting the gradient w.r.t. alpha to zero.

$$\frac{2}{n} K (K\alpha - y) + 2\lambda K \alpha = 0$$

$$K (K\alpha - y + n\lambda \alpha) = 0$$

$$K \left((K + n\lambda I)\alpha - y \right) = 0$$

- Thus, the kernel (in the sense of zero projection) of K should contain

$$(K + n\lambda I)\alpha - y$$

Dual formulation

- Since K is symmetric, it can be diagonalized in an orthonormal basis,

$$K = VDV^T$$

and the kernel Ker(K) corresponds to the subspace with zero on the diagonal in D, and

$$\mathrm{Ker}(K) \perp \mathrm{Im}(K)$$

- This basis remains the same for $K(K+\lambda n I)^{-1}$

since $VDV^T(VDV^T+\lambda n I)^{-1} = VD(D+\lambda n I)^{-1}V^T$

which has diagonal elements of $\dfrac{d_{ii}}{d_{ii}+n\lambda}$

- The problem is thus equivalent to

$$(K+n\lambda I)\alpha - y \in \mathrm{Ker}(K)$$

$$\alpha - (K+n\lambda I)^{-1} y \in \mathrm{Ker}(K)$$

$$\alpha = (K+n\lambda I)^{-1} y + \epsilon, \quad \text{with } K\epsilon = 0$$

Dual formulation

- However, if $\alpha' = \alpha + \epsilon, \quad \text{with } K\epsilon = 0$

  then $\|f - f'\|_H^2 = (\alpha - \alpha')^T K (\alpha - \alpha') = 0$

  and therefore, f=f'.

- Therefore, the solution to the original problem is therefore given by

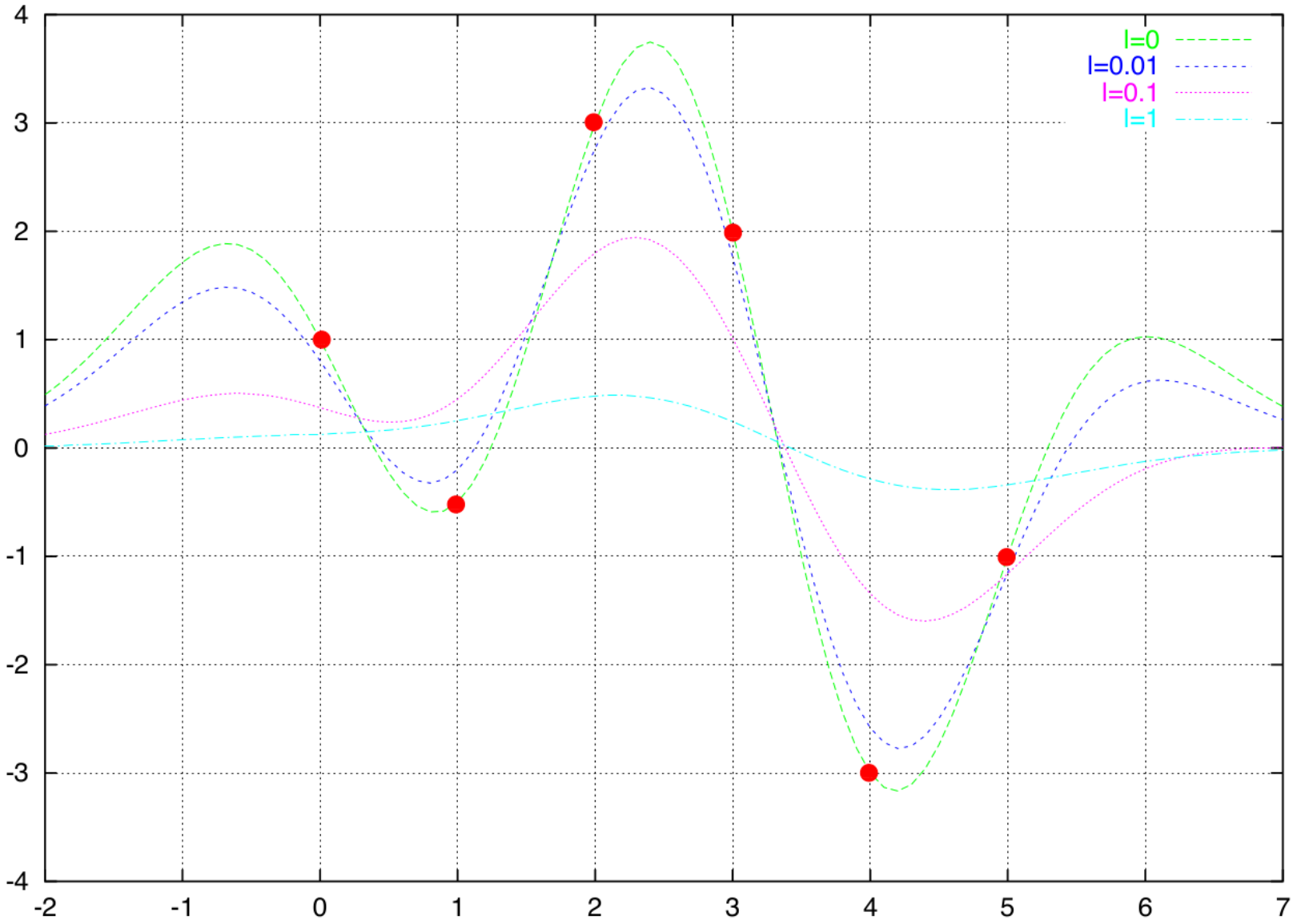
$$\hat{f}(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$$

  with

$$\alpha = (K + n\lambda I)^{-1} y$$

- Note that when lambda goes to zero, the method converges to the classical unregularized least-squares solution. When lambda goes to infinity then the solution converges to f=0.

Example solutions for different regularization values



Plan for this lecture

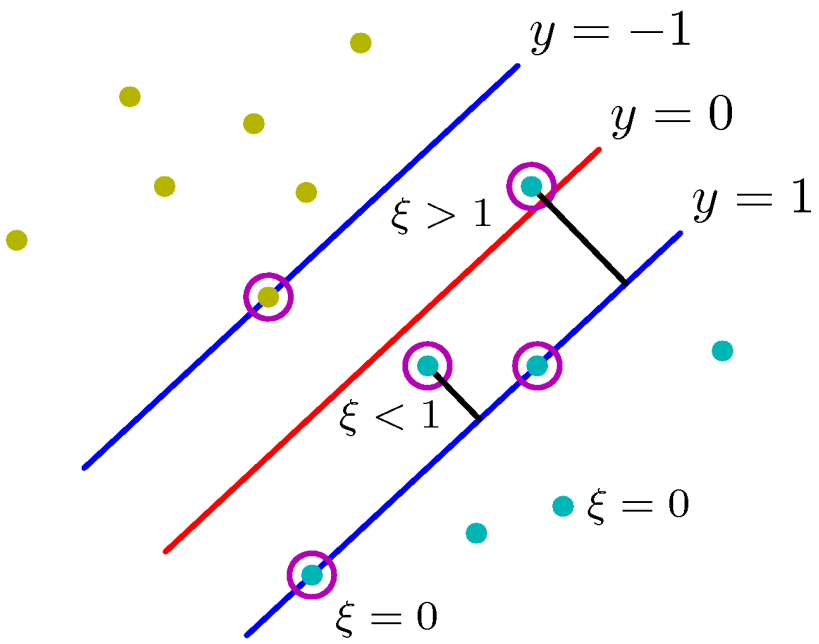
- Kernel trick
  - Distance between points.
  - Distance between sets and points.
  - Data centering.
- Considerations on the RKHS norm
  - Rademacher complexity for RKHS balls.
  - Function smoothness in RKHS.
  - Representer theorem.
- Supervised kernel methods
  - Ridge regression.
  - **SVM.**
  - Logistic discriminant.
- Fisher kernels

Support vector machines revisited

- Quality of classification function measured using hinge-loss

$$L(y_i, f(x_i)) = max(0, 1 - y_i f(x_i))$$

- Recall: convex and piecewise linear upper bound on zero/one loss.
  - Zero if point on the correct side of the margin
  - Otherwise given by absolute difference from score at margin



Support vector machines: optimization problem

- Quality of classification function measured using hinge-loss

$$L(y_i, f(x_i)) = max\left(0, 1 - y_i f(x_i)\right)$$

- Regularization with the norm of f in RKHS associated with kernel k.
- Estimator given by minimizing penalized empirical risk over f in H

$$\hat{f} = \text{argmin}_{f \in H} \left\{ \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \|f\|_H^2 \right\}$$

- This is a convex, but not differentiable objective function.

Support vector machines: reformulated optimization

- Re-formulate as a constrained problem using slack variables

$$\hat{f} = \text{argmin}_{f \in H, \xi \in R^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \|f\|_H^2 \right\}$$

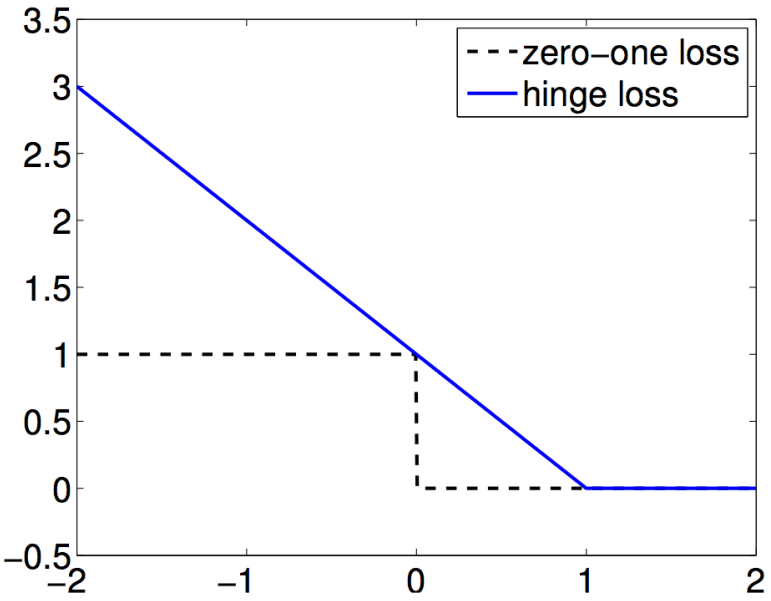
$$\text{subject to: } \xi_i \geq L(f(x_i), y_i)$$

- Rewrite the constraints as a conjunction of linear constraints:

$$\hat{f} = \text{argmin}_{f \in H, \xi \in R^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \|f\|_H^2 \right\}$$

$$\text{subject to: } \xi_i \geq 0 \text{ and } \xi_i \geq 1 - y_i f(x_i)$$

$$L(y_i, f(x_i)) = max(0, 1 - y_i f(x_i))$$



Support vector machines: reformulated optimization

- By the representer theorem we have that

$$\hat{f}(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$$

- Rewrite problem in terms of alpha's

$$\hat{f} = \text{argmin}_{\alpha \in R^n, \xi \in R^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \alpha^T K \alpha \right\}$$

$$\text{subject to: } \xi_i \geq 0 \text{ and } y_i \sum_j \alpha_j k(x_i, x_j) + \xi_i - 1 \geq 0$$

- This is a standard quadratic program, with 2n variables and constraints. Standard QP solvers are suitable for n < 10^4 roughly.

- Highly efficient specific SVM solvers available for much larger problems, in particular for linear SVM case.

Support vector machines: reformulated optimization

- Finally, let us change the notation slightly, from

$$\hat{f} = \operatorname{argmin}_{\alpha \in R^n, \xi \in R^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \alpha^T K \alpha \right\}$$

$$\text{subject to: } \xi_i \geq 0 \text{ and } y_i \sum_j \alpha_j k(x_i, x_j) + \xi_i - 1 \geq 0$$

- To the form

$$\hat{f} = \operatorname{argmin}_{\alpha \in R^n, \xi \in R^n} \left\{ C \sum_{i=1}^{n} \xi_i + \frac{1}{2} \alpha^T K \alpha \right\}$$

$$\text{subject to: } \xi_i \geq 0 \text{ and } y_i \sum_j \alpha_j k(x_i, x_j) + \xi_i - 1 \geq 0$$

- The “cost parameter” C has a natural interpretation in the final solution of the optimization problem.

$$C = 1/(2n\lambda)$$

Support vector machines: Lagrangian

- We introduce Langrange multipliers for the inequality constraints.
- Define the Langrangian of the problem as

$$L(\alpha,\xi,\mu,\nu)=C\sum_{i=1}^{n}\xi_i+\frac{1}{2}\alpha^T K\alpha$$
$$-\sum_{i=1}^{n}\mu_i\left(y_i\sum_j\alpha_j k(x_i,x_j)+\xi_i-1\right)-\sum_{i=1}^{n}\nu_i\xi_i$$

- The Lagrangian can be written in matrix-vector notation as

$$L(\alpha,\xi,\mu,\nu)=C\xi^T 1+\frac{1}{2}\alpha^T K\alpha-\mu^T Y K\alpha-\mu^T\xi+\mu^T 1-\nu^T\xi$$
$$=\xi^T(C1-\mu-\nu)+\mu^T 1+\frac{1}{2}\alpha^T K\alpha-\mu^T Y K\alpha$$

- Where Y is the diagonal matrix with $Y_{ii}=y_i$

Support vector machines: Lagrangian

$$L(\alpha, \xi, \mu, \nu) = \xi^T (C1 - \mu - \nu) + \mu^T 1 + \frac{1}{2} \alpha^T K \alpha - \mu^T Y K \alpha$$

- The Lagrangian is convex quadratic in alpha, and is therefore minimized when gradient is zero, similar to regression case

$$\nabla_\alpha L = K\alpha - KY\mu = K(\alpha - Y\mu) = 0$$

$$\alpha = Y\mu$$

- Lagrangian is linear in xi. Minimum equal to minus infinity, except when gradient with respect to xi is zero:

$$\nabla_\xi L = CI - \mu - \nu = 0$$

$$\mu + \nu = C\,I$$

Support vector machines: Lagrangian

$$L(\alpha,\xi,\mu,\nu)=\xi^T(C1-\mu-\nu)+\mu^T 1+\frac{1}{2}\alpha^T K\alpha-\mu^T Y K\alpha$$

- We obtain the Lagrange dual function as

$$q(\mu,\nu)=\inf_{\alpha,\xi} L(\alpha,\xi,\mu,\nu)$$

- Plugging in the optimal alpha $\alpha=Y\mu$ we get

$$\inf_{\alpha} L(\alpha,\xi,\mu,\nu)=\xi^T(C1-\mu-\nu)+\mu^T 1-\frac{1}{2}\mu^T Y K Y\mu$$

- Adding the minimization over xi we get

$$q(\mu,\nu)=\inf_{\alpha} L(\alpha,\xi,\mu,\nu)=\begin{cases}\mu^T 1-\frac{1}{2}\mu^T YKY\mu & : \text{ if } \nu+\mu=1\mathrm{C} \\ -\infty & : \text{ otherwise}\end{cases}$$

Support vector machines: dual problem

$$q(\mu,\nu)=\inf_{\alpha} L(\alpha,\xi,\mu,\nu)=\begin{cases}\mu^T 1-\frac{1}{2}\mu^T YKY\,\mu & : \text{ if } \nu+\mu=1C \\ -\infty & : \text{ otherwise}\end{cases}$$

- The dual problem consists in maximizing the dual function q, for non-negative Lagrange multipliers:

$$\max_{\mu,\nu} q(\mu,\nu)$$
$$\text{subject to: } \mu\geq 0,\ \ \nu\geq 0$$

- Clearly, for the solution we have $\nu=1C-\mu\geq 0$
- And thus: $\mu\leq 1C$
- Therefore, the dual problem is equivalent to

$$\max_{0\leq\mu\leq 1C}\ \left\{\mu^T 1-\frac{1}{2}\mu^T YKY\,\mu\right\}$$

Support vector machines: dual problem

$$\max_{0 \le \mu \le 1C} \left\{ \mu^T 1 - \frac{1}{2} \mu^T YKY \mu \right\}$$

- Once the dual problem is solved, we can use it to obtain the corresponding alpha vector by $\alpha = Y\mu$ and equivalently $\mu = Y\alpha$
- Therefore, we conclude that $0 \le y_i \alpha_i \le C$
- By complementary slackness, for the solution we have

$$\mu(YK\alpha + \xi - 1) = 0$$
$$\nu\xi = 0$$

- Equivalently, in terms of alpha we have

$$\alpha(YK\alpha + \xi - 1) = 0$$
$$(C1 - Y\alpha)\xi = 0$$

Support vector machines: dual problem

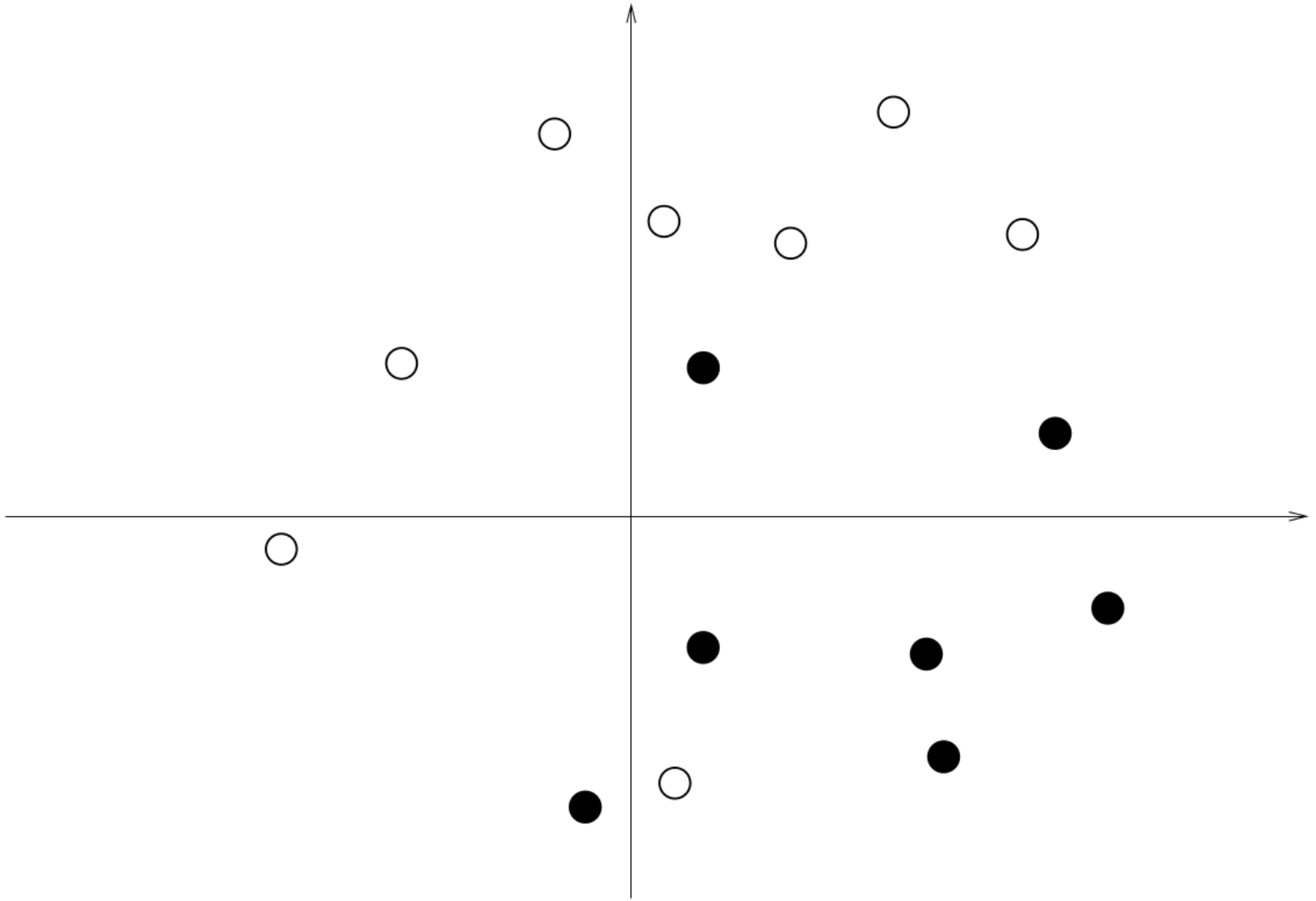
$$\alpha\left(YK\,\alpha+\xi-1\right)=0$$
$$\left(C1-Y\,\alpha\right)\xi=0$$

- If $\alpha_i=0$ then by the second constraint $\xi_i=0$, and by feasibility we therefore conclude that $y_i\sum_{j=1}^{n}\alpha_j K_{ij}=y_i f\left(x_i\right)\geq 1$
  - Thus, points with $\alpha_i=0$ are on the correct side of the margin.

- If $0<y_i\alpha_i<C$ then both constraints are active for the i-th point. Which means that $\xi_i=0$ and thus that $y_i\sum_{j=1}^{n}\alpha_j K_{ij}=y_i f\left(x_i\right)=1$
  - Thus these points are on the margin.

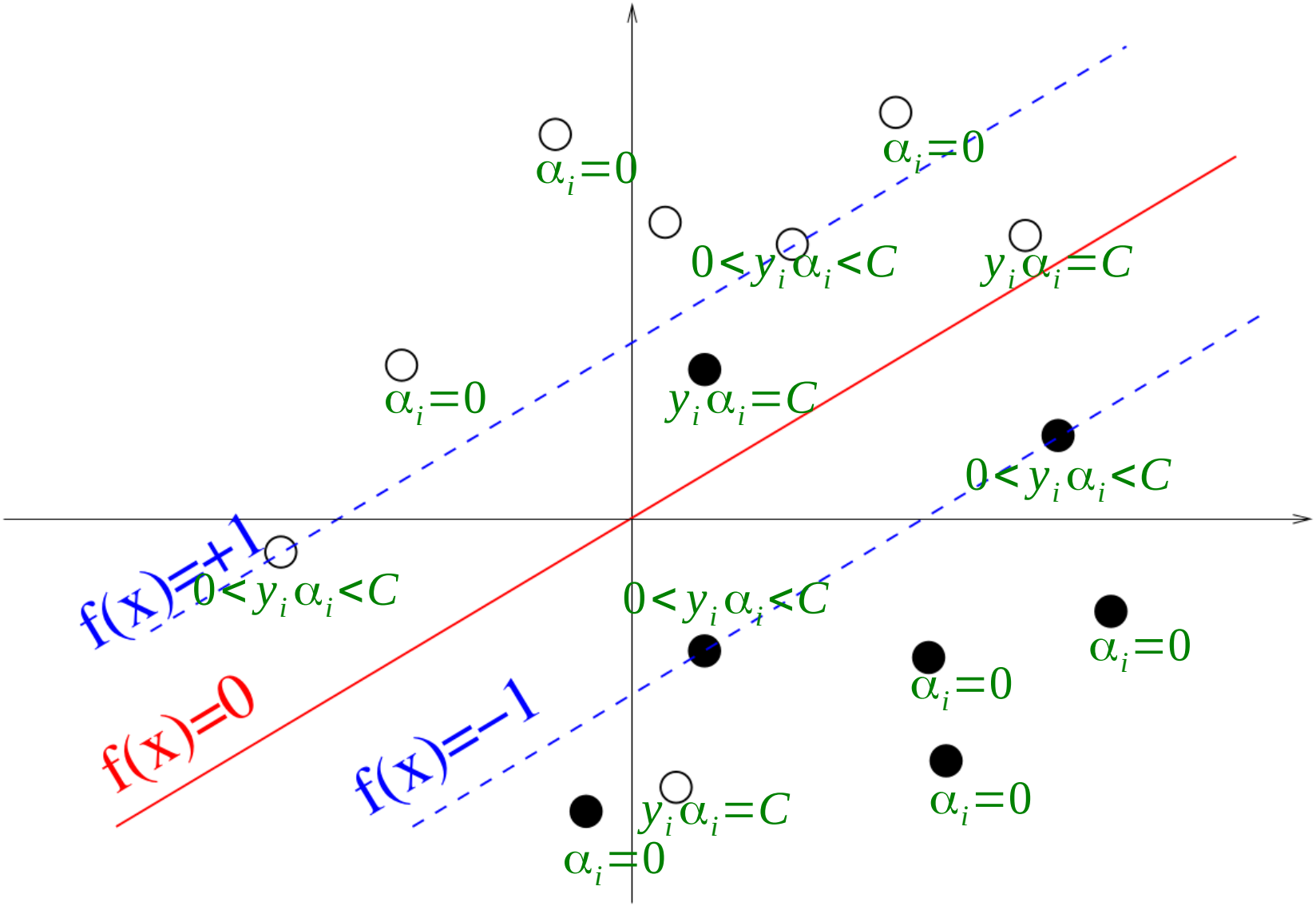
- If $\alpha_i=C$ then the second constraint is not active, $\xi_i\geq 0$ while the first one is: $y_i f\left(x_i\right)=1-\xi\leq 1$
  - Thus these points are on the wrong side of the margin

Support vector machines: geometric interpretation



informatics mathematics

Support vector machines: geometric interpretation



Support vector machines

- The data points with non-zero alpha are called the support vectors.

- Only support vectors are relevant for the classification of new data:

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) = \sum_{i \in SV} \alpha_i k(x_i, x)$$

  - SV is the set of support vectors

- Depending on the problem, the solution can be sparse in alpha.
  - Leads to fast algorithms to identify the subset of non-zero alphas.
  - Makes classification of new points fast, since only a small number of kernel evaluations is needed.

Plan for this lecture

- Kernel trick
  - Distance between points.
  - Distance between sets and points.
  - Data centering.
- Considerations on the RKHS norm
  - Rademacher complexity for RKHS balls.
  - Function smoothness in RKHS.
  - Representer theorem.
- Supervised kernel methods
  - Ridge regression.
  - SVM.
  - **Logistic discriminant.**
- Fisher kernels

Kernel multi-class logistic discriminant

- Map score functions to class probabilities with soft-max

$$p(y=c|x)=\frac{\exp(f_c(x))}{\sum_{c'=1}^{C}\exp(f_{c'}(x))}$$

- Loss function given by negative log-likelihood

$$L(y,\{f_c(x)\})=-\ln p(y|x)=-f_y(x)+\ln\sum_{c=1}^{C}\exp(f_c(x))$$

- Consider class of functions in H which is the RKHS of a p.d. kernel k.
- Estimate the score functions by penalized empirical risk minimization

$$min_{\{f_c\in H\}}\ \lambda\frac{1}{2}\sum_{c=1}^{C}\|f_c\|_H^2+\sum_{i=1}^{n}L(y_i,\{f_k(x_i)\})$$

- By trivial extension of the representer theorem, we have that for the optimal score functions

$$f_c(x)=\sum_{i=1}^{n}\alpha_{ic}k(x_i,x)$$

Kernel multi-class logistic discriminant

$$min_{\{f_c \in H\}} \; \lambda \frac{1}{2} \sum_{c=1}^{C} \|f_c\|_H^2 + \sum_{i=1}^{n} L\left(y_i, \{f_c(x_i)\}\right)$$

- We can now rewrite the optimization problem in terms of the alphas.
- Let us define $\alpha_c = (\alpha_{1c}, \ldots, \alpha_{nc})^T \in R^n$
  and $k_i = \left(k(x_i, x_1), \ldots, k(x_i, x_n)\right)^T \in R^n$

- Now consider the score function of class c for a training point

$$f_c(x_i) = \sum_{j=1}^{n} \alpha_{jc} k(x_j, x_i) = \alpha_c^T k_i$$

- Now consider the optimization problem w.r.t. alpha

$$min_{\{\alpha_c \in R^n\}} \; \lambda \frac{1}{2} \sum_{c=1}^{C} \alpha_c^T K \alpha_c + \sum_{i=1}^{n} \ln \sum_{c'=1}^{C} \exp\left(\alpha_{c'}^T k_i\right) - \sum_{i=1}^{n} \alpha_{y_i}^T k_i$$

  - Where we expanded the loss function as

$$L\left(y_i, \{f_c(x_i)\}\right) = -f_{y_i}(x_i) + \ln \sum_{c=1}^{C} \exp\left(f_c(x_i)\right)$$

Kernel multi-class logistic discriminant

$$min_{\{\alpha_c \in R^n\}} \; \lambda \frac{1}{2} \sum_{c=1}^{C} \alpha_c^T K \alpha_c + \sum_{i=1}^{n} \ln \sum_{c'=1}^{C} \exp(\alpha_{c'}^T k_i) - \sum_{i=1}^{n} \alpha_{y_i}^T k_i$$

- Consider the gradient w.r.t. the alphas

$$\nabla_{\alpha_c} = \lambda K \alpha_c + \sum_{i=1}^{n} p(y=c|x_i) k_i - \sum_{i:y_i=c} k_i$$

$$= \lambda K \alpha_c + \sum_{i=1}^{n} \left(p(y=c|x_i) - [y_i=c] \right) k_i$$

- Let us define the n x C matrix that collects all alphas $A = (\alpha_1, \ldots, \alpha_c)$

$$\nabla_A = \lambda K A + K(\tilde{Y} - P) = K(\lambda A + \tilde{Y} - P)$$

  - Where $\tilde{Y}_{ic} = [y_i = c]$
    and $P_{ic} = p(y=c|x_i)$
  - Note that P depends on A !

Plan for this lecture

- Kernel trick
  - Distance between points.
  - Distance between sets and points.
  - Data centering.
- Considerations on the RKHS norm
  - Rademacher complexity for RKHS balls.
  - Function smoothness in RKHS.
  - Representer theorem.
- Supervised kernel methods
  - Ridge regression.
  - SVM.
  - Logistic discriminant.
- **Fisher kernels**

Fisher kernels

- Proposed by Jaakkola & Haussler, “Exploiting generative models in discriminative classifiers”,In Advances in Neural Information Processing Systems 11, 1998.

- Motivated by the need to represent variably sized objects in a vector space, such as sequences, sets, trees, graphs, etc., such that they become amenable to be used with linear classifiers, and other data analysis tools

- A generic method to define kernels over arbitrary data types based on generative statistical models.

$$p(x;\theta),\ \ x \in X\ ,\ \ \theta \in R^D$$

Fisher kernels

- Given a generative data model

$$p(x;\theta),\ \ x\in X,\ \ \theta\in R^D$$

- Represent data x in X by means of the gradient of the data log-likelihood, or "Fisher score":

$$g(x)=\nabla_\theta \ln p(x),$$
$$g(x)\in R^D$$

- Define a kernel over X by taking the scaled inner product between the Fisher score vectors:

$$k(x,y)=g(x)^T F^{-1} g(y)$$

- Where F is the Fisher information matrix F:

$$F=\boldsymbol{E}_X\left[g(x)g(x)^T\right]$$

- Note 1: The Fisher kernel is a positive definite kernel.
- Note 2: The Fisher kernel is invariant for reparametrization of the model.

Fisher kernels – relation to generative classification

- Suppose we make use of generative model for classification via Bayes' rule
  - Where x is the data to be classified, and y is the discrete class label

$$p(y|x) = p(x|y)\,p(y)/\,p(x),$$
$$p(x) = \sum_{k=1}^{K} p(y=k)\,p(x|y=k)$$

and

$$p(x|y) = p(x;\theta_y),$$
$$p(y=k) = \pi_k = \frac{\exp(\alpha_k)}{\sum_{k'=1}^{K} \exp(\alpha_{k'})}$$

- Classification with the Fisher kernel is at least as powerful as using the Fisher kernel obtained using the marginal distribution p(x) on X.
- This becomes useful when the class conditional models are poorly estimated, either due to bias or variance type of errors.
- In practice often used without class-conditional models, but direct generative model for the marginal distribution on X.

Fisher kernels – relation to generative classification

- Consider the Fisher score vector with respect to the marginal distribution on X

$$\nabla_\theta \ln p(x) = \frac{1}{p(x)} \nabla_\theta \sum_{k=1}^{K} p(x, y=k)$$

$$= \frac{1}{p(x)} \sum_{k=1}^{K} p(x, y=k) \nabla_\theta \ln p(x, y=k)$$

$$= \sum_{k=1}^{K} p(y=k|x) \left[\nabla_\theta \ln p(y=k) + \nabla_\theta \ln p(x|y=k) \right]$$

- In particular for the alpha that model the class prior probabilities we have

$$\frac{\partial \ln p(x)}{\partial \alpha_k} = p(y=k|x) - \pi_k$$

Fisher kernels – relation to generative classification

$$\frac{\partial \ln p(x)}{\partial \alpha_k} = p(y=k|x) - \pi_k$$

- Let the weight vector for the k-th class to be zero, except for the position that corresponds to the alpha of the k-th class where it is one. And let the bias term for the k-th class be equal to the prior probability of that class, then

$$f_k(x) = w_k^T g(x) + b_k = p(y=k|x)$$

- Thus the Fisher kernel based classifier can implement classification via Bayes' rule, and generalizes it to other functions.