第四章：核方法

- 概述
- 核函数
- 核函数用于线性模型
- 核密度估计
- 支持向量机(SVMs)

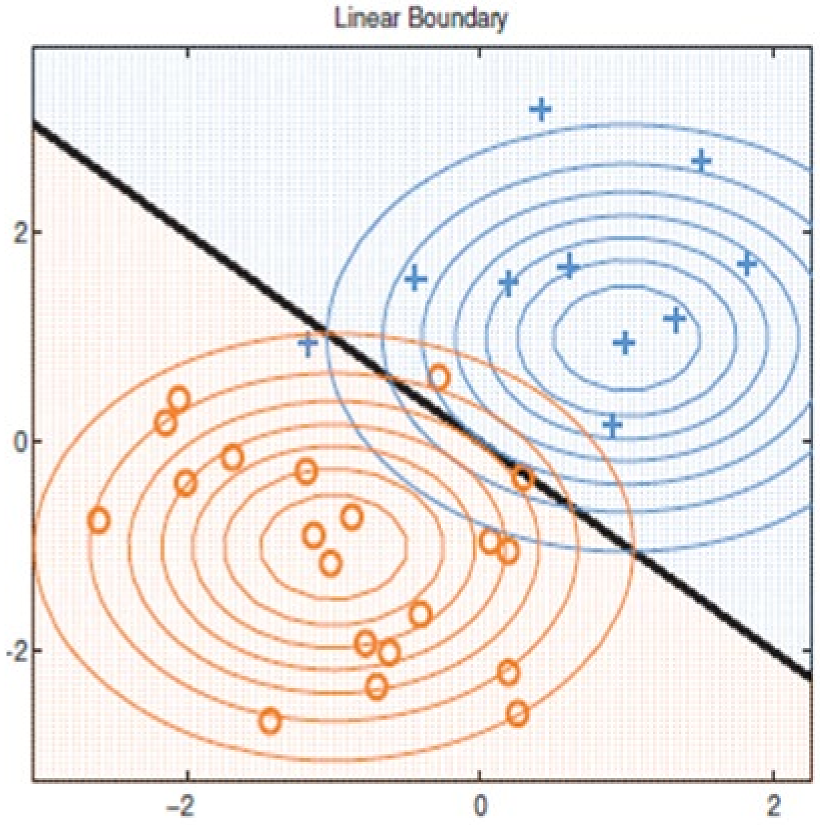
概述

线性可分性

- 设线性判别函数：　$y = w^T x$

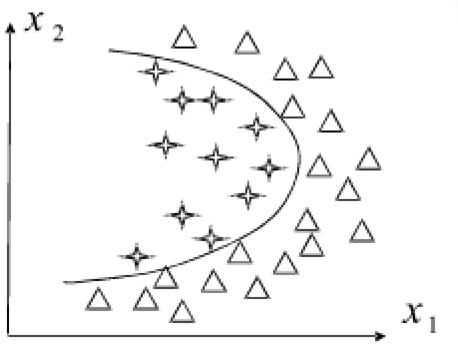
- 如果线性可分
  - ***ω·x>0，则 x∈ C_1***
  - ***ω·x<0，则 x ∈ C_2***
  - ***ω·x=0，则 x 在线性边界上***

- 如图却是线性不可分
  - *不满足上面所列条件*



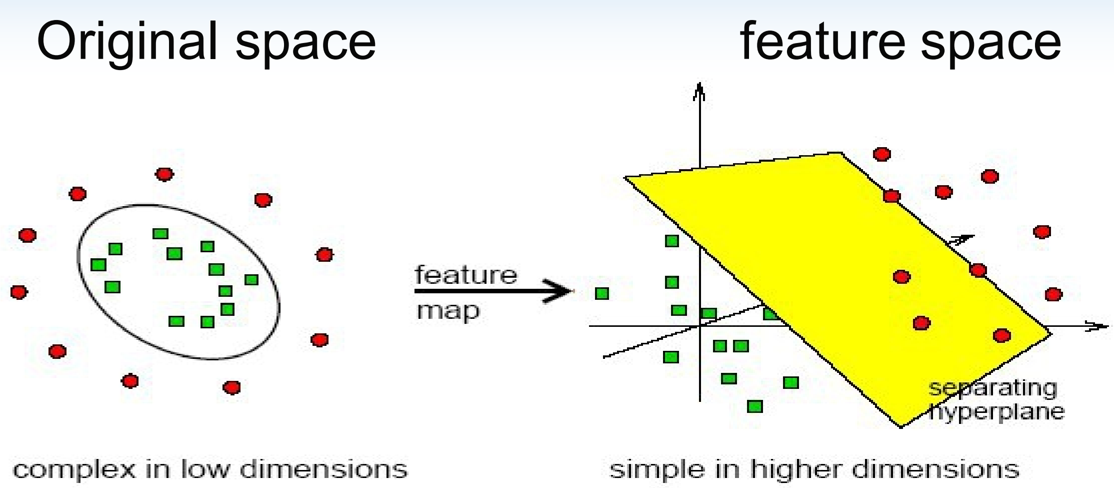
问题的提出

- T如图是一个非线性分类问题



- 解决这类问题，不能直接用线性分类器
- 可以这么做
  - 寻找一个非线性分类器
  - 变换非线性分类问题为线性分类问题
    - ✓ 这样就可以用线性分类器进行处理.

看个例子



- 原始空间中的分类边界是个二次函数:
  - $x_1^2+x_2^2+x_1x_2+2x_1+2x_2+2=0$
- 映射到特征空间: $\Phi: R^2 \rightarrow R^3$
  - $X=(x_1,x_2) \rightarrow Z=(z_1,z_2,z_3) = ((x_1+1)^2, x_1x_2, (x_2+1)^2)$
- 在特征空间中的分类边界变成了平面，线性边界:
  - $z_1+z_2+z_3=0$

特征映射

- 设 $x \in$ 输入空间, H 为特征空间 (Hilbert)
- 特征映射函数:

$$\phi(x): x \rightarrow H$$

- 在特征空间中，如果特征数据是线性可分的
  - $\omega \cdot x > 0$,则 $\Phi(x) \in C_1$
  - $\omega \cdot x < 0$,则 $\Phi(x) \in C_2$
  - $\omega \cdot x = 0$,则 $\Phi(x)$ 在线性边界上

特征间的距离与角度

❖ 两个特征间的距离:

$$\|\phi(x)-\phi(x')\|^2=(\phi(x)-\phi(x'))^T(\phi(x)-\phi(x'))$$
$$=\phi(x)^T\phi(x)-2\phi(x)^T\phi(x')+\phi(x')^T\phi(x')$$
$$=k(x,x)-2k(x,x')+k(x',x')$$

❖ 两个特征间的角度:

$$\because \phi(x)^T\phi(x')=\|\phi(x)\|\bullet\|\phi(x')\|\cos(\theta)$$

$$\therefore \cos(\theta)=\frac{\phi(x)^T\phi(x')}{\|\phi(x)\|\bullet\|\phi(x')\|}$$

$$=\frac{\phi(x)^T\phi(x')}{\sqrt{\phi(x)^T\phi(x)}\sqrt{\phi(x')^T\phi(x')}}=\frac{k(x,x')}{\sqrt{k(x,x)}\sqrt{k(x',x')}}$$

核函数定义

- 满足下列条件的函数，称为核函数
  - $k(x, x) = \varphi(x)^T\varphi(x).$
  - 具有对称性：$k(x, x') = k(x', x).$
  - 函数值非负：$k(x, x') \geq 0$

构造核函数

构造核函数：方法1

- 根据核函数的定义进行构造
  - 寻找映射函数 **$\varphi(x)$**, 从而得到核函数 **$\varphi(x)^T\varphi(x)$.**

构造核函数：方法 2

❖ 直接构造核函数：

- 一个函数*k*为核函数的充分必要条件:
  - ✓ 对于所有输入点集合 {xn}，*k 构成的* Gram 矩阵 **K** 是半正定的
  - ✓ Gram 矩阵

$$\mathbf{K}=\begin{pmatrix} k(x_1,x_1) & \cdots & k(x_1,x_N) \\ & \vdots & \\ k(x_N,x_1) & \cdots & k(x_N,x_N) \end{pmatrix}$$

构造核函数：方法3

- 用已知核函数构造新的核函数
  - 遵循一定的构造规则，将已知核函数作为构件
- 这是一种构造新的核函数的有效方法

核函数的构造规则(1)

- $k_1(,)$ $k_1(,)$, $k_3(,)$是已知核函数:
  - $c > 0$ 为常数， $k(x, x') = ck_1(x, x')$ 为核函数
  - $f(\cdot)$ 是任意函数： $k(x, x') = f(x)k1(x, x')f(x')$ 为核函数
  - $q(\cdot)$ 为非负系数的多项式: $k(x, x') = q\,(k_1(x, x'))$ 为核函数
  - 核函数作为指数: $k(x, x') = \exp(k_1(x, x'))$ 为核函数
  - 核函数的和: $k(x, x') = k_1(x, x') + k_2(x, x')$ 为核函数
  - 核函数的积: $k(x, x') = k_1(x, x')k_2(x, x')$ 为核函数
  - $\varphi(x)$ 是特征映射函数: $k(x, x') = k_3\,(\varphi(x),\varphi(x'))$ 为核函数

核函数的构造规则(2)

- **A** is 是对称半正定矩阵：　　$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{A} \mathbf{x}'$　为核函数
- $\mathbf{k_a}(,)\ \mathbf{k_b}(,)$ 是已知核函数，$\mathbf{x} = (\mathbf{x_a}, \mathbf{x_b})$
  - $k(x, x') = \mathbf{k_a}(\mathbf{x_a}, \mathbf{x}_a') + k_\mathbf{b}(\mathbf{x_b}, \mathbf{x}_b')$　为核函数
  - $k(x, x') = \mathbf{k_a}(\mathbf{x_a}, \mathbf{x}_a')\ k_\mathbf{b}(\mathbf{x_b}, \mathbf{x}_b')$　为核函数

已知核函数：线性核

- 线性核函数: $K(\mathrm{x},\mathrm{x}') = \mathbf{x}^T\mathbf{x}'$
  - 取特征映射函数： $\boldsymbol{\varphi}(\boldsymbol{x}) = \mathbf{x}$:
  - *则有* $\boldsymbol{\varphi}(\boldsymbol{x})^T\boldsymbol{\varphi}(\boldsymbol{x}) = \mathbf{x}^T\mathbf{x}'$
  - 所以, $\mathbf{x}^T\mathbf{x}'$ 为核函数

已知核函数：高斯核

- 高斯核函数:

$$k(x,z)=\exp\left(-\frac{\left\|x-x'\right\|^2}{2\sigma^2}\right)$$

- $\because \|\mathbf{x}-\mathbf{x}'\|^2 = \mathbf{x}^T\mathbf{x} + (\mathbf{x}')^T\mathbf{x}' - 2\mathbf{x}^T\mathbf{x}'$
- $\therefore k(\mathbf{x},\mathbf{x}') = \exp(-\mathbf{x}^T\mathbf{x}/2\sigma^2)\exp(\mathbf{x}^T\mathbf{x}'/\sigma^2)\exp((\mathbf{x}')^T\mathbf{x}'/2\sigma^2)$
- $\therefore$ 高斯核函数是满足要求的

已知核函数：高斯扩展核

❖ 对高斯核函数的距离部分进行扩展

- *用非线性核* $\boldsymbol{\kappa(x, x')}$*, 取代* $\boldsymbol{x^T x'}$
- 则距离变成：$\|\mathbf{x} - \mathbf{x}'\|^2 = \mathbf{x}^T\mathbf{x} + (\mathbf{x}')^T\mathbf{x}' - 2\mathbf{x}^T\mathbf{x}' = \kappa_1(\mathbf{x}, \mathbf{x}') + \kappa_1(\mathbf{x}', \mathbf{x}') - 2\kappa_1(\mathbf{x}, \mathbf{x}')$
- *因此，得到高斯扩展核函数：*

$$k(x,z) = \exp\left\{-\frac{1}{2\sigma^2}(k_1(x,x) + k_1(x',x') - 2k_1(x,x'))\right\}$$

核观点的扩展

- ***对输入的扩展***
  - 将实数向量扩展到符号元素.
- ***这样，核函数就可以定义在对象上，比如：***
  - 图形,
  - 集合,
  - 字符串,
  - 文本

一个定义在集合上的核函数

- 给定集合 Ω, 非向量空间 U 定义为:

$$U=\{A \mid A\subseteq\Omega\}=2^{\Omega}\qquad \text{(幂集)}$$

- 在U上定义一个函数, 对于任意的 $A_1\in U, A_2\in U$

$$k(A_1,A_2)=2^{|A_1\cap A_2|}$$

- 证明k为核函数
  - *定义特征映射函数：* $\varphi(A)=2^{A}$， $\varphi(B)=2^{|B|}$
  - *定义 φ() 函数的点积：* $\varphi^{T}(A)*\varphi(B)=|2^{A}\cap 2^{B}|=2^{|A\cap B|}$
  - $\therefore\ \varphi(A_1)^{T}\varphi(A_2)=2^{|A1\cap A2|}=k(A_1,A_2)$
  - *所以,* $k(A_1,A_2)$ *是核函数*

构造核函数举例

- 已知线性核函数：$\mathbf{x}^T\mathbf{x}'$
- 根据构造规则，我们可以得到下面一些核函数:
  - $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'+\mathbf{c}$,　 $c>0$
  - $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}'+\mathbf{c})^2$, $c>0$
  - $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}'+\mathbf{c})^M$, $c>0$

基于概率模型定义核函数

- 给定概率模型 $p(x)$，*可以定义核函数*
  - $k(x, x') = p(x)p(x')$
- 证明其满足核函数要求
  - *将概率分布函数* $p(\mathbf{x})$ *看成为映射函数* $\Phi(\mathbf{x})$
  - $k(x, x') = p(x)p(x') = \Phi^T(x)\ \Phi(x')$
  - $k(x, x') = p(x)p(x') = p(x')p(x)$，*满足对称性*
  - $k(x, x') = p(x)p(x') \geq 0$

概率核模型的扩展

- 已知概率核模型： $k(x, x') = p(x)p(x')$
- 根据构造规则: $k(x,x') = ck_1(x,x')$, $k(x,x') = k_1(x,x') + k_2(x,x')$,
- 可知下面函数也是核函数

$$k(x,x') = \sum_i p(x|i)p(x'|i)p(i) \qquad k(x,x') = \int p(x|z)p(x'|z)p(z)dz$$

- 如果数据是有序序列：观测值 $\mathbf{X} = \{x_1, \ldots, x_L\}$， 对应隐状态 **Z={z1,…,zL}**
- 下面函数也是核函数

$$k(\mathbf{X},\mathbf{X}') = \sum_{\mathbf{Z}} p(\mathbf{X}|\mathbf{Z})p(\mathbf{X}'|\mathbf{Z})p(\mathbf{Z})$$

Mercer核函数(正定核函数)

- 如果一个核函数满足下面条件
  - 对于任意一组输入 $\{xi\}^N_{i=1}$.，其Gram 矩阵是正定的

$$\text{Gram} = \begin{pmatrix} f(x_1, x_1) & \cdots & f(x_1, x_N) \\ & \vdots & \\ f(x_n, x_1) & \cdots & f(x_N, x_N) \end{pmatrix}$$

- 核函数分析
  - ∵Gram 矩阵是正定的，∴ $K = U^T \Lambda U$ (Λ为特征值对角矩阵)
  - 其中，$k_{ij} = (\Lambda^{1/2} U_{:i})^T (\Lambda^{1/2} U_{:j})$
  - 设$\Phi(x_i) = (\Lambda^{1/2} U_{:i})$，则 $k_{ij} = \Phi(x_i)^T \Phi(x_j)$
  - 这表明，核矩阵中的项，可由特征向量U的点积来计算。
  - 如果核是Mercer，则存在映射φ: $x \in X \rightarrow R^D$

构造Mercer核函数

- 一般来说，确定一个Mercer核函数是困难的
  - 对需要函数分析的技术
- 构造Mercer核函数方法
  - 使用一套标准规则，可以从简单的核函数，构建新的Mercer核函数
  - 例如，如果κ1和κ2都是Mercer，那么κ(x, x’) = κ1(x, x’)+κ2(x,x’)也是

Fisher 核函数

❖ 使用生成模型定义核函数的更有效方法：

- *利用 log似然函数：*log*p*(*x*|*θ*)
- *构造 log似然的梯度向量：g(x)，又称为得分(score)向量，最大似然估计参数*

$$g(\theta,x)=\nabla_{\theta}\log p(x\,|\,\theta)\,|_{\hat{\theta}}$$

- **构造***Fisher 信息矩阵：***F**

$$\mathbf{F}=E_{x}[g(\theta,x)g(\theta,x)^{T}]$$

- **可以看出，如果E(***g(θ,x)***)***=0，***F** *等于g(θ,x)的协方差矩阵，即* **F= D(***g(θ,x)***)**
- *取 Fisher核函数：*

$$k(x,\,x')\;=\;g(\theta,\,x)^{T}\mathrm{F}^{-1}g(\theta,\,x')$$

Fisher 核函数的含义

- 证明得分(score)向量 $g(\theta,x)$ 的均值等于0：

$$
\begin{aligned}
E(g(\theta, x)) &= E(\nabla_\theta \log p(x \mid \theta)) = \int \nabla_\theta \log p(x \mid \theta) p(x \mid \theta) dx \\
&= \int \frac{\nabla_\theta p(x \mid \theta)}{p(x \mid \theta)} p(x \mid \theta) dx = \int \nabla_\theta p(x \mid \theta) dx \\
&= \nabla_\theta \int p(x \mid \theta) dx = \nabla_\theta 1 = 0
\end{aligned}
$$

- Fisher 信息矩阵的含义：
  - 因为 **F= D($g(\theta,x)$)**，表明 ***x*** 本身对参数θ影响的大小，也就是包含多少关于θ的信息
- Fisher核函数的含义：
  - Fisher核函数表示score函数与均值的马氏距离
  - 也即表示目前参数与最佳参数的距离

近似 Fisher 核函数

- 在实践中，计算Fisher信息矩阵**F**通常做不到。
- 一种近似方法
  - 用样本平均值近似Fisher信息矩阵

$$\mathrm{F} \approx \frac{1}{N}\sum_{n=1}^{N} g(\theta, x) g(\theta, x)^{T}$$

- 还可以省略Fisher信息矩阵:

$$k(x, x') = g(\theta, x)^{T} g(\theta, x')$$

Sigmoid核函数

- 如果核函数是tanh()函数，即

$$k(x, x') = \tanh(ax^T x' + b)$$

- 则称为sigmoid核函数
- 它不是一个Mercer 核函数

径向基核函数（RBF kernel）

- 如果一个核函数是一个径向基函数
  - 即它仅仅是 **||x − x||** *的函数：* $\phi_j = h(\| x - \mu_j \|)$
- 比如，

$$k(x, x') = \exp\left(- \frac{\left\| \mathrm{x} - \mathrm{x}' \right\|^2}{2\sigma^2} \right)$$

核函数用于广义线性学习机

核函数用于分类

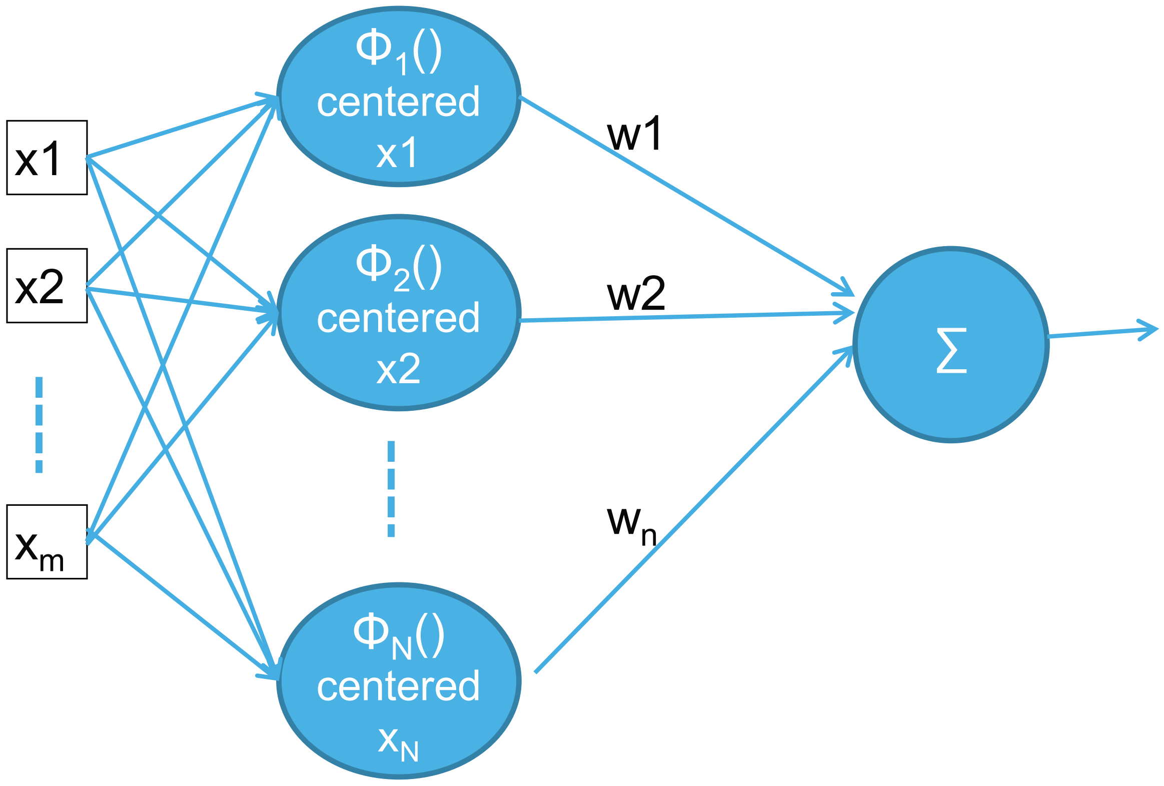
- 定义基于核的输入特征向量：$\Phi(x)=(k(x,\mu_1), \ldots, k(x,\mu_K))$
  - $\mu_1,\ldots, \mu_K$为一组质心
- 分类函数

$$f(x)=\sum_n w_n \phi_n(\| x-x_n \|)$$

- 如果核函数$k()$是径向基(RBF)核函数，则分类器称为径向基网络

径向基网络

$$f(x)=\sum_n w_n\phi_n(\|x-x_n\|)$$

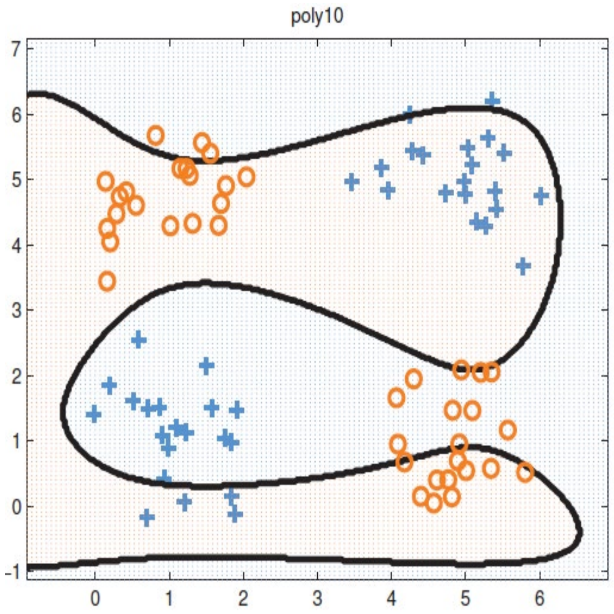


- 常用核函数：高斯核

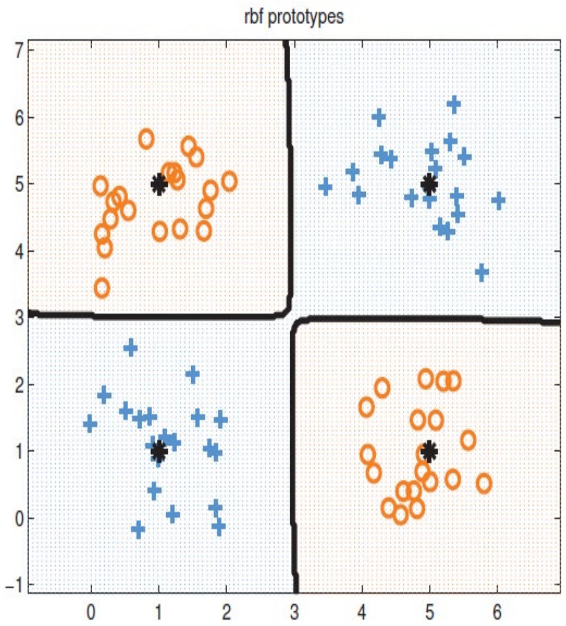
$$\phi(x)=\exp\left\{-\frac{x^2}{2\sigma^2}\right\}$$

核函数用于回归

- 基于核的输入特征向量：$\Phi(x)=(k(x,\mu_1), \ldots, k(x,\mu_K))$
- 定义$p(y|x,\theta) = \mathrm{Ber}(w^T\varphi(x))$，可以将基于核的特征向量用于逻辑回归
  - 这是一种简单的定义非线性决策边界方法。
- 例如



10次多项式展开拟合线性逻辑回归分类器



使用径向基网络，4个质心

核函数用于回归

- 基于核的学习机的主要问题：如何选择质心μ_k？
  - 对于低维欧氏空间，可以使质心均匀平铺数据所占空间，由于维度灾难，在高维空间不可行
  - 另一方法是对数据聚类，为每个类中心分配一个质心。
    - 但，聚类是无监督学习，可能不会产生对预测有用的表示法。而且，还需要给定聚类的数量。
  - 一种更简单的方法是将每个样本xi都作为一个质心，
    - $\varphi(x)=[\kappa(x, x_1), \ldots, \kappa(x, x_N)]$
    - 这里，D=N，参数与样本点一样多。采用稀疏线性模型，去掉部分样本点，这种方法称为稀疏向量机
    - L1VM，L2VM
  - 创建稀疏核学习机的另一种非常流行的方法：支持向量机(SVM)。

稀疏线性模型

- 我们已经知道，输入变量与输出的互信息，可以选择特征
  - 问题在于，它一次只看一个变量。如果存在交互效果，则可能失效
  - 例如，如果y=xor(x1，x2)，x1和x2本身都不能预测响应，但它们一起可以完美地预测响应。
- 稀疏线性模型：基于模型一次选择变量集。
  - 如果模型是一个广义线性模型，$p(y|x) = p(y|\, f(w^T x))$，
    - 可以通过促使权重向量w稀疏来选择特征，即有很多零。
    - 常用方法有：ℓ_0，ℓ_1，ℓ_2规则化

模型参数规则化

- 监督学习中，通常求解模型的目标是“基于参数规则化的模型误差最小”

$$\omega^{*} = \underset{\omega}{\operatorname{argmin}} \left(\sum_{n} L(y_n, f(x_n, \omega)) + \lambda \Omega(\omega) \right)$$

  - 其中，Ω(ω)为规则化项，约束模型的参数尽可能多的为0，使模型尽量的简单

- Ω(ω)常用的模型有：零范数规则化项、一范数规则化项、二范数规则化项

  - ℓ_0范数：$||\omega||_0$，向量中非0元素个数
  - ℓ_1范数：$||\omega||_1$，向量中各元素绝对值之和，也称叫“稀疏规则算子”（Lasso regularization）
  - ℓ_2**范数：** $||\omega||_2$，向量各元素的平方和然后开方，也称叫“权值衰减”(weight decay)。

模型参数规则化

- 我们不需要用核函数φ（x）=[κ（x，x1），…，κ（x，xN）]来定义我们的特征向量，而是可以使用原始特征向量x，但要修改算法，使其用对核函数κ（xx，x）的调用来替换形式为x，x的所有内积。这称为内核技巧。事实证明，许多算法都可以用这种方式进行内核化。下面我们给出一些例子。请注意，我们要求内核是Mercer内核才能使此技巧发挥作用。

$$\omega^* = \arg\min_{\omega}\left(\sum_n L\left(y_n, f\left(x_n;\omega\right)\right) + \lambda\Omega(\omega)\right)$$

  - 其中，Ω(ω)为规则化项，约束模型的参数尽可能多的为0，使模型尽量的简单

- Ω(ω)常用的模型有：零范数规则化项、一范数规则化项、二范数规则化项
  - ℓ_0范数：$||\omega||_0$，向量中非0元素个数
  - ℓ_1范数： $||\omega||_1$，向量中各元素绝对值之和， 也称叫“稀疏规则算子”（Lasso regularization）
  - ℓ_2**范数**： $||\omega||_2$，向量各元素的平方和然后开方，也称叫“权值衰减”(weight decay)。

支持向量机(SVM)

2022/10/24

智能与多媒体实验室 40

最大间隔分类器

- 支持向量机中最简单的模型
- 它也是最早提出的模型
- 它只适用于特征空间中线性可分的数据
- 它是更加复杂的支持向量机算法的主要模块
- 它展示了这类学习器的关键特征

2022/10/24

智能与多媒体实验室 41

2类分类问题

- 使用线性分类模型：

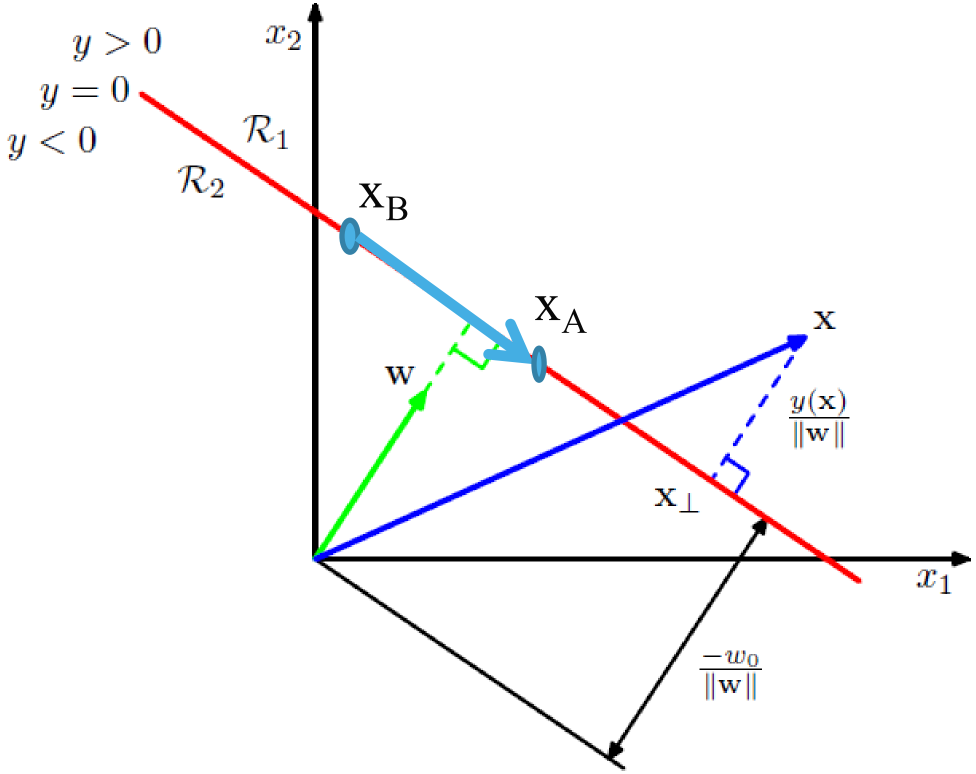
$$y(x) = w^T \phi(x) + b$$

- 训练集 D={ ($\boldsymbol{x_1}$, t_1), . . . , ($\boldsymbol{x_N}$, t_N)}，$t_n \in \{-1, 1\}$
- *基于分类器$\mathbf{y(x)}$的符号，对新数据点 $\mathbf{x}$ 进行分类*
- 假定问题是线性可分的
- 所以，至少存在一组参数 **w 和** b
  - $y(x_n) > 0$ 当 $t_n = +1$，$y(x_n) < 0$ 当 $t_n = -1$,
  - 所以，对于所有x_n，$t_n y(x_n) > 0$

2022/10/24 智能与多媒体实验室 42

决策边界的方向

- 决策边界由分类模型决定： $y(\boldsymbol{x}) = \mathrm{w}^T\mathrm{x} + \mathrm{w}_0$
- 设点 x_A 和 x_B 在决策边界上
- $\therefore\ y(\mathrm{x}_A) = y(\mathrm{x}_B) = 0,$
- $\therefore\ \mathrm{w}^T(\mathrm{x}_A - \mathrm{x}_B) = 0$
- $\therefore$ w 决定了决策边界的方向



任意点x 到边界的距离

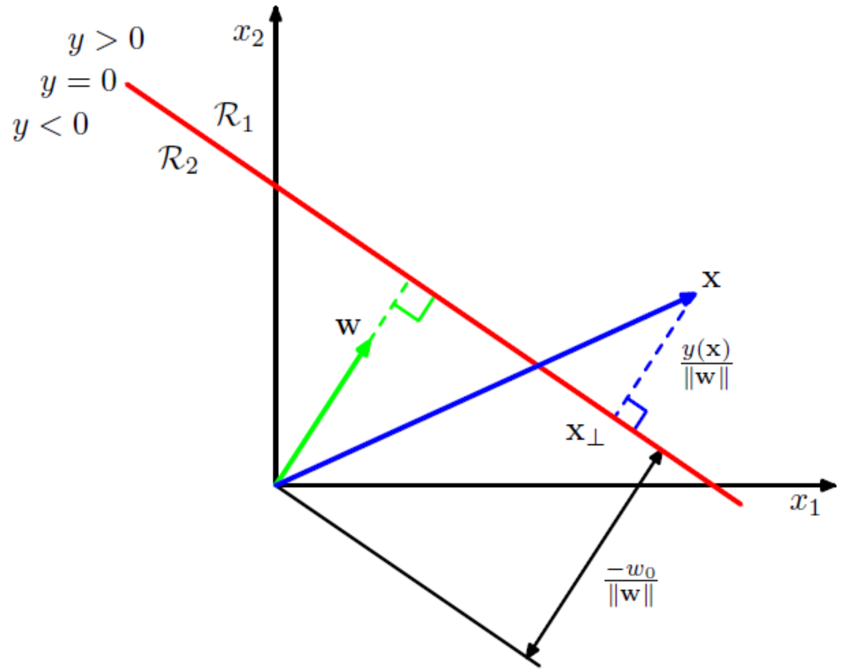
- 决策边界：$y(\mathbf{x}) = \mathrm{w}^T\mathrm{x} + \mathrm{w}_0$，设$x$到边界的距离为$r$
- $\because y(\mathrm{x}_{\perp}) = \mathrm{w}^T\mathrm{x}_{\perp} + \mathrm{w}0 = 0,$
- $\therefore$我们有：

$$x = x_{\perp} + r\frac{w}{\|w\|}$$

$$w^T x + w_0 = w^T(x_{\perp} + r\frac{w}{\|w\|}) + w_0$$

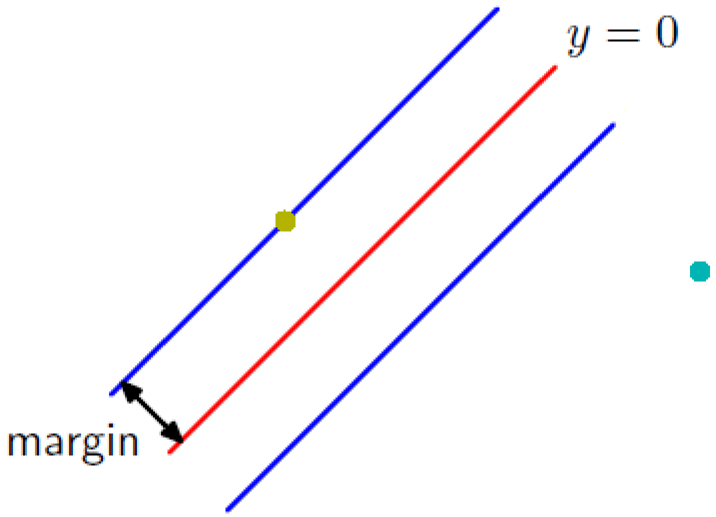
$$y(x) = r\|w\|$$

$$r = \frac{y(x)}{\|w\|}$$



边界间隔（margin）

- 训练集中任意样本到决策边界距离的最小值



边界范围最大化

- 所有数据被正确分类： $t_n y(x_n) > 0$ (对于所有n).
- 数据点 x_n 到决策边界的距离:

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n(w^T\phi(x_n)+b)}{\|w\|}$$

- 优化参数w 和 b，使边界范围最大化

$$\arg\max_{w,b}\left\{\frac{1}{\|w\|}\min_n[t_n(w^T\phi(x_n)+b)]\right\}$$

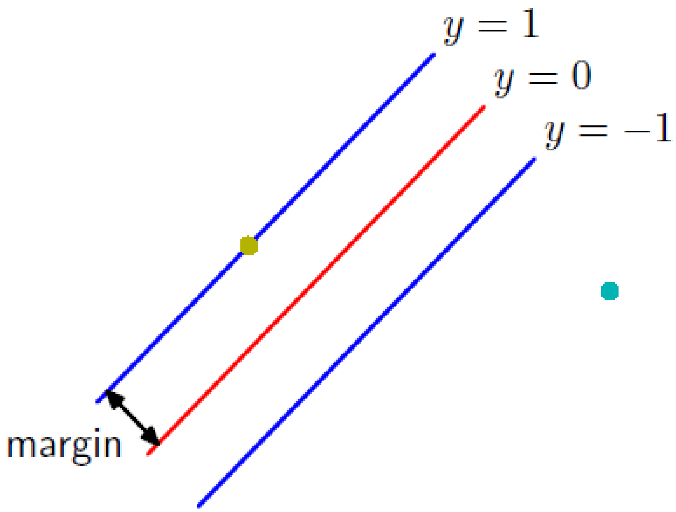
- 直接求解这个最优化问题非常复杂

边界间隔最大化的等价问题

- 我们知道：对于所有x_n，$t_n y(x_n) > 0$
- 对决策边界做一个尺度变换： $w \to \kappa w$ ， $b \to \kappa b$,
  - 使得对于所有点x_n：　$t_n y(x_n) \geq 1$
  - 而离边界最近的点x_n : $t_n y(x_n) = 1$
- 边界范围最大化问题就变成：$\mathbf{maximize}\ \|\mathbf{w}\|^{-1}$,
  - 等价于：

$$\arg\min_{w,b} \frac{1}{2} \| w \|^2$$

$$\text{s.t.} \quad t_n y(x_n) \geq 1$$



拉格朗日函数

- 给定一个最优化问题： Minimize f(w) w∈Ω⊆Rn
  - s.t. gi(w)≤0
  - hi(w)=0
- 拉格朗日函数定义为：

$$L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{k} \beta_i h_i(\mathbf{w})$$

$$= f(\mathbf{w}) + \boldsymbol{\alpha}\,\mathbf{g}(\mathbf{w}) + \boldsymbol{\beta}\,\mathbf{h}(\mathbf{w})$$

Kuhn-Tucker定理

- 给定一个最优化问题： Minimize f(w) w∈Ω⊆Rn
  - s.t. gi(w)≤0
  - hi(w)=0
  - 一个点w*是最优点的充要条件：
    - ⑩存在α*，β*，满足：

$$\frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial w} = 0$$

$$\frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial \beta} = 0$$

$$\alpha_i^* g_i(w^*) = 0 \qquad i = 1, \ldots, k$$

$$g_i(w^*) = 0 \qquad i = 1, \ldots, k$$

$$\alpha_i^* \geq 0 \qquad i = 1, \ldots, k$$

求解边界间隔最大化问题

- 运用拉格朗日乘数法，($a_n \geq 0$) :

$$L(w,b,\mathbf{a}) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n\{t_n(w^T\phi(x_n)+b)-1\}$$

- 根据**Kuhn-Tucker定理**，对 L(w, b, a) 求偏导，并令其为0 :

$$w = \sum_{n=1}^{N} a_n t_n \phi(x_n) \qquad 0 = \sum_{n=1}^{N} a_n t_n \qquad a_n\{t_n y(x_n)-1\} = 0$$

- 将 w，b 带入 ***L(w, b, a)***，得到

$$L(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

一个新的受约束最优化问题

$$\arg\min_{\mathbf{a}}\left(L(\mathbf{a})\right) = \arg\min_{\mathbf{a}}\left(\sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)\right)$$

s.t. $a_n \geq 0$ $\qquad k(x,x') = \phi(x)^T \phi(x')$

$$0 = \sum_{n=1}^{N} a_n t_n$$

- 这是一个二次规划问题
- M个变量的二次规划通常具有O(M³)的计算复杂性。

分类器的新形式

- 分类器的原始形式：

$$y(x) = w^T \phi(x) + b$$

- 因为参数：

$$w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

- 所以，有分类器的新形式：

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$$

- 其中，

$$a_n \geq 0,$$
$$a_n \{t_n y(x_n) - 1\} = 0$$
$$t_n y(x_n) - 1 \geq 0,$$

支持向量

- 在分类器的新形式下，要求满足

$$a_n \geq 0 , \qquad a_n\{t_n y(x_n) - 1\} = 0 \qquad t_n y(x_n) - 1 \geq 0 ,$$

- 所以，对于所有的数据点，必须，或者 $a_n = 0$ 或者 $t_n y(x_n)$ -1=0 → $t_n y(x_n)$ =1
  - 对于对应于$a_n = 0$ 的那些点x_n，无论 $t_n y(x_n)$ -1等于多少，定理都满足.
  - 所以，只需要考虑$t_n y(x_n)$ -1=0 的那些点，它们位于边界的间隔线上
- 我们称 $t_n y(x_n) = 1$,的那些点为支持向量(**support vectors)**

参数b的确定

- 任意支持向量中的点 x_n，满足： $t_n y(x_n) = 1$, 即：

$$t_n\left(\sum_{m\in S} a_n t_n k(x,x_n)+b\right)=1$$

  - 这里，S 表示支持向量的点集合

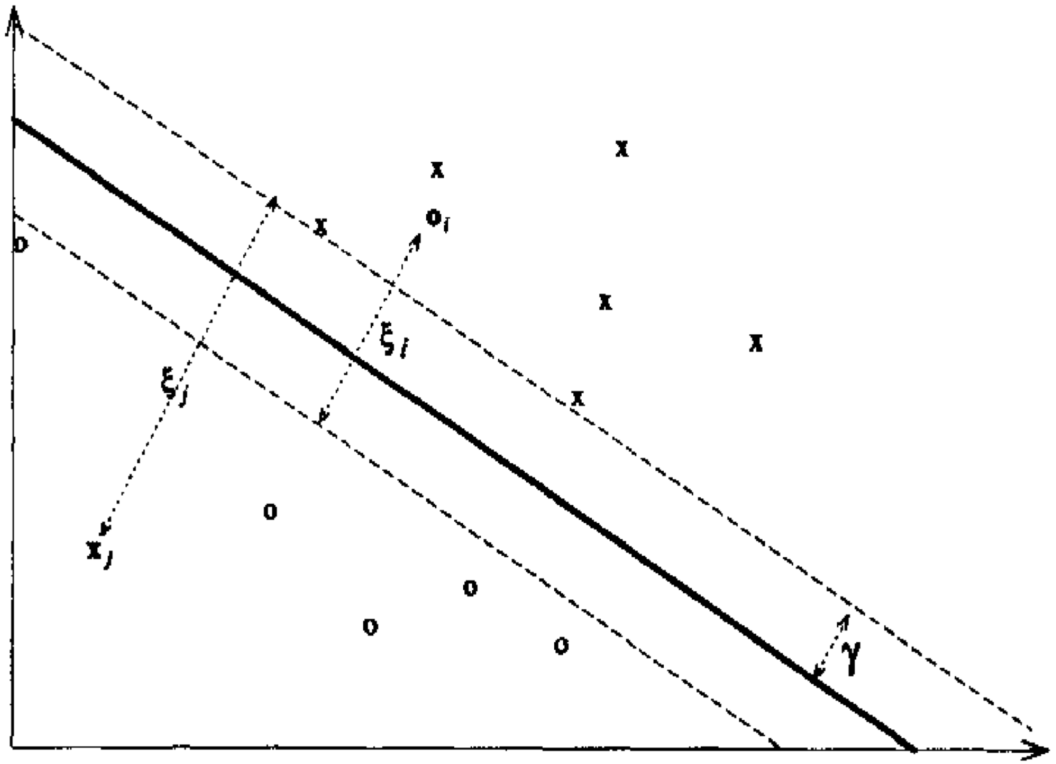
- 计算 b 的步骤：
  - 因为 $t_n^2 = 1$, 所以，上式两边同乘t_n
  - 对所有支持向量的点，求上式，然后求它们的和，得到：

$$b=\frac{1}{N_S}\sum_{n\in S}\left(t_n-\sum_{m\in S} a_n t_n k(x_n,x_m)\right)$$

- N_S is 表示支持向量中点的个数

软间隔优化

- 最大间隔分类器是一个重要的方法，但数据线性不可分，就不能使用
  - 如果数据有噪声，特征空间一般线性不可分
  - 最大间隔分类器总是完美地产生一个没有训练误差的一致假设
  - 当数据不能完全分开时，间隔是个负数
- 为了能优化间隔，引入松弛变量
  - 允许在一定程度上违反间隔约束
  - 定义：固定r>0，样本对应于目标间隔r的松弛变量
  - $\xi_n = \max(0，r - t_n y(x_n))$



引入松弛变量

- 回忆最大间隔分类的优化问题：

$$\arg\min_{w,b} \frac{1}{2} \| w \|^2$$

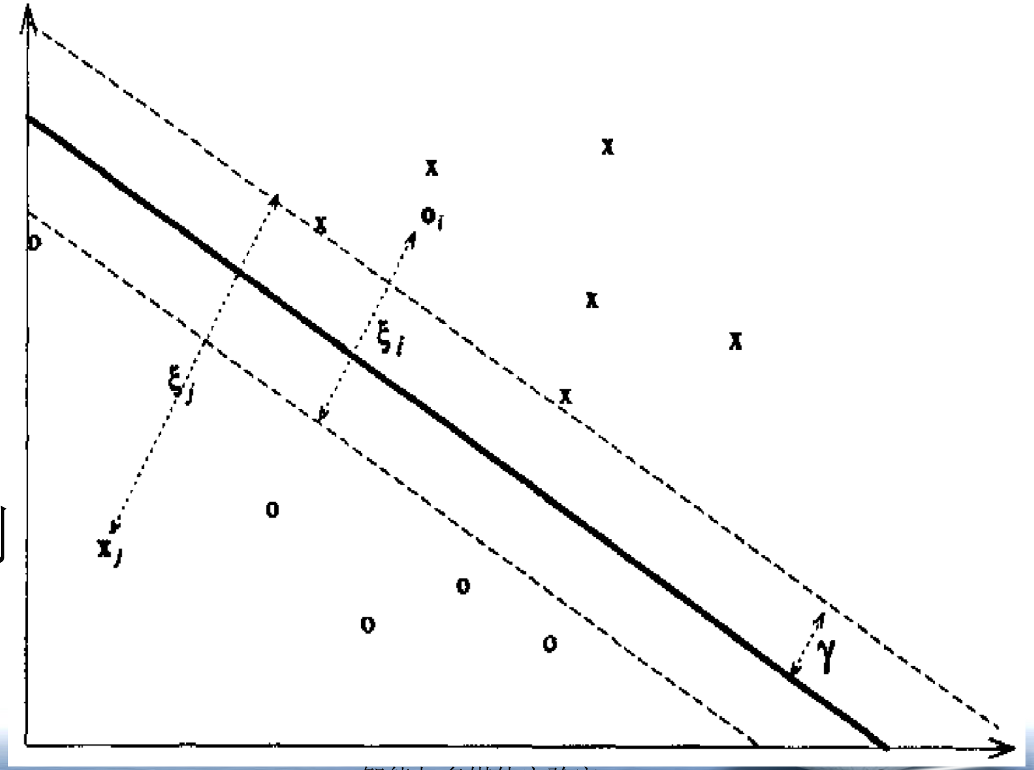
$$\text{s.t.} \quad \mathbf{t}_n \mathbf{y}(\mathbf{x}_n) \geq \mathbf{1}$$

- 引入松弛变量，约束变成：

$$\text{s.t.} \quad \mathbf{t}_n \mathbf{y}(\mathbf{x}_n) \geq \mathbf{1} - \xi_n$$

$$\xi_n \geq 0$$

- $\xi_n = 0$, :正确分类，点在间隔上或间隔的正确一侧
- $0 < \xi_n \leq 1$: 点位于间隔内，但位于间隔的正确一侧
- $\xi_n > 1$: 点位于间隔的错误一侧，分类错误



优化问题变成为

$$\arg\min(C\sum_{n=1}^{N}\xi_n+\frac{1}{2}\|\mathrm{w}\|^2)$$

- C > 0 控制松弛变量于间隔的平衡.
- 拉格朗日函数变成为：

$$L(\boldsymbol{\omega},b,\boldsymbol{a})=\frac{1}{2}\|\boldsymbol{\omega}\|^2+C\sum_{n=1}^{N}\xi_n-\sum_{n=1}^{N}a_n\{t_n y(x_n)-1-\xi_n\}-\sum_{n=1}^{N}\mu_n\xi_n$$

- 根据**Kuhn-Tucker**定理，得到

$$\frac{\partial L}{\partial \omega}=0 \quad \Rightarrow \quad \boldsymbol{\omega}=\sum_{n=1}^{N}a_n t_n \Phi(x_n)$$

$$\frac{\partial L}{\partial b}=0 \quad \Rightarrow \quad \sum_{n=1}^{N}a_n t_n=0$$

$$\frac{\partial L}{\partial \xi_n}=0 \quad \Rightarrow a_n=C-\mu_n$$

2022/10/24
智能与多媒体实验室
57

用于构造生成式模型的核函数

用核函数构造生成式模型

- 有类核函数，称为平滑核函数，可用于非参数密度估计。
  - 这可用于无监督密度估计，p(x)
  - 还可估计生成模型p(y，x)，用以分类和回归

平滑核函数

- 平滑核函数是单参数函数，且满足以下性质：

$$\int k(x)dx = 1 \qquad \int xk(x)dx = 0 \qquad \int x^2 k(x)dx \succ 0$$

- 高斯核函数是平滑核函数的简单例子：

$$k(x) = \frac{1}{(2\pi)^{\frac{1}{2}}} e^{-x^2/2}$$

- 可以通过引入带宽参数h来控制核函数的宽度：

$$k_h(x) = \frac{1}{h} k(\frac{x}{h})$$

- 可以通过定义RBF核来推广到向量值输入：

$$k_h(\boldsymbol{x}) = k_h(\|\boldsymbol{\omega}\|)$$

  - 高斯核情况下：

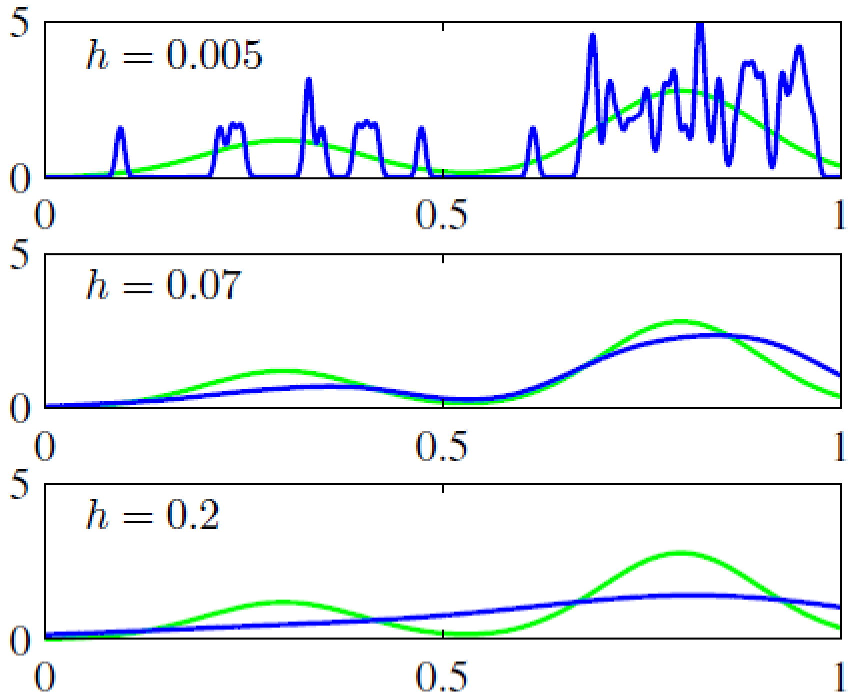
$$k_h(x) = \frac{1}{h^D (2\pi)^{\frac{D}{2}}} \prod_i^D \exp\left(-\frac{1}{2h^2} x_i^2\right)$$

平滑密度分布举例

- 选一平滑核函数

$$p(x)=\frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{1/2}}\exp\left\{\frac{\|x-x_n\|^2}{2h^2}\right\}$$

  - *参数 h 控制分布的平滑度*
  - *h的优化关联着模型的复杂性.*
    - ✓ *如果h太小 (上图， 蓝色曲线)，分布的噪声太大,*
    - ✓ *如果h太大 (下图),双峰性质被冲淡了*
    - ✓ *最好的分布应该是h处于某个中间值*



高斯混合模型

- 将N个高斯分布的平均作为分布的模型，用于估计密度分布

$$p(x \mid D) = \frac{1}{N} \sum_{1 \le n \le N} N(x \mid \mu_n, \sigma^2 \mathbf{I})$$

- 该模型使用的难点：需要给出分布中，簇的个数 N 和每个簇的*位置* $\boldsymbol{\mu}_n$

- 一种估计N与 μ_n 的方案：
  - *将每个样本数据点分别作为每个簇的中心* $\boldsymbol{\mu}_n = \mathbf{x}_n$

$$p(x \mid D) = \frac{1}{N} \sum_{1 \le n \le N} N(x \mid x_n, \sigma^2 \mathbf{I})$$

核密度估计 (KDE)

- 将高斯混合模型一般化：

$$\hat{p}(x) = \frac{1}{N} \sum_{1 \le n \le N} k_h(x - x_n)$$

- **这种方法就叫做核密度估计 (KDE)**
- 这种方法也叫 **Parzen 窗口密度估计**
- 这是一种简单的非参密度模型.

基于boxcar核的密度估计

- boxcar核函数：

$$k(x) = \mathbf{I}(|x| \le 1)$$

- 使用boxcar核函数做kde
  - 固定带宽，计算在多少样本数据点为中心的超立方体中有x点。

$$\hat{p}(x) = \frac{1}{N}\sum_{1\le n\le N} k_h(x - x_n) = \frac{1}{N}\sum_{1\le n\le N} \mathbf{I}(|x - x_n| \le h)$$

从核密度估计到K最近邻模型（1）

- 在基于boxcar核的kde中
  - 如果不固定带宽*h*，允许每个数据点的带宽或体积不同。

$$\hat{p}(x) = \frac{1}{N}\sum_{1\le n\le N} k_h(x - x_n) = \frac{1}{N}\sum_{1\le n\le N} \mathrm{I}(|x - x_n| \le h)$$

  - 增大x周围的体积，直到体积中有K个样本数据点，不管它们的类标签是什么
  - 生成的体积为V(x)（以前是h^D），设在这个体积中，有$N_c(x)$个样本属于c类。
- 基于这个思路，估计类条件密度：

$$p(x \mid y = c, D) = \frac{N_c(x)}{N_c V(x)}$$

- 其中，N_c 表示训练集中属于c类的样本个数

从核密度估计到K最近邻模型（2）

- 类先验分布，可以这样估计

$$p(y=c \mid D)=\frac{N_c}{N}$$

- 因此，类后验分布可以估计为

$$p(y=c \mid x, D)=\frac{\frac{N_c(x)}{N_c V(x)} \frac{N_c}{N}}{\sum_{c'} \frac{N_{c'}(x)}{N_c V(x)} \frac{N_{c'}}{N}}=\frac{N_c(x)}{\sum_{c'} N_{c'}}=\frac{N_c(x)}{K}$$

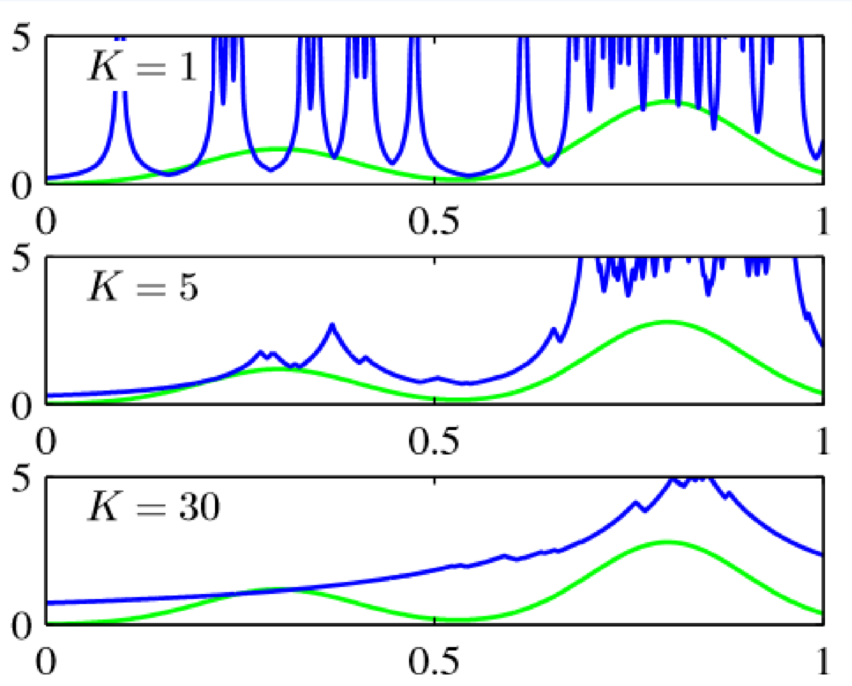
  - 这就是所谓的KNN密度估计算法
  - 这是推导KNN的另一种方法

KNN 密度估计举例

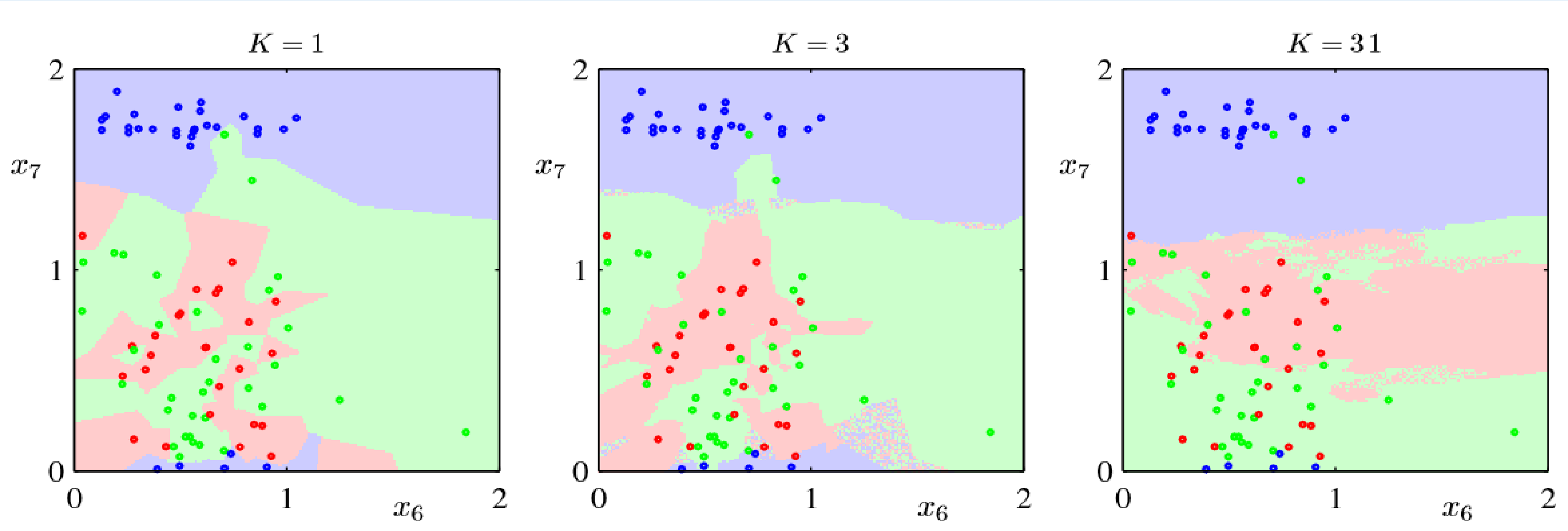
- 在KNN密度估计中，设 $N_c(x)$=K

$$p(x \mid y = c, D) = \frac{N_c(x)}{N_c V(x)}$$

- *K 控制着分布的平滑度*
  - ✓ *K 太小，导致密度模型噪声大 (如上图),*
  - ✓ *K太大，导致分布的双峰被平滑掉了 (如下图)*



基于 KNN 的分类举例



- K控制平滑度
- 太小的K，使每个类别形成许多小区域，
- 太大的K导致很少有较大区域。

核回归（1）

- 还可以将 KDE 用于回归问题
  - 目标是求一个条件均值

$$f(x) = E[y|x] = \int yp(y|x)dy = \frac{\int yp(x,y)dy}{\int p(x,y)dy}$$

  - 采用 KDE 来近似联合密度分布 p(x,y)：

$$p(x,y) \approx \frac{1}{N}\sum_{i=1}^{N} k_h(x-x_i)k_h(y-y_i)$$

$$f(x) = \frac{\frac{1}{N}\sum_{i=1}^{N} k_h(x-x_i)\int yk_h(y-y_i)dy}{\frac{1}{N}\sum_{i=1}^{N} k_h(x-x_i)\int k_h(y-y_i)dy}$$

核回归（2）

- 这里要用到平滑核的性质： $\int k_h(y - y_i)dy = 1 \qquad \int yk_h(y - y_i)dy = y_i$

- 因此，函数*f*(x)变成为：

$$f(x) = \frac{\sum_{i=1}^{N} k_h(x - x_i)y_i}{\sum_{i=1}^{N} k_h(x - x_i)}$$

- 设：

$$w_i(x) = \frac{k_h(x - x_i)}{\sum_{j=1}^{N} k_h(x - x_j)}$$

- 则可得到最终的回归函数：

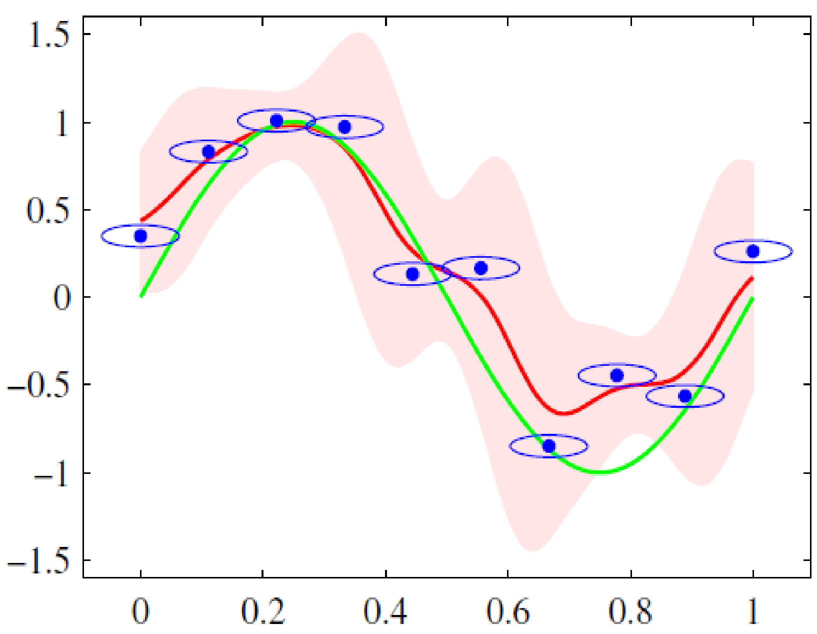
$$f(x) = \sum_{i=1}^{N} w_i(x)y_i$$

核平滑

- 从核回归的过程可以看出：
  - *核回归实际上只是训练数据输出值的加权平均*
  - *权重依赖于x 于训练数据的相似性*
- 因此，这种核回归方法又称为核平滑
  - *这种方法还称为：Nadaraya-Watson 模型.*

Nadaraya-Watson 模型

- *Nadaraya-Watson 核回归模型*
  - 蓝色点：样本数据集
  - 绿色曲线: 原始正弦曲线
  - 红线 : 推导出的回归函数
- *红色阴影：两个标准差区域*
- *蓝色椭圆：核函数的一个标准差等值线。*



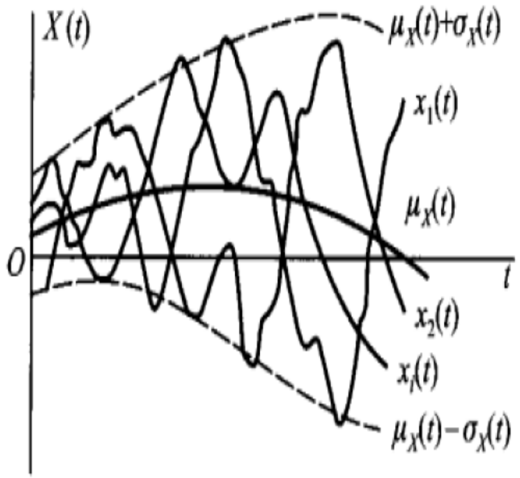
高斯过程

问题的提出

- 我们有一组观测值(xi,yi) i=1,…,M，M>0。希望估计未知函数f，满足 yi=f(xi)
- 一种解决思路
  - 假设 未知函数 f 受噪声影响。
  - 假设一组观测集上的函数值 f = { $f(x_1)$,…,$f(x_M)$ }，服从高斯分布，
    - ❿ 均值**μ**= (m(x1),…,m(xM))，m是平均函数
    - ❿ 协方差$\sum_{ij}=K(xi, xj)$，K是正定（Mercer）核。
  - 假设 M=N+1，（N：训练数据个数: $x_1,\ldots,x_M$，1个测试数据: x_*）
  - 运用联合高斯分布$p(f(x_1),\ldots,f(x_N),f(x_*))$，可以根据$f(x_1),\ldots, f(x_N)$ 的信息推断$f(x_*)$

高斯过程 (GP)

- 随机过程：
  - 随机变量集合{ f(t)：t∈ T}，
  - 它可被看成是一个无限维向量
- 高斯过程是一个随机过程
  - 随机变量的任意有限子集合
  - 这些变量服从多元高斯分布



高斯过程的定义

- 随机过程 $\{f(t) : t \in \mathrm{T}\}$,
- 如果对于任意有限集合 $t_1, \dots, t_m \in \mathrm{T}$，满足:
  - $f(t_i) \sim N(m(t_i), k(t_i, t_i))$ for any i
  - $t_1, \dots, t_m \in \mathrm{T}$ 的联合分布：

$$\begin{pmatrix} f(t_1) \\ \vdots \\ f(t_m) \end{pmatrix} \sim N\left(\begin{pmatrix} m(t_1) \\ \vdots \\ m(t_n) \end{pmatrix}, \begin{pmatrix} k(t_1, t_1) & \cdots & k(t_1, t_m) \\ \vdots & \ddots & \vdots \\ k(x_m, t_1) & \cdots & k(t_m, t_m) \end{pmatrix}\right)$$

- 则称$f(\cdot)$ 是一个高斯过程，表示为：
  - $f(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$
  - $m(t) = E[x(t)]$, $k(t, t') = E[(f(t) - m(t))(f(t') - m(t'))]$

高斯过程参数要求

- 什么类型的函数 m(·) 和 k(·, ·) 能生成高斯过程呢?
  - m(·)：任何实数值函数
  - k(·, ·)：对任意一组 $t_1, . . . , t_m \in X$，满足：

$$K = \begin{pmatrix} k(t_1, t_1) & \cdots & k(t_1, t_m) \\ \vdots & \ddots & \vdots \\ k(t_m, t_1) & \cdots & k(t_m, t_m) \end{pmatrix}$$

    - **K** 为半正定矩阵

基于GP的回归

- 设回归函数的先验分布满足高斯分布(GP)
  - $f(x) \sim GP(m(x), \kappa(x, x))$.
  - m(x),是均值函数； κ(x, x)是核函数或者协方差函数
    - $m(x) = E[f(x)]$
    - $k(x, x) = E(f(x) - m(x))(f(x) - m(x))^T$
  - 要求k() = 为半正定核函数,
- 对于任意有限个数据点，GP过程定义了一个联合高斯分布

$$p(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}(\mathbf{f} \mid \boldsymbol{\mu}, \mathbf{K})$$

  - 这里，$K_{ij} = \kappa(x_i, x_j)$，$\mu = (m(x_1),\ldots,m(x_m))$

基于无噪声观测值的预测

- 已知：
  - 训练集 $D = \{(x_i\,,\,f_i),\ i=1\sim N\}$
  - $f_i = f(x_i)$, 是函数f 在x_i 处的无噪声观测值
- 问题
  - 给定测试集 $\mathbf{X}_*$，预测函数在对应点的输出$\mathbf{f}_*$,
- 求解
  - 基于GP预测$f(x)$在x的值，根据条件，希望GP返回确定性的答案
  - 根据GP定义，有下面联合分布：

$$\begin{pmatrix} f \\ f_* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{bmatrix} \mathrm{K} & \mathrm{K}_* \\ \mathrm{K}_*^{\mathrm{T}} & \mathrm{K}_{**} \end{bmatrix}\right)$$

- 其中：
  - $\mathbf{K} = k(\mathrm{X},\mathrm{X})$ 为$N \times N$维
  - $\mathbf{K}_* = k(\mathrm{X},\mathrm{X}_*)$ 为$N \times N_*$维
  - $\mathbf{K}_{**} = k(\mathrm{X}_*,\mathrm{X}_*)$ 为$N_* \times N_*$维

预测过程

❖ 根据高斯模型规则，从联合分布可以得到后验分布：

$$\begin{pmatrix} f \\ f_* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^{\mathrm{T}} & \mathbf{K}_{**} \end{bmatrix}\right) \longrightarrow$$

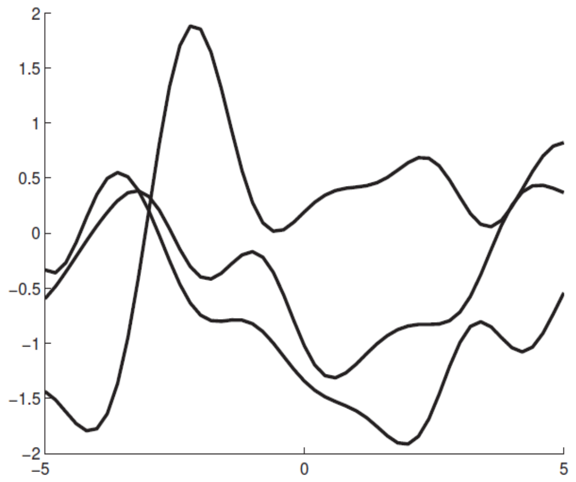
$$p(\mathbf{f}_* \mid \mathbf{X}_*, \mathbf{X}, \mathbf{f}) = \mathcal{N}(\mathbf{f}_* \mid \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

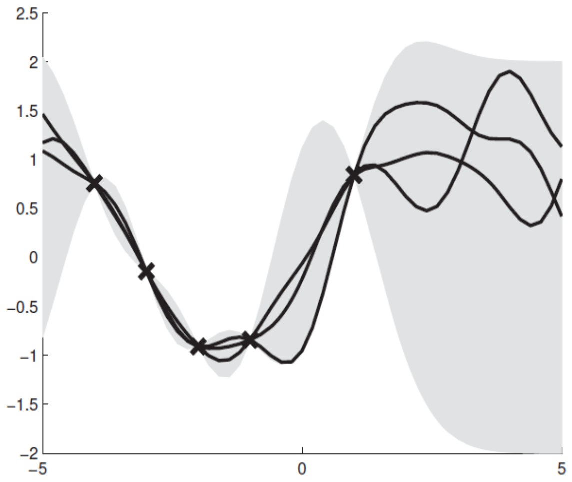
$$\boldsymbol{\mu}_* = \boldsymbol{\mu}(\mathbf{X}_*) + \mathbf{K}_*^T\mathbf{K}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{X}))$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T\mathbf{K}^{-1}\mathbf{K}_*$$

❖ 如果采用平方指数核函数，有下面例子：

- 左：从GP先验采样的一些函数
  - ❿ 基于平方指数核函数。
- 右图：从GP后验采样的样本
  - ❿ 基于5次无噪声观测值
  - ❿ 阴影区域表示E[f(x)]±2std(f(x”))





基于平方指数核的高斯过程波动性

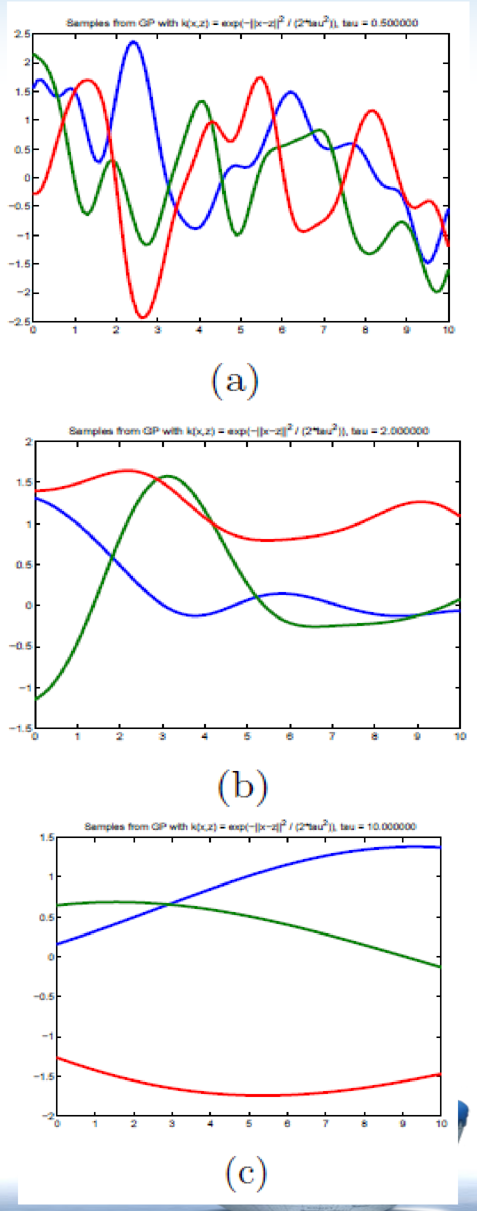
- 考虑一个简单的0均值高斯过程:
  - $f(\cdot) \sim GP(0, k(\cdot, \cdot))$
  - $f : T \rightarrow R$， $T = R$.
  - $k(\cdot, \cdot)$: 平方指数核函数

$$k(t, t') = \exp\left(-\frac{\left\|t - t'\right\|^2}{2\beta^2}\right)$$

(a) β = 0.5

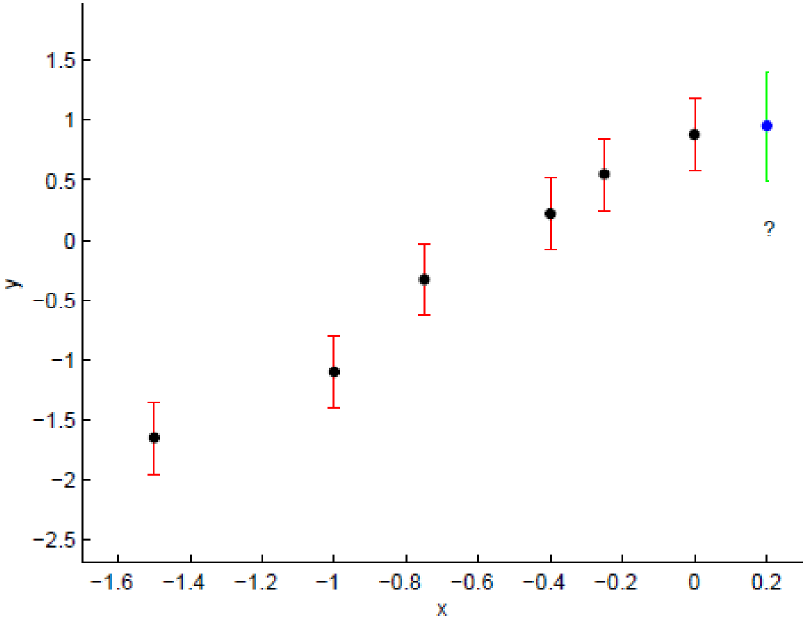
(b) β = 2

(c) β = 10.



基于噪声观测值的预测

- 假设：观测值有噪声
  - 设：$y_i = f(x_i) + \varepsilon_i$，这里：$\varepsilon \sim N(0, \sigma_y^2)$
- 模型必须要接近观测数据
  - $\mathrm{cov}[y_i, y_j] = \mathrm{cov}[f_i, f_j] + \mathrm{cov}[\varepsilon_i, \varepsilon_j] = \kappa(x_i, x_j) + \sigma_y^2 \delta_{ij}$,
    - 这里：$\delta_{ij} = \mathbf{I}(i=j)$,
  - 协方差也可表示为：$\mathrm{cov}[\mathbf{y}|\mathbf{X}] = \mathbf{K}_{XX} + \sigma_y^2 \mathbf{I}_N = \mathbf{K}_\sigma$



- 求解

$$\begin{pmatrix} y \\ f_* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_X \\ \mu_* \end{pmatrix}, \begin{bmatrix} \mathbf{K}_\sigma & \mathbf{K}_{X,*} \\ \mathbf{K}_{X,*}^T & \mathbf{K}_{**} \end{bmatrix}\right)$$

$$p(\mathbf{f}_* \mid \mathbf{X}_*, D) = N(\mathbf{f}_* \mid \boldsymbol{\mu}_{*|X}, \Sigma_{*|X})$$

$$\boldsymbol{\mu}_{*|X} = \boldsymbol{\mu}_* + \mathbf{K}_{X,*}^T \mathbf{K}_\sigma^{-1}(y - \boldsymbol{\mu}_X)$$

$$\Sigma_{*|X} = \mathbf{K}_{**} - \mathbf{K}_{X,*}^T \mathbf{K}_\sigma^{-1} \mathbf{K}_{X,*}$$

基于有噪声观测值的预测结果