

SVM

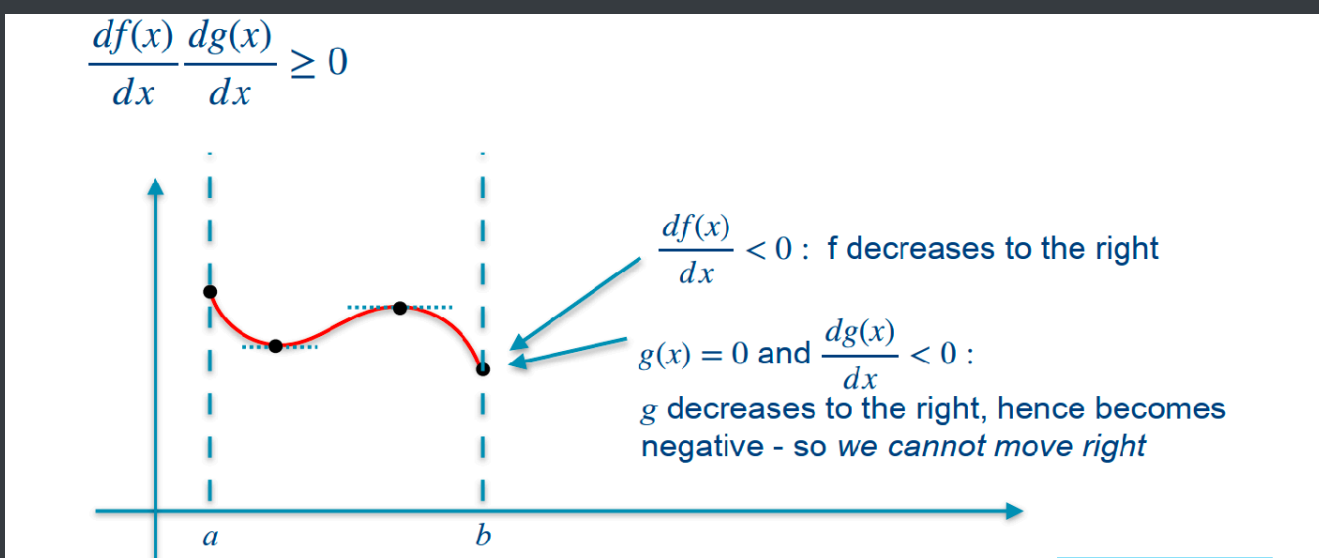
mathematical background --- the KKT conditions

minimization 问题和 maximization 本质相同，知道 minimization 的解法就可以通过 minimize $-f$ 的方法求出 maximize f ，所以这里只讨论 minimize；

- basics

目标点 either 在极值点，or 在边界点；

判断边界点是不是 minima candidate 的方法：



f 和 g 变大的方向一致！ (moving into admissible region increases f)

- lagrange multipliers

- under equality constraint

To optimize $f(x)$ under constraint $g(x) = 0$:

find stationary points of $L(x, \lambda) = f(x) - \lambda g(x)$

Stationary point : all partial derivatives = 0

$$\nabla_x L(x, \lambda) = 0 \Rightarrow \nabla_x f(x) - \lambda \nabla_x g(x) = 0 \Rightarrow \nabla_x f(x) = \lambda \nabla_x g(x)$$

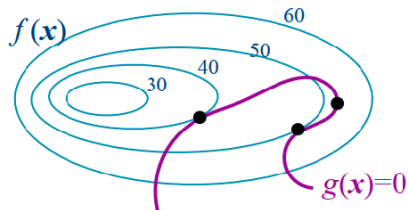
$$\frac{\partial L(x, \lambda)}{\partial \lambda} = 0 \Rightarrow g(x) = 0$$

x is "on the curve"

gradients of f and g in x are parallel

contour lines of f and g in x are parallel

$g(x)=0$ runs momentarily *along* a contour line of f



KILL FIFTEEN

点在 $g(x)=0$ 曲线上，且 f 和 g 的梯度平行（梯度是等高线的正交方向）

2. generalizing to inequalities

$$\nabla_x L(x, \lambda) = 0, \text{ which implies } \nabla_x f(x) = \lambda \nabla_x g(x)$$

On the border ($g(x) = 0$),
we require $\lambda \geq 0$

Inside the admissible area ($g(x) \neq 0$),
we require $\lambda = 0$ so that $\nabla_x f(x) = 0$

边界上 $g(x) = 0$ ，需要 f 和 g 增大方向一致，因此需要 $\lambda \geq 0$ ，

-> constraint is active

非边界上 $g(x) \neq 0$ ，需要 $f(x)$ 导数为0，则必须 $\lambda = 0$ ；

-> constraint is inactive

These two conditions are summarized as $\lambda \geq 0$ and $\lambda \cdot g(x) = 0$

■ KKT条件

To minimize $f(\mathbf{x})$ under constraints $g_i(\mathbf{x}) \geq 0$ and $h_i(\mathbf{x})=0$:

consider $L(\mathbf{x}, \lambda, \nu) = f(\mathbf{x}) - \sum_i \lambda_i g_i(\mathbf{x}) - \sum_i \nu_i h_i(\mathbf{x})$

Local minima are characterized by

$$\nabla_{\mathbf{x}} f(\mathbf{x}) - \sum_i \lambda_i \nabla_{\mathbf{x}} g_i(\mathbf{x}) - \sum_i \nu_i \nabla_{\mathbf{x}} h_i(\mathbf{x}) = 0$$

$$\forall i : h_i(\mathbf{x}) = 0$$

$$\forall i : g_i(\mathbf{x}) \geq 0$$

$$\forall i : \lambda_i \geq 0$$

$$\forall i : \lambda_i g_i(\mathbf{x}) = 0$$

In admissible region

If on border, moving into admissible region increases f

These are known as the **Karush-Kuhn-Tucker (KKT) conditions**.

Actually, one version. KKT are often derived for $g(\mathbf{x}) \leq 0$ instead of ≥ 0 , and $+\sum$ instead of $-\sum$ in L . Equivalent, but "our" version is more in line with SVM derivations, see later.

"complementary slackness": either we're on the border ($g_i(\mathbf{x})=0$, g_i is active) or we're not ($\lambda_i=0$, g_i is inactive)

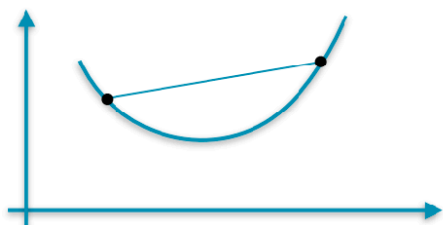
mathematical background --- duality

motivation: global minimum vs. local minima

凸优化问题：函数（function）和定义域（admissible region）都是凸的

1. $\forall \mathbf{x}, \mathbf{y} \in \text{Adm}, \forall \alpha \in [0,1] : f(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y})$
2. $\forall \mathbf{x}, \mathbf{y} \in \text{Adm}, \forall \alpha \in [0,1] : \alpha \mathbf{x} + (1 - \alpha) \mathbf{y} \in \text{Adm}$

1: linear interpolation never underestimates f

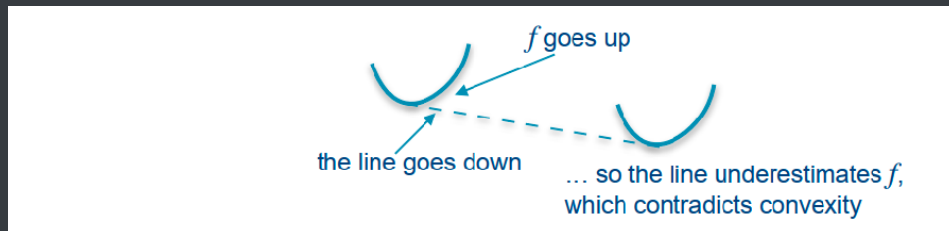


2: The admissible region is convex



- f 是凸的，则 local minimum 就是 global minimum

证明（反证法）：如果存在两个不同的 local minima，与 f 是凸的 矛盾；



- 如果不是凸函数，solution是找一个等价的凸优化问题；

双重问题

- dual function

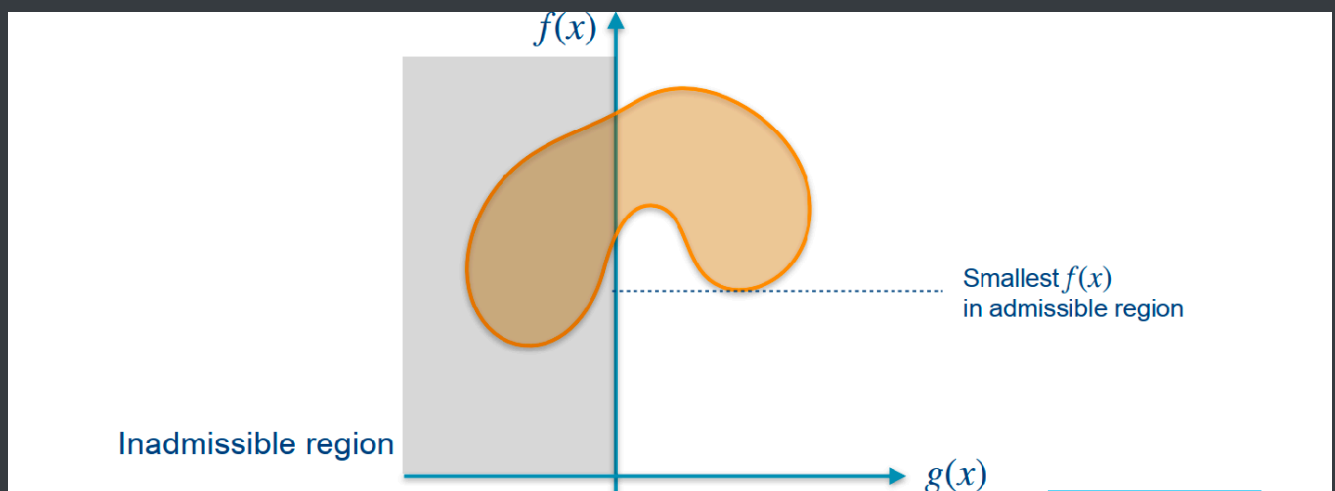
$$\tilde{f}(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$$

- property: when $f(x)$ is convex

$$\min_{x \in \text{Adm}} f(x) = \max_{\lambda \geq 0, \nu} \tilde{f}(\lambda, \nu) \text{ (where } \lambda \geq 0 \text{ means } \forall i : \lambda_i \geq 0)$$

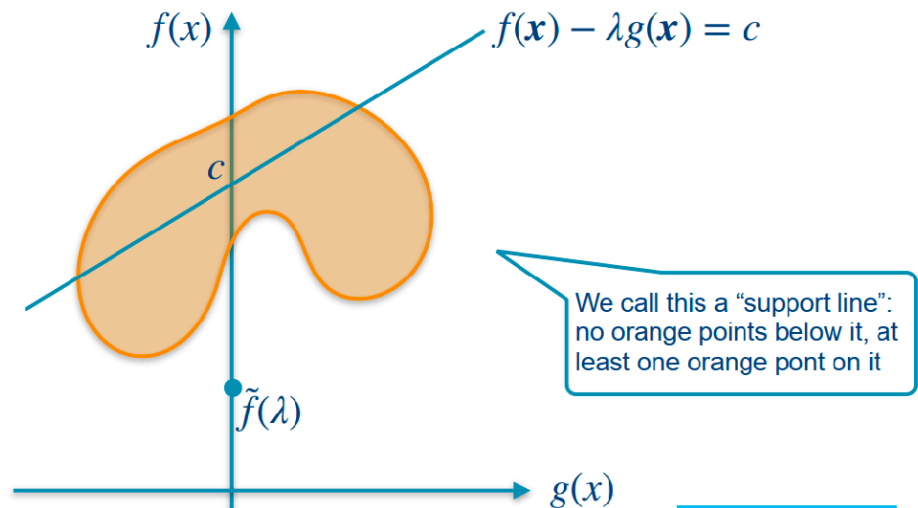
证明是非平凡的，直觉说明：

画出 $f(x) \sim g(x)$ ，则有：



将拉格朗日乘子法表示出来：

- $\tilde{f}(\lambda) = \inf_x L(\mathbf{x}, \lambda) = \inf_x (f(\mathbf{x}) - \lambda g(\mathbf{x}))$ is the smallest c obtainable for any \mathbf{x} , given λ

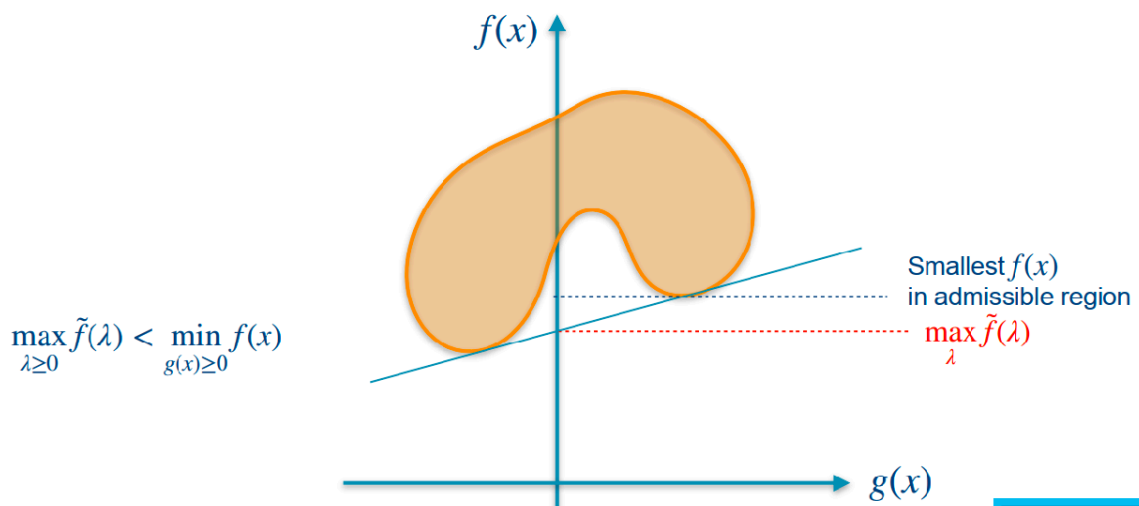


则给定任意 λ , 这条直线的斜率一定, 将其移动到最下面 (与区域相切) 即可获得 dual function 的值 (取 x 使其到 \inf 下界) ;

然后在 $\lambda \geq 0$ 的限制下调整斜率, 使其取到 \max 的 dual function 值;

在图中非凸的情况下, 会出现 dual function 的 \max 值与要求的极小值不相等:

The highest intercept can be strictly smaller than $\min_{g(x) \geq 0} f(x)$



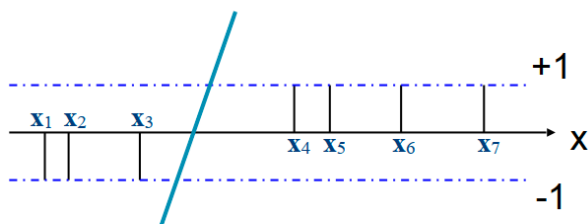
linear SVMs

motivation: maximum **margin**, determining points are **support vectors**

margin: 分割平面离最近点的距离

转化成凸优化问题：

Illustration for 1-D inputs:



Given w, b , the “hyperplane” in 1-D space is the point x for which $w \cdot x + b = 0$

Many w, b fulfill $\forall i : y_i(w \cdot x_i + b) \geq 1$

Only one solution with smallest possible slope $\|w\|$. For that solution, the margin is maximal (separator is right in the middle between x_3 and x_4)

这里 x_3 和 x_4 就是support vectors；

minimize $f(w, b) = \|w\|^2/2$ under the constraints $y_i(w x_i + b) \geq 1$ for $i=1, \dots, N$ (with N the size of the dataset)

目标函数是凸的，定义域也是凸的；

如何计算？

$$\nabla_{w,b}(\|w\|^2/2) - \sum_i \lambda_i \nabla_{w,b}((y_i(w x_i + b) - 1)) = 0$$

$$\text{Zeroing the gradient to } w \text{ gives: } w - \sum_i \lambda_i y_i x_i = 0$$

$$\text{and hence: } w = \sum_i \lambda_i y_i x_i$$

$$\text{Zeroing the partial derivative to } b \text{ gives, in addition: } \sum_i \lambda_i y_i = 0$$

- b can be computed from any single support vector. E.g., if x_i is a positive support vector: $w x_i + b = 1$, so $b = 1 - w x_i$
- For reasons of numerical stability, one can do this for all support vectors and average the results:

$$b = 1 - \frac{\sum_{x_i \in SV} w x_i}{|SV|} \quad \text{with } SV \text{ the set of support vectors}$$

所以得到计算 w 和 b 的方程如下：

$$\mathbf{w} = \sum_i \lambda_i y_i \mathbf{x}_i = \sum_{\mathbf{x}_i \in SV} \lambda_i y_i \mathbf{x}_i \qquad b = 1 - \frac{\sum_{\mathbf{x}_i \in SV} \mathbf{w} \mathbf{x}_i}{|SV|}$$

注意w的公式中只需要对作为support vector的x进行计算，因为其他x对应的lambda是0；

现在需要解lambda，用dual function：

求 f(w) 的min值，对应的拉格朗日乘子为

$$L(\mathbf{w}, \lambda) = \|\mathbf{w}\|^2/2 - \sum_i \lambda_i (y_i (\mathbf{w} \mathbf{x}_i + b) - 1)$$

前面已经对w求导，带入得dual function：

Fill in $\mathbf{w} = \sum_i \lambda_i y_i \mathbf{x}_i$ and write as a function \tilde{f} of λ :

$$\begin{aligned} \tilde{f}(\lambda) &= \frac{1}{2} \left(\sum_i \lambda_i y_i \mathbf{x}_i \right) \left(\sum_j \lambda_j y_j \mathbf{x}_j \right) - \sum_i \lambda_i \left(y_i \left(\sum_j \lambda_j y_j \mathbf{x}_j \mathbf{x}_i + b \right) - 1 \right) \\ &= \frac{1}{2} \left(\sum_i \sum_j \lambda_i \lambda_j y_i y_j \mathbf{x}_i \mathbf{x}_j \right) - \sum_i \lambda_i y_i \left(\sum_j \lambda_j y_j \mathbf{x}_j \mathbf{x}_i + b \right) + \sum_i \lambda_i \\ &= \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \mathbf{x}_i \mathbf{x}_j - \sum_i \sum_j \lambda_i \lambda_j y_i y_j \mathbf{x}_i \mathbf{x}_j - b \sum_i \lambda_i y_i + \sum_i \lambda_i \\ &= \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \mathbf{x}_i \mathbf{x}_j \end{aligned} \qquad \sum_i \lambda_i y_i = 0$$

只需要求dual function的最大值，得到lambda；

what vectors tend to become support vectors, if we maximize this formula?

by assigning high non-zero lambda values to similar x's (high $\mathbf{x}_i \cdot \mathbf{x}_j$) with opposite y values

what if there's no solution?

slack variables & cost function:

- Solution: soften the constraints using *slack variables* :

$$\text{Minimize } \|\mathbf{w}\|^2/2 + C \left(\sum_i \xi_i \right) \text{ under constraints } (\mathbf{w}\mathbf{x}_i + b) \cdot \mathbf{y}_i \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

allow some points in or at the wrong side of the margin, at a (minimized) cost

variant: support vector regression (not only for binary results)

- Simply minimize $\|\mathbf{w}\|^2/2$ under constraints

$$\bullet y_i - (\mathbf{w}\mathbf{x}_i + b) \leq \varepsilon$$

$$\bullet (\mathbf{w}\mathbf{x}_i + b) - y_i \leq \varepsilon$$

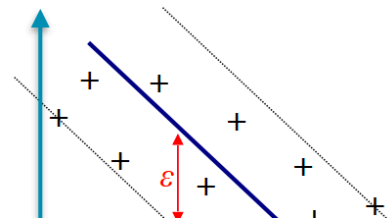
All y_i must be within ε from the prediction

- Again, we can use slack variables if a solution may not exist:

$$\text{minimize } \|\mathbf{w}\|^2/2 + C \sum_i (\xi_i + \xi_i^*) \text{ subject to}$$

$$\bullet y_i - (\mathbf{w}\mathbf{x}_i + b) \leq \varepsilon + \xi_i$$

$$\bullet (\mathbf{w}\mathbf{x}_i + b) - y_i \leq \varepsilon + \xi_i^*$$



variant: least squares SVMs

解凸优化问题太麻烦，还是换成equality限制条件下的优化问题

$$\begin{aligned} &\text{minimize } \|\mathbf{w}\|^2 + C \sum_{i=1}^N e_i^2 \\ &\text{subject to } y_i(\mathbf{w}\mathbf{x}_i + b) = 1 - e_i \end{aligned}$$

kernels & non-linear SVMs

for the data that are not linearly separable, transform data into a **high-dimensional space** and learn an SVM in that space

用一个函数 phi 把样本点转换到高维空间：

- That means we want to maximize

$$\sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) \quad \text{with } \sum_i \lambda_i y_i = 0 \text{ and } \forall i : \lambda_i \geq 0$$

注意到用phi转化后两个样本点还是点乘关系，所以直接抽象成一个函数：

$$\text{maximize } \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad \text{with } \sum_i \lambda_i y_i = 0 \text{ and } \forall i : \lambda_i \geq 0$$

K --- kernel function

如何构造核函数？

能表示转化空间点乘的充要条件：K必须对任意数据集满足 **对称 & 半正定**！！

如何应用核函数？

尝试经典核函数，调参（cross validation / error bound）；

$$\bullet \quad \text{err}(f) \leq \text{err}(f, T) + (\text{some function of VC-dimension})$$

VC-dimension小可能导致bias，VC-dimension大可能导致overfitting

kernel trick

任意输入向量求点积的场景都可以用kernel trick，在SVMs的场景中可以理解为求两个向量的 similarity

kernel也是一个降维方法：如果样本量是N，那么kernel matrix的大小永远是 N^2 ，如果样本量不大维度很高，则降低复杂度的效果非常显著；

对于non-vectorial的输入，应用kernel计算similarity可以理解为隐式地定义一个特征空间，我们不知道这个特征是什么；

SUMMARY

- SVMs vs. instance based learning
 - SVMs only stores support vectors,
 - IBL stores all examples

