SVM

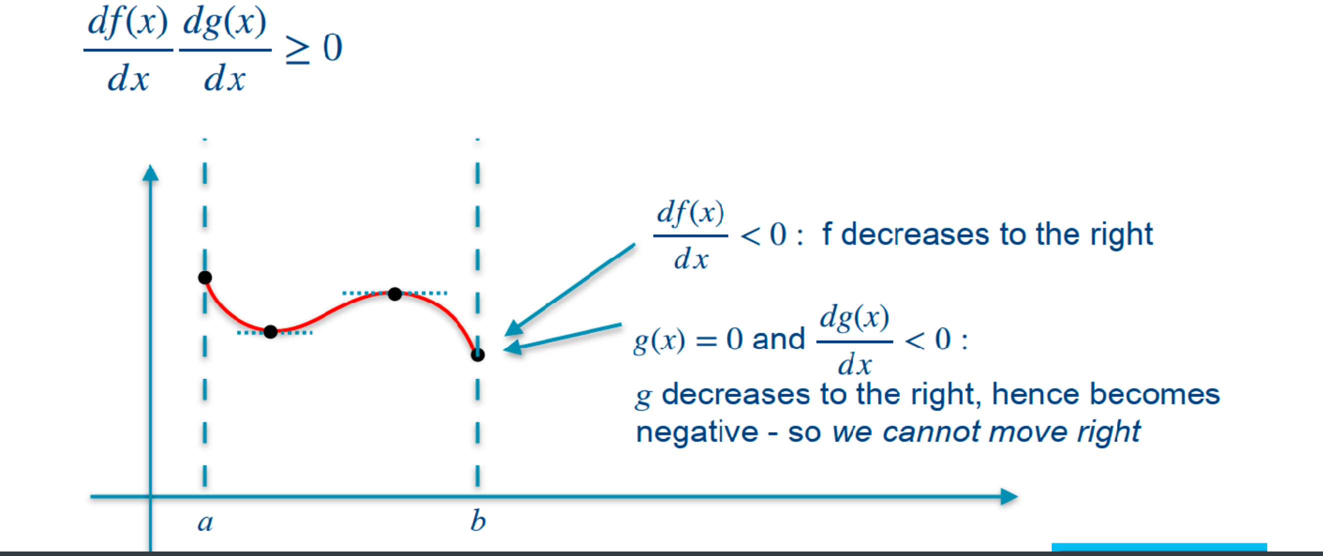
mathematical background --- the KKT conditions

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；

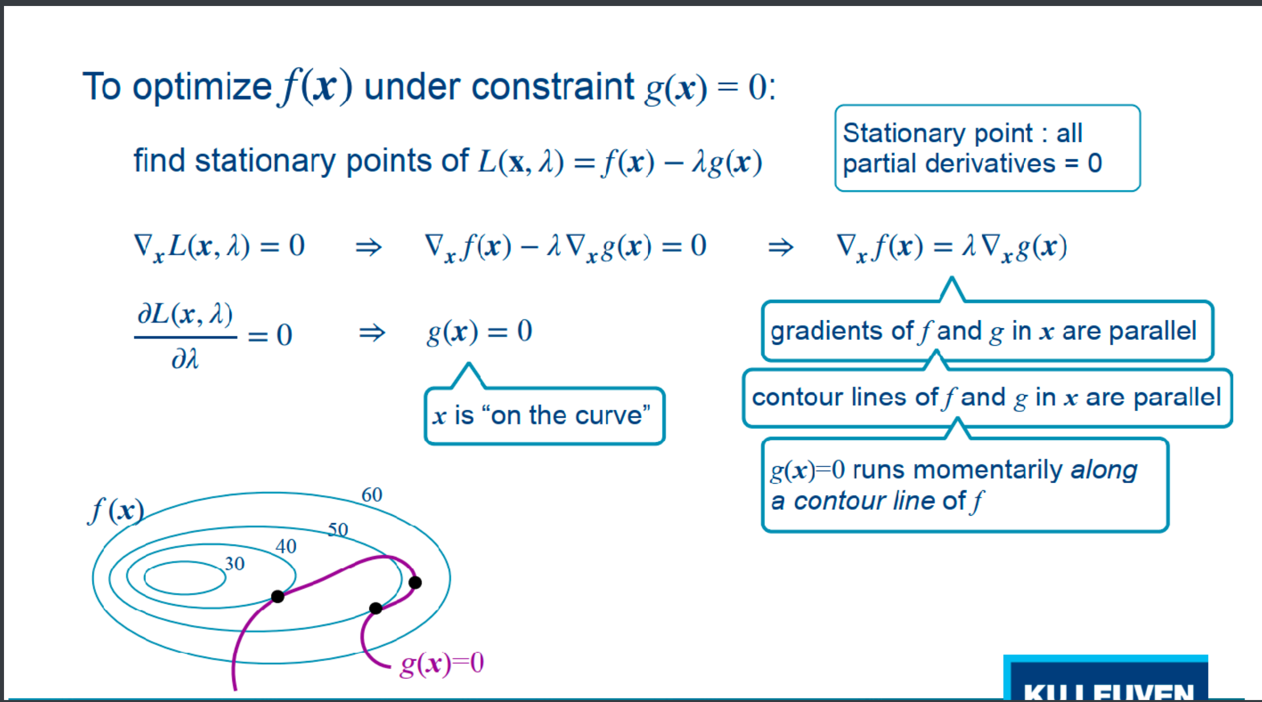
  判断边界点是不是minima candicate的方法：

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

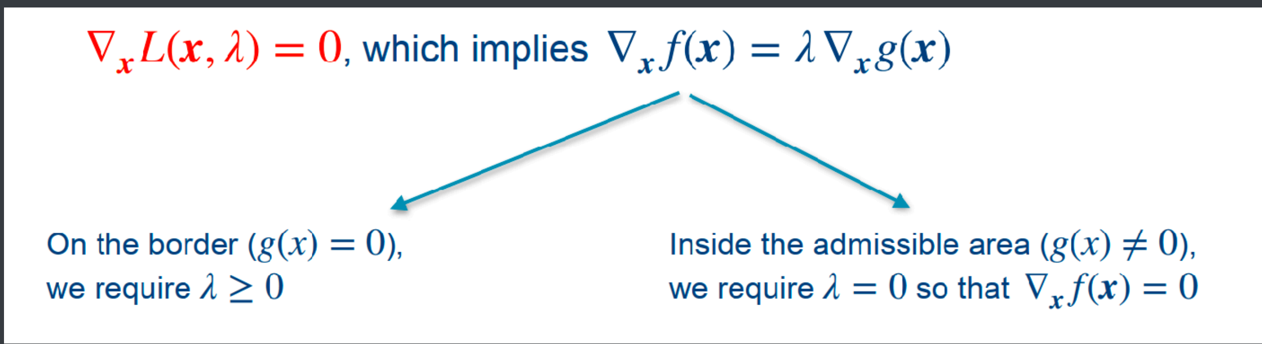
- lagrange multipliers

  1. under equality constraint



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



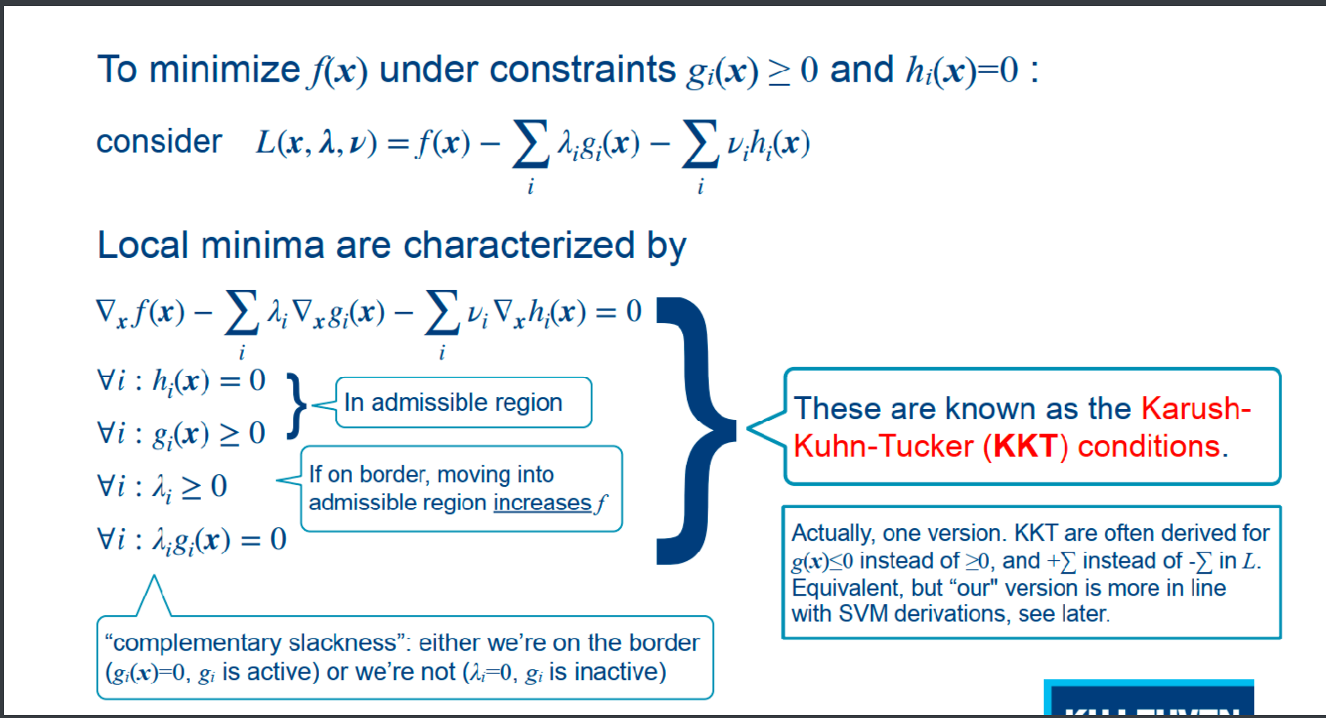
边界上 g(x) = 0，需要 f 和 g 增大方向一致，因此需要 lambda >= 0,

-> constraint is active

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

-> constraint is inactive

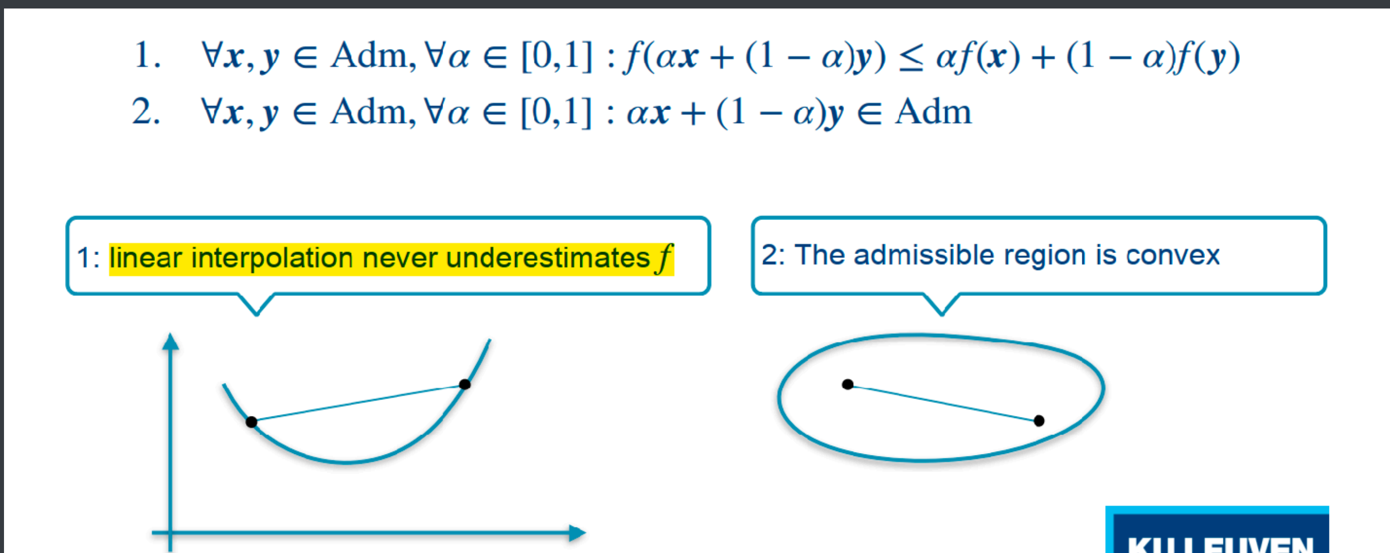
- KKT条件



mathematical background --- duality

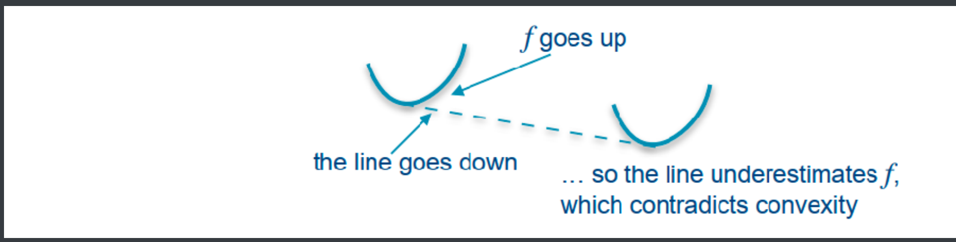
motivation: global minimum vs. lcoal minima

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



- 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

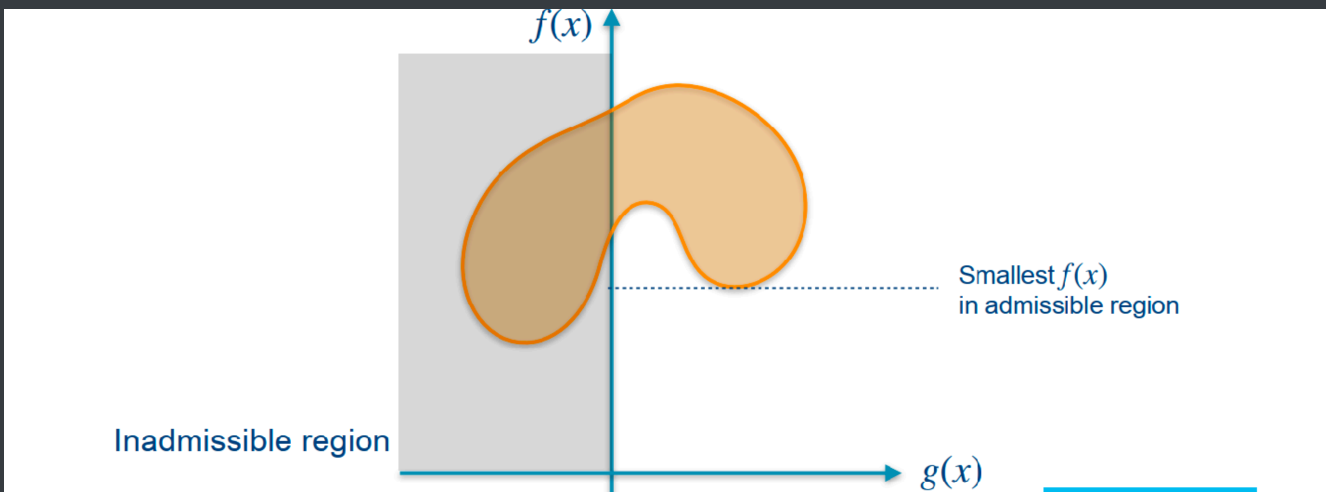
$$\tilde{f}(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$$

- property: when f(x) is convex

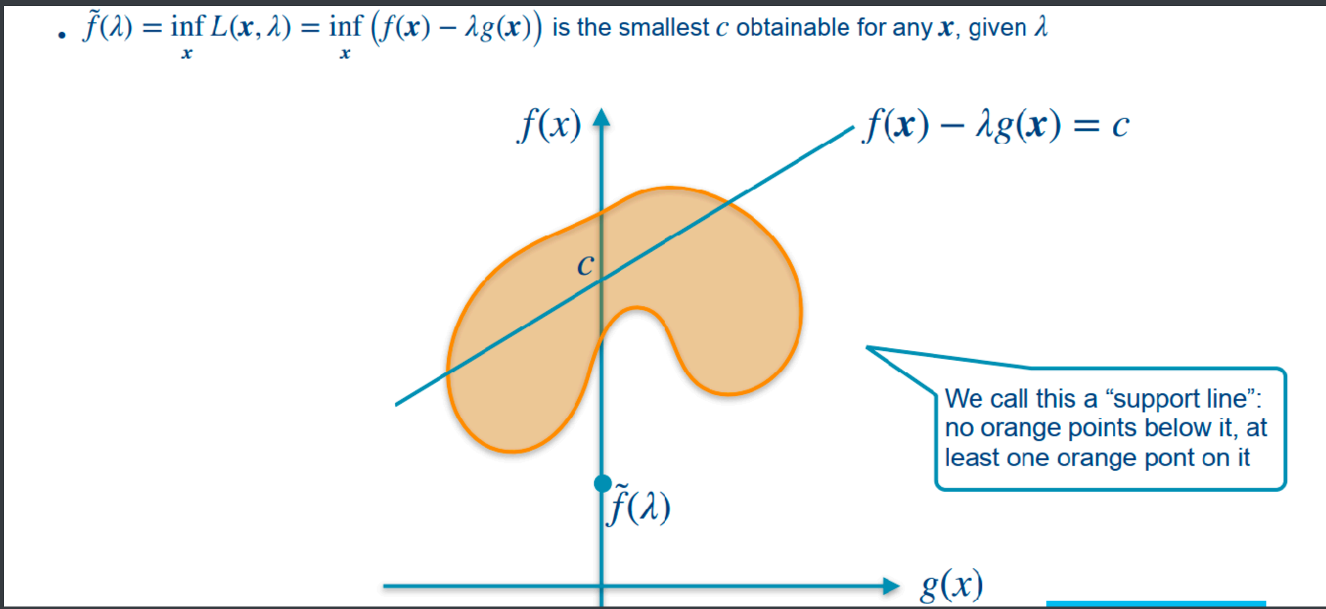
$$\min_{x \in Adm} f(x) = \max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\nu}} \tilde{f}(\boldsymbol{\lambda}, \boldsymbol{\nu}) \text{ (where } \boldsymbol{\lambda} \geq \mathbf{0} \text{ means } \forall i : \lambda_i \geq 0)$$

证明是非平凡的，直觉说明：

画出f(x) ~ g(x)，则有：



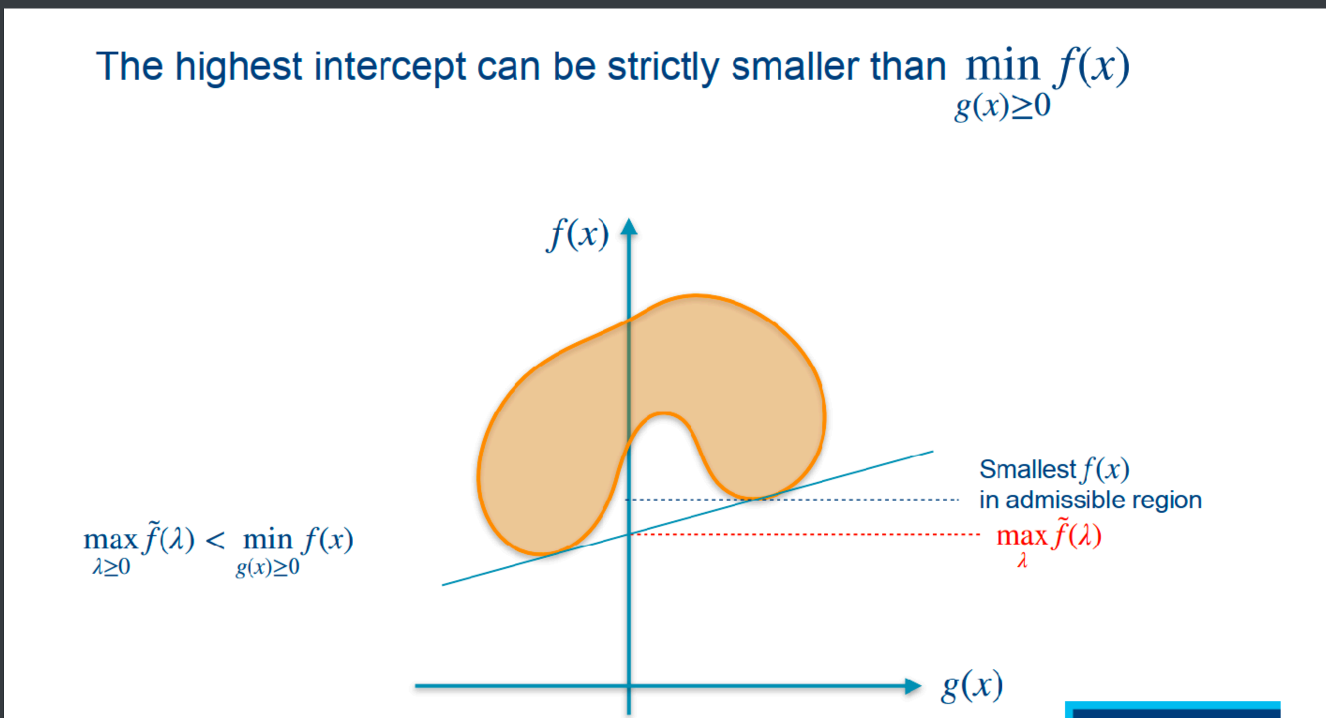
将拉格朗日乘子法表示出来：



则给定任意lambda，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 >= 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：



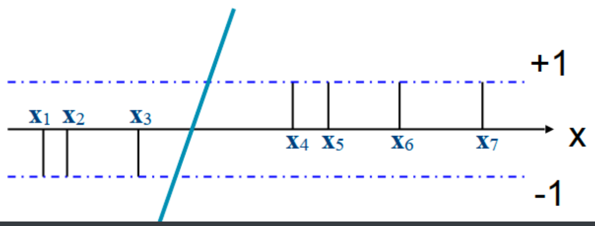
linear SVMs

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 $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$

Many w, b fulfill $\forall i : y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1$

Only one solution with smallest possible slope $\|\boldsymbol{w}\|$. 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 $f(\boldsymbol{w}, b) = \|\boldsymbol{w}\|^2/2$ under the constraints $y_i(\boldsymbol{w}\boldsymbol{x}_i + b) \geq 1$ for i=1, … 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？

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

Zeroing the gradient to w gives: $\boldsymbol{w} - \sum_i \lambda_i y_i \boldsymbol{x}_i = 0$

and hence: $\boldsymbol{w} = \sum_i \lambda_i y_i \boldsymbol{x}_i$

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

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

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

所以得到计算w和b的方程如下：

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

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 $L(w, \lambda) = \|w\|^2/2 - \sum_i \lambda_i (y_i (w x_i + b) - 1)$

前面已经对w求导，带入得dual function：

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

$$\begin{aligned}
\tilde{f}(\lambda) &= \frac{1}{2}(\sum_i \lambda_i y_i x_i))(\sum_j \lambda_j y_j x_j)) - \sum_i \lambda_i \big(y_i (\sum_j \lambda_j y_j x_j x_i + b) - 1\big) \\
&= \frac{1}{2}(\sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i x_j) - \sum_i \lambda_i y_i (\sum_j \lambda_j y_j x_j x_i + b) + \sum_i \lambda_i \\
&= \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i x_j - \sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i x_j - b \sum_i \lambda_i y_i + \sum_i \lambda_i \\
&= \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i x_j
\end{aligned}$$

$$\sum_i \lambda_i y_i = 0$$

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 xi * xj) with opposite y values

<u>what if there's no solution?</u>

slack variables & cost function:

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

  Minimize $\|w\|^2/2 + C(\sum_i \xi_i)$ under constraints $(\boldsymbol{w}\boldsymbol{x}_i + b)\cdot y_i \geq 1 - \xi_i$ and $\xi_i \geq 0$

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

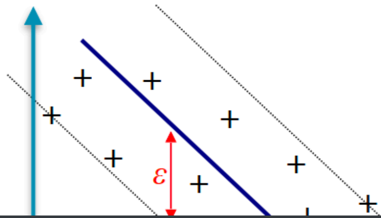
variant: support vector regression (not only for binary results)

- Simply minimize $\|w\|^2/2$ under constraints
  - $y_i - (\boldsymbol{w}\boldsymbol{x}_i + b) \leq \varepsilon$
  - $(\boldsymbol{w}\boldsymbol{x}_i + b) - y_i \leq \varepsilon$

All y_i must be within ε from the prediction

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

  minimize $\| w \|^2/2 + C\sum_i (\xi_i + \xi_i^*)$ subject to
  - $y_i - (\boldsymbol{w}\boldsymbol{x}_i + b) \leq \varepsilon + \xi_i$
  - $(\boldsymbol{w}\boldsymbol{x}_i + b) - y_i \leq \varepsilon + \xi_i^*$



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

$$\text{minimize } \|w\|^2 + C\sum_{i=1}^{N} e_i^2$$

$$\text{subject to } y_i(\boldsymbol{w}\boldsymbol{x}_i + b) = 1 - e_i$$

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

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

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：

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

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）；

- $\mathrm{err}(f) \leq \mathrm{err}(f,T)$ + (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-vectoria的输入，应用kernel计算similartiy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

- SVMs vs. neural networks

  SVMs gains expressiveness by kernels,

  ANN gains expressiveness by representation throughout its layers