

Support Vector Machine (SVM)

- Binary Classification with linear models
- $y \in \{-1, 1\}$ and $x \in \mathbb{R}^d$

Constraint Optimization

Lagrange function

- $L(x, \lambda) := f(x) + \lambda h(x)$
 - λ is *Lagrange Multiplier*

Equality Constraint

- $\min_x f(x) \quad s.t. \quad h(x) = 0$
 - \forall point x on the constraint surface $h(x) = 0$, we have $\nabla h(x)$ orthogonal to the surface
 - Because if $\nabla h(x)$ has tangent component, we can move along the direction to make $h(x) \neq 0$
 - For a local minimum x^* , $\nabla f(x^*)$ must be orthogonal to the surface
 - Generally, x^* is a local minimum $\Rightarrow \exists \lambda \quad s.t. \quad \nabla f(x^*) + \lambda \nabla h(x^*) = 0$
 - (There may be some corner cases not satisfying this equation)
 - 可以说, 极值点往往是限制平面与函数相切的点, 那么此时该点处平面的梯度与函数的梯度就应该是相反的
- When we have K constraints:
 - $\min_x f(x) \quad s.t. \quad h_i(x) = 0, i \in \{1, \dots, K\}$
 - then $L(x, \lambda) = f(x) + \sum_{i=1}^K \lambda_i h_i(x)$
- now we can say that:

$$x^* \text{ is a local minimum } \Rightarrow \exists \lambda \begin{cases} \nabla_x L(x^*, \lambda) = 0 & (1) \\ \nabla_\lambda L(x^*, \lambda) = 0 & (2) \end{cases}$$

- (1)说明 $\lambda f(x^*) + \sum_{i=1}^n \lambda_i \nabla h_i(x^*) = 0$
- (2)说明 $h_i(x^*) = 0, \quad \forall i$

Inequality Constraint

- $\min_x f(x) \quad s.t. \quad g(x) \leq 0$
 - \forall point x on the surface $g(x) = 0$, $\nabla g(x)$ must be orthogonal to the surface and *point out of* the region of $g(x) \leq 0$
 - For a local minimum x^*
 - the constraint is **active**:
 - if x^* on the surface $g(x) = 0$, then $-\nabla f(x^*)$ must have the same direction as $\nabla g(x^*)$
 - $\exists \mu > 0, \quad s.t. \quad \nabla f(x^*) + \mu \nabla g(x^*) = 0$
 - the constraint is **inactive**:
 - if x^* within the surface, we only need $\nabla f(x^*) = 0$
 - $\exists \mu = 0, \quad s.t. \quad \nabla f(x^*) + \mu \nabla g(x^*) = 0$
 - 显然地, 如果函数的最小值点在平面内部, 自然这个平面的约束就没有用了, 极值点就是这个最小值点; 而如果函数的最小值点在平面外, 那么就与equality constraint一样, 极值点在平面的边界上
 - 总结:

$$\exists \mu \geq 0, \quad s.t. \quad \nabla f(x^*) + \mu \nabla g(x^*) = 0 \begin{cases} \mu = 0 & \text{inactive} \\ \mu > 0 & \text{active} \end{cases}$$

K.K.T conditions

- $\min_x f(x) \quad s.t. \quad h_i(x) = 0, i \in \{1, \dots, K\}, \quad g_j(x) \leq 0, j \in \{1, \dots, L\}$
- $L(x, \lambda, \mu) = f(x) + \sum_{i=1}^K \lambda_i h_i(x) + \sum_{j=1}^L \mu_j g_j(x)$
- **K.K.T conditions:**

$$x^* \text{ is a local minimum} \Rightarrow \begin{cases} \nabla_x L(x^*, \lambda, \mu) = 0 \\ h_i(x^*) = 0 & \forall i \\ g_j(x^*) \leq 0 & \forall j \\ \mu_j \geq 0 & \forall j \\ \mu_j g_j(x^*) = 0 & \forall j \end{cases}$$

- 对于 $\mu_j g_j(x^*) = 0$, 若 x^* 在边界处, 显然有 $g_j(x^*) = 0$; 若 x^* 在内部, 则有 $\mu = 0$, 所以该式恒等于 0

Primal Form

- Maximize Margin Criterion**
 - minimum distance to the hyperplane $w^T x + b = 0$, over all training data
 - Structural Risk Minimization*
- x 到 $w^T x + b = 0$ 的距离为 $\Delta x = \frac{w^T x + b}{\|w\|}$
- we can define $\gamma_i := y_i \Delta x_i = \frac{y_i (w^T x_i + b)}{\|w\|} \geq 0$ (as $y_i \in \{-1, 1\}$)
- $\gamma = \min_{i=1, \dots, n} \gamma_i = \min_{i=1, \dots, n} \frac{y_i (w^T x_i + b)}{\|w\|}$
 - this is called **geometric margin**
- now we need to maximize this distance:

$$\max_{w, b, \gamma} \gamma \quad s.t. \frac{y_i (w^T x_i + b)}{\|w\|} \geq \gamma, \quad \forall i$$

- 但显然三个参数对于我们来说还是太多了, 所以我们可以先假设找到了离超平面 $w^T x + b = 0$ 最近的点 x_0 , 此时有 $\gamma_0 = \frac{y_0 (w^T x_0 + b)}{\|w\|} = \gamma$
- then we can update our object:

$$\max_{w, b} \frac{y_0 (w^T x_0 + b)}{\|w\|} \quad s.t. \quad y_i (w^T x_i + b) \geq y_0 (w^T x_0 + b), \quad \forall i$$

- 我们可以发现, $w^T x + b = 0$ 是一个超平面, 而 $k w^T x + k b = 0, \forall k$ 都是同一个超平面, 那么这样的话 $y_0 (w^T x_0 + b)$ 就可以是任意值 (which is called **functional margin**), 所以我们可以干脆令 $y_0 (w^T x_0 + b) = 1$, 背后的逻辑是不管真实的 γ 是多少, 我都可以找到一个 k 使上式等于 1 (同理, 不管上式等于多少, 我们也总能找到相应的 w 使得几何距离不变), 而且显然这个等式也是不影响 γ 的
- then we can say that:

$$\max_{w, b} \frac{1}{\|w\|} \quad s.t. \quad y_i (w^T x_i + b) \geq 1, \quad \forall i$$

- now we have the **primal form** of SVM:

$$\min_{w, b} \frac{1}{2} w^T w \quad s.t. \quad y_i (w^T x_i + b) \geq 1, \quad \forall i$$

- 我们管 x_0 这样的用来确定超平面的向量叫做**支持向量(support vector)**, 由此我们即可以看出支持向量机的命名缘由: 这些向量支撑了超平面的选择, 我们只需考虑这些支持向量, 而不用管别的向量, 就可以确定最终的超平面
- Convex Quadratic Programming
 - 这个函数是凸的, 所以总能找到一个最优解, 调包即可
 - Standard package: $O(d^3)$

Dual Form

- 2 problems for SVM:
 - may be not linearly separable
 - margin γ may be too small (there may be some outlier samples)
- 我们可以改写 primal form, 把条件写进函数中:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i (w^T x_i + b)) \quad (\alpha_i \geq 0)$$

$$p^* = \min_{w, b} \max_{\alpha \geq 0} L(w, b, \alpha) \quad (\text{primal form of SVM})$$

- if $1 - y_i (w^T x_i + b) > 0$, then $\alpha \rightarrow \infty$ and $L \rightarrow \infty$
 - if p^* exists, $1 - y_i (w^T x_i + b) \leq 0$
- if $1 - y_i (w^T x_i + b) < 0$, then $\alpha = 0$
- if $1 - y_i (w^T x_i + b) = 0$, then $\alpha_i (1 - y_i (w^T x_i + b)) = 0$
- this is part of **K.K.T conditions!**

- Dual form of SVM

$$d^* = \max_{\alpha \geq 0} \min_{w, b} L(w, b, \alpha)$$

Weak Duality

- $d^* \leq p^*$

$$\begin{aligned} d^* &= \max_{\alpha \geq 0} \min_{w, b} L(w, b, \alpha) \leq \max_{\alpha \geq 0} L(w, b, \alpha) \quad \forall w, b \\ d^* &\leq \min_{w, b} \max_{\alpha \geq 0} L(w, b, \alpha) = p^* \end{aligned}$$

- $p^* - d^*$ is called **duality gap**

Strong Duality

- $d^* = p^*$
- without duality gap!
- Slater's Condition
 - when objective is **convex** and constraints are **linear**, then $d^* = p^*$
 - our function just fits these conditions!
- So now we can solve the dual problem to solve the primal problem
 - if we have the solution of primal problem w_p, b_p and the solution of dual problem α_d

$$\begin{aligned} p^* &= \min_{w, b} \max_{\alpha \geq 0} L(w, b, \alpha) = \max_{\alpha \geq 0} L(w_p, b_p, \alpha) \\ &\geq L(w_p, b_p, \alpha_d) \geq \min_{w, b} L(w, b, \alpha_d) = \max_{\alpha \geq 0} \min_{w, b} L(w, b, \alpha) = d^* \end{aligned}$$

- 根据Slator's Condition，这里我们就可以直接取等了

Dual problem

- 我们首先来看函数的内层 $\min_{w, b} L(w, b, \alpha)$
 - 由于 L 是凸函数，所以我们可以通过找导函数的零点来求最小值

$$\begin{aligned} \frac{\partial L}{\partial w} &= w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \\ \frac{\partial L}{\partial b} &= - \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

- 所以我们有：

$$\begin{aligned} w &= \sum_{i=1}^n \alpha_i y_i x_i = 0 \\ \sum_{i=1}^n \alpha_i y_i &= 0 \end{aligned}$$

- 可以发现这其实也是K.K.T conditions中的 $\nabla_{w, b} L(w, b, \alpha) = 0$

- Substitute w into L :

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i x_i \right)^T \left(\sum_{j=1}^n \alpha_j y_j x_j \right) + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i y_i x_i^T \left(\sum_{j=1}^n \alpha_j y_j x_j \right) \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned}$$

- now we need to solve the **dual problem of SVM**:

$$\begin{aligned} \max \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \alpha_i \geq 0 \quad \forall i, \quad \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

- **SMO**(Sequential Minimal Optimization)

- every time select α_i, α_j , fix remaining $n - 2$ variables
- $\alpha_i y_i + \alpha_j y_j = -\sum_{k \neq i, j} \alpha_k y_k$ represent α_j with α_i and solve problem w.r.t α_i
- iterate until convergence
- When we have solution α^* for d^* , how to get w^*, b^* :

$$w^* = \sum \alpha_i y_i x_i$$

$$b^* = \frac{1}{y_i} - w^{*T} x_i$$

(x_i, y_i) is a support vector

- How to find a support vector:
 - recall: $\begin{cases} 1 - y_i(w^T x_i + b) = 0 & \Rightarrow \alpha_i > 0 \quad \text{active} \\ 1 - y_i(w^T x_i + b) > 0 & \Rightarrow \alpha_i = 0 \quad \text{inactive} \end{cases}$
 - then: $\begin{cases} \text{if } \alpha_i^* = 0 & \Rightarrow (x_i, y_i) \text{ is not a support vector} \\ \text{if } \alpha_i^* > 0 & \Rightarrow (x_i, y_i) \text{ is a support vector} \end{cases}$
 - we can save **only** the α_i^* of support vectors to save the model
 - α 相当于权重，只有被认为是支持向量的权重才会大于0，其他都置0，表示其他的向量对于超平面完全是没用的，可以不用考虑
- when we have a new test point x :

$$f(x) = \sum_{i \in \text{support vectors}} \alpha_i^* y_i x^T x_i + b^*$$

- 对于一个新来的点，我们只要让其和支持向量做内积，看 $f(x)$ 大于0还是小于0即可将其分类

Kernel Trick

- Define similarity between 2 points X, Z
 - **Linear Kernel** $K(X, Z) = X^T Z$
 - **Polynomial Kernel** $K(X, Z) = (X^T Z + 1)^P$
 - **Gaussian / RBF(Radial Basis Function) Kernel** $K(X, Z) = \exp(-\frac{\|X-Z\|^2}{2\sigma^2})$
- e.g. $\Phi: \mathbb{R}^d \rightarrow \mathbb{R}^d$
 - $\Phi(X)$ maps X to higher dimension
 - $X = (X_1, X_2)^T \in \mathbb{R}^2 \rightarrow \Phi(X) = (1, X_1, X_2, X_1^2, X_2^2, X_1 X_2) \in \mathbb{R}^6$
- 很多时候，当数据线性不可分时，我们可以通过给数据升维，使得数据变得线性可分
- **Kernel Trick**:
 - a way to get $\Phi(X)^T \Phi(Z)$ directly in original space, without explicitly mapping data to high-dimension first
 - 我们虽然希望可以用高维数据进行计算，但实际上我们并不需要直接去求他们的内积，而是直接使用上述的核函数即可，因为这些核函数就可以写成一些高维数据的内积
 - **这样我们将上述dual problem中的 $x_i^T x_j$ 部分替换为 $K(x_i, x_j)$ 即可**
- e.g. $X = (X_1, X_2), Z = (Z_1, Z_2)$

$$(X^T Z + 1)^2 = (X_1 X_2 + Z_1 Z_2 + 1)^2 = (1, \sqrt{2}X_1, \sqrt{2}X_2, X_1^2, X_2^2, \sqrt{2}X_1 X_2)^T (1, \sqrt{2}X_1, \sqrt{2}X_2, X_1^2, X_2^2, \sqrt{2}X_1 X_2)$$
- Gaussian Kernel

$$K(X, Z) = \exp(-\frac{\|X\|^2}{2\sigma^2}) \exp(-\frac{\|Z\|^2}{2\sigma^2}) \exp(-\frac{X^T Z}{\sigma^2})$$

and we know that:

$$\exp(-\frac{X^T Z}{\sigma^2}) = 1 + \frac{X^T Z}{\sigma^2} + \frac{1}{2!} (\frac{X^T Z}{\sigma^2})^2 + \frac{1}{3!} (\frac{X^T Z}{\sigma^2})^3 + \dots$$

$$= \sum_{p=0}^{\infty} \frac{1}{p!} (\frac{X^T Z}{\sigma^2})^p$$

- union of polynomial kernels from $p = 0$ to ∞
- implicitly map data to infinite-dimension space and do inner product
- 这里的 σ 可以自行控制，当 σ 大时，泰勒多项式会很快趋近于0，即此时映射成的向量维数不会很高；而当 σ 小时，则能够泰勒展开较多的项，此时就会映射成一个较高维的向量。可以说， σ 控制了映射维数的大小
- $\sigma \rightarrow 0$ 时， $K(X, Z) \rightarrow 0$ ，此时类似 $1 - NN$ ，只有与 X 最近的点才会起作用
- d^* 与 p^* :
 - p^* 计算 w, b 共 $d + 1$ 维，其对偶形式 d^* 则计算 α 共 n 维，一般来说 n 是很大的，但我们依然选择去用 d^* 计算，主要是考虑到了高斯核函数，根据上述的泰勒展开，我们可以发现理论上这个核函数可以把原始向量映射到一个**无穷维的空间**中，那么显然此时 w, b 也可以无限大，再用 p^* 来计算就不那么划算了

Soft-margin SVM

- Slack variables to handle outliers
 - previously, we require $y_i(w^T x_i + b) \geq 1$. Now, we introduce slack variables $\xi_i \geq 0$ and require $y_i(w^T x_i + b) \geq 1 - \xi_i$
 - must restrict ξ_i to be small
 - ξ_i 表示样本最多可以与 $w^T x_i + b = 0$ 偏离的程度

Unconstrained Form

- Soft-margin SVM

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

化简后可得 $\min_{w, b} \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \max\{0, 1 - y_i(w^T x_i + b)\}$

- becomes an **unconstrained** optimization
- $\max\{0, 1 - y_i(w^T x_i + b)\}$ is called **Hinge Loss**
- $c \rightarrow \infty \Rightarrow \xi_i = 0 \Rightarrow$ hard margin
- 要特别注意的是，任何一个在**软间隔内部**的点也是支持向量，也就是说，软间隔是由那些分错的点+分对但在间隔内部+在间隔上的点组成的

Dual Form

- Dual Form of soft-margin SVM

$$\begin{aligned} \max_{\alpha, \beta \geq 0} \min_{w, b, \xi} \quad & L(w, b, \xi, \alpha, \beta) \\ = \quad & \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - \xi_i - y_i(w^T x_i + b)) - \sum_{i=1}^n \beta_i \xi_i \end{aligned}$$

- Partial derivatives

$$\begin{aligned} \frac{\partial L}{\partial w} &= w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \\ \frac{\partial L}{\partial b} &= - \sum_{i=1}^n \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} &= c - \alpha_i - \beta_i = 0 \end{aligned}$$

整理可得

$$\begin{aligned} \max \quad & \frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i x_i \right)^T \left(\sum_{j=1}^n \alpha_j y_j x_j \right) + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i y_i x_i^T \left(\sum_{j=1}^n \alpha_j y_j x_j \right) \\ = \max \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq c, \quad \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

- $c \rightarrow \infty \Rightarrow$ hard margin
- c is small \Rightarrow no individual point dominates the prediction(hyperplane) and is robust to outlier
- c controls the **strength of regularization**