# Support Vector Machines 1

# Review: The Kernel Trick

- A function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is an inner product kernel if there exists a function $\Phi(\boldsymbol{x})$ mapping to an inner product space such that

$$k(\boldsymbol{u}, \boldsymbol{v}) = \langle \Phi(\boldsymbol{u}), \Phi(\boldsymbol{v}) \rangle \qquad \forall \boldsymbol{u}, \boldsymbol{v}.$$

- A machine learning algorithm is said to be *kernelizable* if it is possible to formulate the algorithm such that all training instances $\boldsymbol{x}_i$ and any test instance $\boldsymbol{x}$ occur exclusively in inner products of the form $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$, $\langle \boldsymbol{x}_i, \boldsymbol{x} \rangle$ or $\langle \boldsymbol{x}, \boldsymbol{x} \rangle$.
- Suppose $\boldsymbol{\Phi}$ is a feature map associated to an inner product kernel $k$.
- If we apply a kernelizable algorithm to the training data

$$(\Phi(x_1), y_1), \ldots, (\Phi(x_n), y_n)$$

  then we can formulate the algorithm such that transformed feature vectors only appear via inner products $\langle \boldsymbol{\Phi}(\boldsymbol{x}), \boldsymbol{\Phi}(\boldsymbol{x}') \rangle$ with other transformed feature vectors.
- Can implement by evaluating $k(x, x')$

  which eliminates the need to ever compute $\boldsymbol{\Phi}(\boldsymbol{x})$ explicitly.

# OSM Hyperplane Classifier

- Our goal today is to kernelize the OSM hyperplane (equivalently, regularized ERM with the hinge loss). This is the linear classifier

$$\boldsymbol{x} \mapsto \operatorname{sign}\{(\boldsymbol{w}^*)^T \boldsymbol{x} + b^*\}$$

where $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ is a solution of

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \forall i \in \{1, \ldots, n\} \\ & \xi_i \geq 0 \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{OSM}$$

- It's not obvious how to kernelize this method.
- To kernelize it, we will apply the theory of constrained optimization.

# Constrained Optimization

- A *constrained optimization problem* has the form

$$\min_{u \in \mathbb{R}^d} \quad f(u)$$

$$\text{s.t.} \quad g_i(u) \le 0 \quad \forall i \in \{1, \ldots, r\}$$

$$h_j(u) = 0 \quad \forall j \in \{1, \ldots, s\}$$

- If $\boldsymbol{u}$ satisfies all of the constraints, it is said to be *feasible*.

$$\text{feasible set} := \{u : g_i(u) \le 0 \ \forall i,\ h_j(u) = 0 \ \forall j\}$$

$$f, g_1, \ldots, g_r, h_1, \ldots, h_s : \mathbb{R}^d \longrightarrow \mathbb{R}$$

# Lagrangian

- The *Lagrangian* is

$$L(u,\lambda,\nu) = f(u) + \sum_{i=1}^{r} \lambda_i g_i(u) + \sum_{j=1}^{s} \nu_j h_j(u)$$

- $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_r]^T$ and $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_s]^T$ are called

Lagrange multipliers, dual variables

# Dual Function

- The *Lagrangian dual function* is

$$L_D(\lambda, \nu) = \min_{x \in \mathbb{R}^d} L(x, \lambda, \nu)$$

- $L_D$ is always concave
- The *dual optimization problem* is

$$\max_{\lambda, \nu : \lambda_i \geq 0} L_D(\lambda, \nu) = \max_{\lambda, \nu : \lambda_i \geq 0} \left[ \min_{x \in \mathbb{R}^d} L(x, \lambda, \nu) \right]$$

- The original constrained optimization problem is sometimes called the primal (optimization) problem

# Rewriting the Primal

- Recall the Lagrangian:

$$L(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\boldsymbol{u}) + \sum_{i=1}^{r} \lambda_i g_i(\boldsymbol{u}) + \sum_{j=1}^{s} \nu_j h_j(\boldsymbol{u})$$

- Observe that the primal may be re-written

$$\min_{u} \underbrace{\left[ \max_{\nu, \lambda : \lambda_i \geq 0} L(u, \lambda, \nu) \right]}_{= \begin{cases} f(u) & u \text{ feasible} \\ \infty, & u \text{ not feasible} \end{cases}}$$

# Weak Duality

- Denote the optimal objective function values of the primal and dual

$$p^* = \min_{\boldsymbol{u}} \max_{\boldsymbol{\lambda},\boldsymbol{\nu}:\lambda_i \geq 0} L(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\nu})$$

$$d^* = \max_{\boldsymbol{\lambda},\boldsymbol{\nu}:\lambda_i \geq 0} \min_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \max_{\boldsymbol{\lambda},\boldsymbol{\nu}:\lambda_i \geq 0} L_D(\boldsymbol{\lambda}, \boldsymbol{\nu})$$

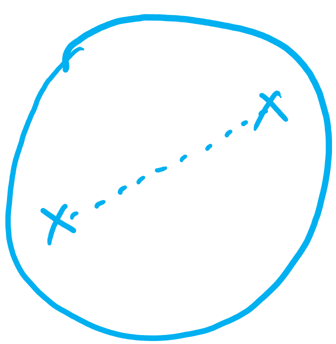
- *Weak duality* refers to the following fact, which always holds.
- **Theorem:** $d^* \leq p^*$

# Poll

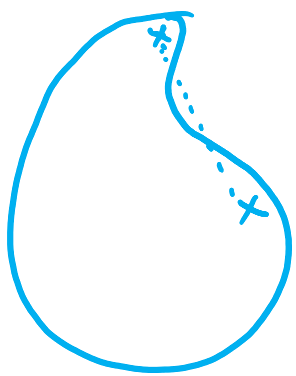
True or False: The constraint set (the set of all feasible points) is a *convex set* iff all of the functions $\cancel{f}$, $g_1, \ldots, g_r$, $h_1, \ldots, h_s$ are *convex functions*.

(A) True

(B) False

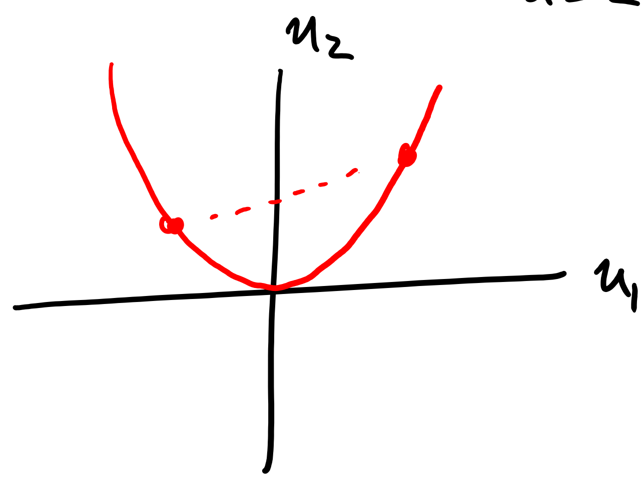


convex set



not a convex set

d=2




$$h_1(u) = -u_2 + u_1^2$$

# Strong Duality

- If $p^* = d^*$, we say *strong duality* holds.
- The original constrained optimization problem is said to be *convex* if $f$ and $g_1, \ldots, g_r$ are convex functions and $h_1, \ldots, h_s$ are affine.
- We state the following without proof.
- **Theorem:** If the original problem is convex and a constraint qualification holds, then $p^* = d^*$.
- Examples of constraint qualifications:
  - All $g_i$ are affine
  - (Strict feasibility) $\exists \boldsymbol{u}$ s.t. $h_j(\boldsymbol{u}) = 0\ \forall j$ and $\underline{g_i(\boldsymbol{u}) < 0}\ \forall i$

Affine: An affine function is a function of the form $u \longmapsto c^T u + d$

# Poll

- Recall the constrained optimization problem defining the optimal soft-margin hyperplane classifier:

$$u = \begin{bmatrix} w \\ b \\ \xi \end{bmatrix}$$

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i = f(u) \qquad \text{(OSM)}$$

$$\left.\begin{aligned} \text{s.t.} \quad & y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned}\right\} g_i(u),\ i \in \{1, \ldots, 2n\}$$

- True or false: Strong duality holds for problem OSM.

  (A) True

  (B) False

# Big Picture

- For unconstrained optimization problems with differentiable objective function, we saw that
  - $\nabla f(\boldsymbol{u}) = \mathbf{0}$ is *necessary* for $\boldsymbol{u}$ to be a global minimizer
  - If $f$ is convex, then $\nabla f(\boldsymbol{u}) = \mathbf{0}$ is *sufficient* for $\boldsymbol{u}$ to be a global minimizer
- For constrained optimization problems with differentiable objective and constraints, a similar result holds where $\nabla f(\boldsymbol{u}) = \mathbf{0}$ is replaced by the

  Karush - Kuhn - Tucker (KKT) conditions

- We can use these conditions to solve and understand constrained optimization problems.

# KKT Conditions: Necessity

- From now on assume $f$, $g_i$ and $h_j$ are all differentiable.
- **Theorem:** If $p^* = d^*$, $\boldsymbol{u}^*$ is primal optimal, and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is dual optimal, then the KKT conditions hold:

zero vector

1) $\nabla_u f(u^*) + \sum_{i=1}^{r} \lambda_i^* \nabla_u g_i(u^*) + \sum_{j=1}^{s} \nu_i^* \nabla_u h(u^*) = 0$

2) $g_i(u^*) \leq 0 \quad \forall i$

3) $h_j(u^*) = 0 \quad \forall j$

4) $\lambda_i^* \geq 0 \quad \forall i$

5) $\lambda_i^* g_i(u^*) = 0 \quad \forall i$ (complementary slackness)

# KKT Conditions: Sufficiency

- **Theorem:** If the original problem is convex and $\tilde{\boldsymbol{u}}$, $\tilde{\boldsymbol{\lambda}}$, $\tilde{\boldsymbol{\nu}}$ satisfy the KKT conditions

  1. $\nabla_{\boldsymbol{u}} f(\tilde{\boldsymbol{u}}) + \sum_{i=1}^{r} \tilde{\lambda}_i \nabla_{\boldsymbol{u}} g_i(\tilde{\boldsymbol{u}}) + \sum_{j=1}^{s} \tilde{\nu}_j \nabla_{\boldsymbol{u}} h_j(\tilde{\boldsymbol{u}}) = \mathbf{0}$
  2. $g_i(\tilde{\boldsymbol{u}}) \leq 0 \ \forall i$
  3. $h_j(\tilde{\boldsymbol{u}}) = 0 \ \forall j$
  4. $\tilde{\lambda}_i \geq 0 \ \forall i$
  5. $\tilde{\lambda}_i g_i(\tilde{\boldsymbol{u}}) = 0 \ \forall i$

  then $\tilde{\boldsymbol{u}}$ is primal optimal, $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ is dual optimal, and strong duality holds.

# How Is This Useful?

- Sometimes it is easier to solve the primal (analytically or computationally) by first solving the dual, and then using the KKT conditions to relate the primal solution to the dual solution.

- For example: suppose strong duality holds. If $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is dual optimal, then (by the KKT necessity theorem) any primal optimal point $\boldsymbol{u}^*$ is a solution of

$$\nabla_u f(u^*) + \sum_{i=1}^{r} \lambda_i^* \nabla_u g_i(u^*) + \sum_{j=1}^{s} \nu_j^* \nabla_u h_j(u^*) = 0$$

- We can use this to find a $\boldsymbol{u}^*$. In other words, we can recover a primal solution from a dual solution.

# OSM Hyperplane

- Let's apply this theory to the OSM hyperplane classifier.

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \tag{OSM}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \quad \forall i \in \{1,\ldots,n\} \qquad (\alpha_i)$$

$$\xi_i \geq 0 \quad \forall i \in \{1,\ldots,n\} \qquad (\beta_i)$$

$$u = \begin{bmatrix} w \\ b \\ \xi \end{bmatrix} \in \mathbb{R}^{d+1+n}, \quad \xi = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix}, \quad \lambda = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

- First observation: This is a convex optimization problem, and the constraint functions are all affine, hence strong duality holds.

# Lagrangian

The Lagrangian is

$$
\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \overbrace{\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i}^{f(x)} \\
&\quad - \sum_{i=1}^{n} \alpha_i \overbrace{(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i)}^{-g_i(x),\ i=1,\dots,n} - \sum_{i=1}^{n} \underbrace{\beta_i \xi_i}_{-g_i(x),\ i=n+1,\dots,2n} \\
&= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + \sum_{i=1}^{n} \alpha_i \\
&\quad - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \xi_i \left(\frac{C}{n} - \alpha_i - \beta_i\right).
\end{aligned}
$$

# Dual Function

- The dual function is

$$L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}).$$

- The optimization problem defining the dual function is an unconstrained minimization with a convex, differentiable objective function.

- Therefore, for fixed $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, we know that $\boldsymbol{w}, b$ and $\boldsymbol{\xi}$ achieve the minimum iff

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum \alpha_i y_i \boldsymbol{x}_i = \mathbf{0}$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = \frac{C}{n} - \alpha_i - \beta_i = 0 \quad \forall i.$$

# Dual Function

- Plugging in these formulas the Lagrangian simplifies to

$$
\begin{aligned}
L_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2}\Big\| \sum_i \alpha_i y_i \boldsymbol{x}_i \Big\|^2 - \sum_i \alpha_i y_i \Big\langle \sum_j \alpha_j y_j \boldsymbol{x}_j, \boldsymbol{x}_i \Big\rangle + \sum_i \alpha_i \\
&= \frac{1}{2}\Big\langle \sum_i \alpha_i y_i \boldsymbol{x}_i, \sum_j \alpha_j y_j \boldsymbol{x}_j \Big\rangle - \Big\langle \sum_i \alpha_i y_i \boldsymbol{x}_i, \sum_j \alpha_j y_j \boldsymbol{x}_j \Big\rangle + \sum_i \alpha_i \\
&= -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_i \alpha_i .
\end{aligned}
$$

# Dual Optimization Problem

- Therefore, the dual optimization problem

$$\max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta} \geq \mathbf{0}} L_D(\boldsymbol{\alpha}, \boldsymbol{\beta})$$

may be written

$$\begin{aligned}
\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \quad & -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^{n} \alpha_i \\
\text{s.t} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \left.\begin{aligned} & \alpha_i + \beta_i = \frac{C}{n} \quad \forall i \\ & \alpha_i \geqslant 0, \beta_i \geqslant 0 \quad \forall i. \end{aligned}\right] \iff 0 \leq \alpha_i \leq \frac{C}{n} \ \forall i
\end{aligned}$$

# Dual Optimization Problem

- In summary, the dual optimization problem can be written

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^{n} \alpha_i \\
\text{s.t} \quad & \sum_i \alpha_i y_i = 0 \\
& 0 \le \alpha_i \le \frac{C}{n}, \quad \forall i = 1, \ldots, n
\end{aligned}
$$

- This is another quadratic program.
- What do you notice? dot products

# Solving Primal From Dual

- The problem is convex with ~~linear~~ affine constraints so strong duality holds.
- Let $\boldsymbol{\alpha}^*$ denote a dual solution, obtained by solving the dual QP.
- Let $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ denote a primal solution; we don't know it yet, just suppose it exists
- By the KKT necessity theorem, $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*)$ satisfy the KKT conditions.
- We may use these to compute $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ from $\boldsymbol{\alpha}^*$.
- From the first KKT condition

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

- See lecture notes for how to recover $b^*$.

# Final Classifier

$$h(x) = \text{sign}\left\{ (w^*)^T x + b^* \right\}$$

$$= \text{sign}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i \langle x_i, x \rangle + b^* \right\}$$

# Summary: Dual Formulation

- The OSM hyperplane is kernelizable: The dual is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^{n} \alpha_i \\ \text{s.t} \quad & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq \frac{C}{n}, \quad \forall i = 1, \ldots, n \end{aligned}$$

- The classifier is expressed

$$f(\boldsymbol{x}) = \operatorname{sign}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b^* \right\}$$

- Taking any $j$ with $0 < \alpha_j^* < \frac{C}{n}$, the offset is given by

$$b^* = y_j - \sum_{i=1}^{n} \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$$

# The Support Vector Machine

- Let $k$ be an inner product/SPD kernel
- The *support vector machine* is the classifier obtained by solving

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{n} \alpha_i \\ \text{s.t} \quad & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq \frac{C}{n}, \quad \forall i = 1, \ldots, n\end{aligned}$$

- The classifier is expressed

$$f(\boldsymbol{x}) = \operatorname{sign}\left\{\sum_{i=1}^{n} \alpha_i^* y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right\}$$

where $b^* = y_j - \sum_{i=1}^{n} \alpha_i^* y_i k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for any $j$ with $0 < \alpha_j^* < \frac{C}{n}$.

# Support Vectors

- From complementary slackness,

- If $\boldsymbol{x}_i$ satisfies

  we call $\boldsymbol{x}_i$ a *support vector*.

- Therefore, if $\boldsymbol{x}_i$ is *not* a support vector, then

- Conclusion: $\boldsymbol{w}^*$ depends only on the SVs: