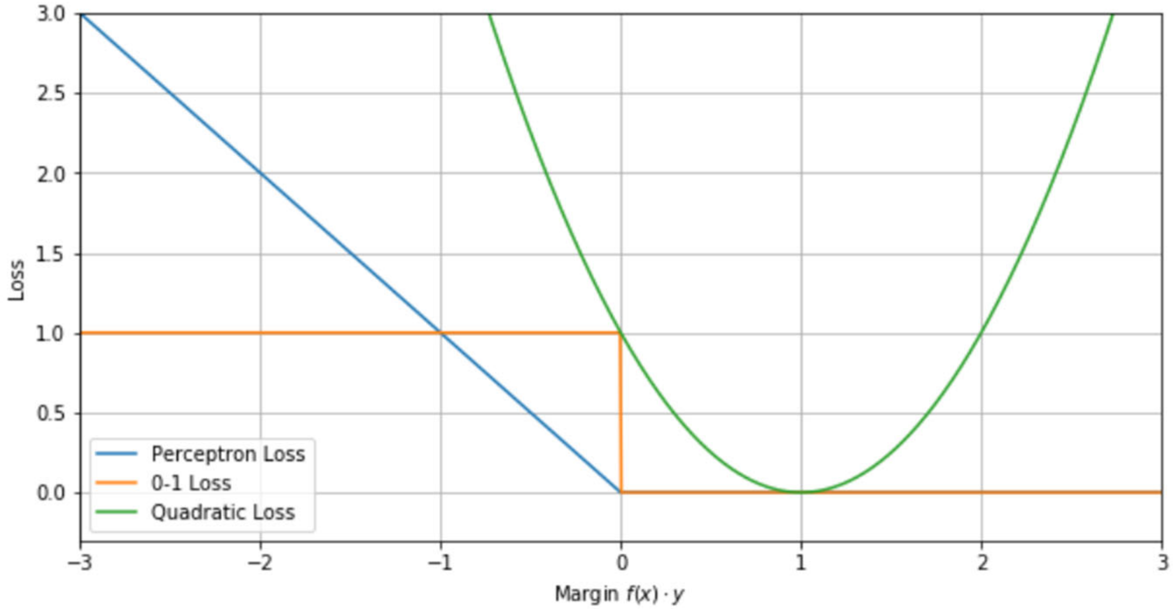
Questions

ADALINE Algorithm

- Questions on Piazza?
  - Please provide your feedback
- Question for You!

Could square Loss be used for Classification?

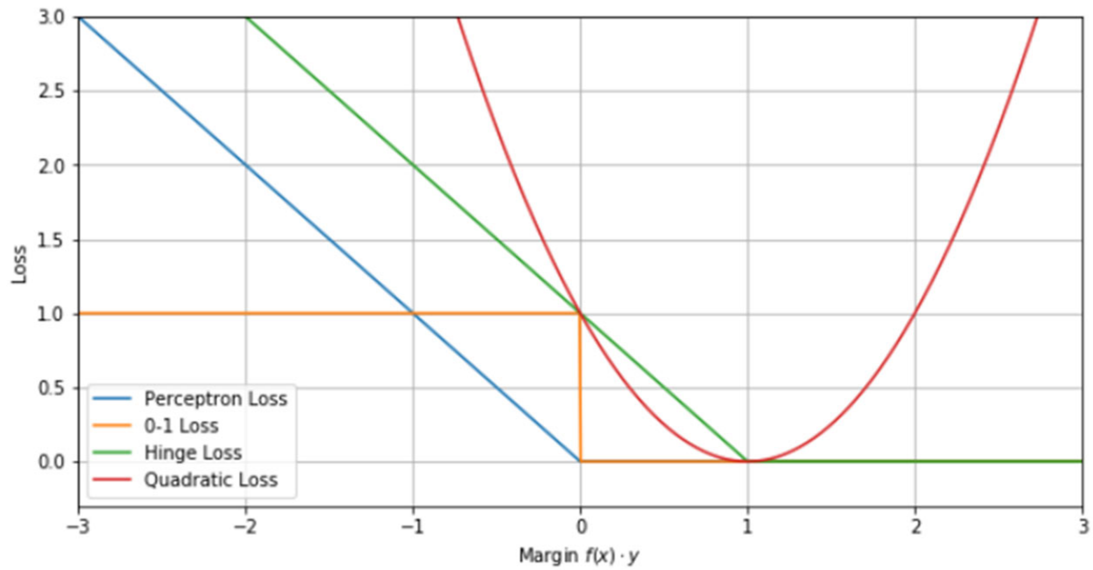


Questions

ADALINE Algorithm

- Questions on Piazza?
  - Please provide your feedback
- Question for You!

Could square Loss be used for Classification?



Center for Data Science
NYU

DS-GA 1003
Machine Learning

Week 5: Lecture 5

Support Vector Machines – Margin Based Classifiers

NYU Center for Data Science

DS-GA 1003
Machine Learning

How should we incorporate regularization into perceptron?

Week 5: Lecture 5

Support Vector Machines – Margin Based Classifiers

Adapted from Rosenberg, Miolane, Sontag, Rudin

Announcements

- Please check Week 5 agenda on NYU Classes
  - Homework 3
  - Midterm
  - Recordings
- Remember to post to Piazza

Check Calendar linked to NYU Classes for important dates

Review

- General minimization problems with constraints take the form

$$\text{minimize } f(x)$$
$$\text{subject to } g(x) \le 0$$

where $x \in \mathbb{R}^n$

- Suppose that the minimizer x occurs at the boundary of the constraint set
  - Here g(x) = 0 is an active constraint

- If we can find a vector *u* such that

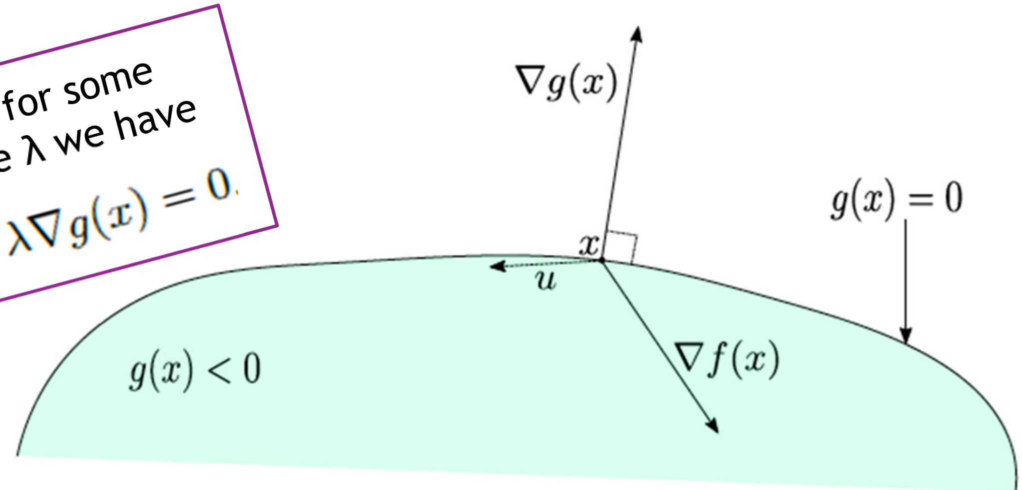
$$\langle u, \nabla g(x) \rangle < 0 \quad \text{and} \quad \langle u, \nabla f(x) \rangle < 0$$

then we can decrease the value of both *g* and *f* for some small number δ > 0

$$g(x + \delta u) \simeq g(x) + \delta \langle u, \nabla g(x) \rangle \le 0.$$
$$f(x + \delta u) \simeq f(x) + \delta \langle u, \nabla f(x) \rangle < f(x)$$

Therefore for some nonnegative λ we have

$$\nabla f(x) + \lambda \nabla g(x) = 0.$$



Review

- Suppose we want to solve

$$\text{minimize } x^2$$
$$\text{subject to } x \geq b$$

where $x \in \mathbb{R}$

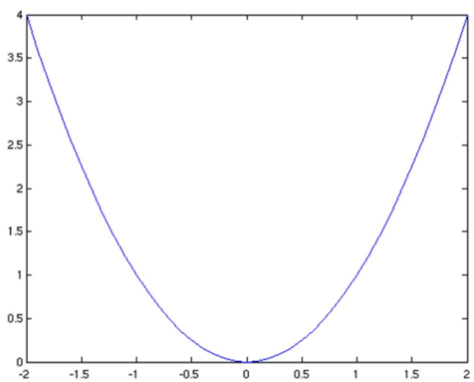
- We want to switch from constraint to penalization by studying the Lagrangian

$$L(x, \alpha) = x^2 - \alpha(x - b)$$

- We add a constraint on the additional variable to solve

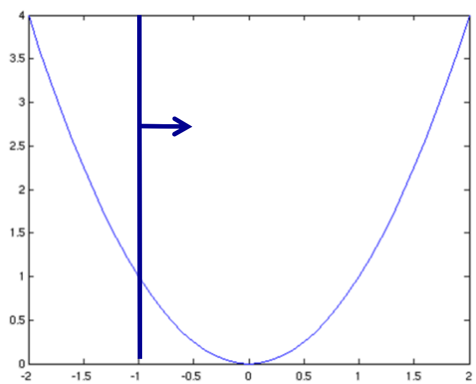
$$\min_x \max_\alpha \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

No Constraint



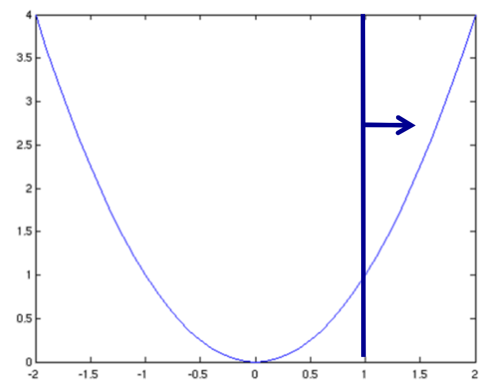
$x^*=0$

$x \geq -1$



$x^*=0$

$x \geq 1$



$x^*=1$

Review

- Suppose we want to solve

$$\text{minimize } x^2$$
$$\text{subject to } x \geq b$$

where $x \in \mathbb{R}$

- We want to switch from constraint to penalization by studying the Lagrangian

$$L(x, \alpha) = x^2 - \alpha(x - b)$$

- We add a constraint on the additional variable to solve

$$\min_x \max_\alpha \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

Having *min* outside forces *max* to give us the constraints

$x<b \rightarrow (x-b)<0 \rightarrow \max_\alpha -\alpha(x-b) = \infty$

$x=b \rightarrow \alpha$ can be anything

$x>b,\ \alpha \geq 0 \rightarrow (x-b)>0 \rightarrow \max_\alpha -\alpha(x-b) = 0,\ \alpha^*=0$

Review

- Suppose we want to solve

$$\text{minimize } x^2$$
$$\text{subject to } x \geq b$$

where $x \in \mathbb{R}$

- We want to switch from constraint to penalization by studying the Lagrangian

$$L(x, \alpha) = x^2 - \alpha(x - b)$$

- We add a constraint on the additional variable to solve

$$\min_x \max_\alpha \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

- Switching the order we can solve

$$\max_\alpha \min_x \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

x<b → (x-b)<0 → $\max_\alpha -\alpha(x-b) = \infty$

x=b → α can be anything

x>b, α≥0 → (x-b)>0 → $\max_\alpha -\alpha(x-b) = 0$, α*=0

Review

- Suppose we want to solve

$$\text{minimize } x^2$$
$$\text{subject to } x \geq b$$

where $x \in \mathbb{R}$

$$\boxed{\frac{\partial}{\partial x} L(x, \alpha) = 2x - \alpha \Rightarrow x = \frac{\alpha}{2}}$$

- We want to switch from constraint to penalization by studying the Lagrangian

$$L(x, \alpha) = x^2 - \alpha(x - b)$$

- We add a constraint on the additional variable to solve

$$\min_x \max_\alpha \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

- Switching the order we can solve

$$\max_\alpha \min_x \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

Review

- Suppose we want to solve

$$\text{minimize } x^2$$
$$\text{subject to } x \geq b$$

where $x \in \mathbb{R}$

$$\frac{\partial}{\partial x} L(x, \alpha) = 2x - \alpha \Rightarrow x = \frac{\alpha}{2}$$

$$\max_\alpha \min_x L(x, \alpha) = \max_\alpha b\alpha - \frac{\alpha^2}{4}$$

- We want to switch from constraint to penalization by studying the Lagrangian

$$L(x, \alpha) = x^2 - \alpha(x - b)$$

- We add a constraint on the additional variable to solve

$$\min_x \max_\alpha \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

- Switching the order we can solve

$$\max_\alpha \min_x \; L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

- We obtain the expected solution

$$x = \frac{2b}{2} = b$$

Agenda

- Convexity
  - Sets, Functions
- Duality
  - Min-Max Inequality
  - Complementary Slackness
- Support Vector Machines
  - Hard Margin, Soft Margin
- Understanding Support Vector Machines through Duality

References

- D. Rosenberg, Lecture Notes (link)
- Optional
  - D. Rosenberg, Lecture Notes (link)

Convexity

- Convex applies to both sets and functions
- Set C is convex if for any x_1 and x_2 in C we have

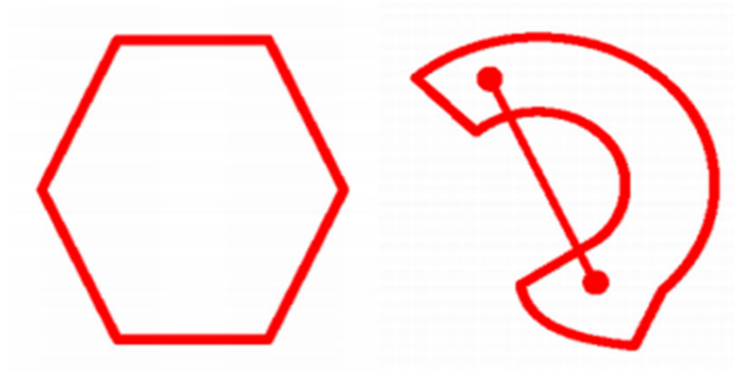
$$\theta x_1 + (1-\theta)x_2 \in C$$

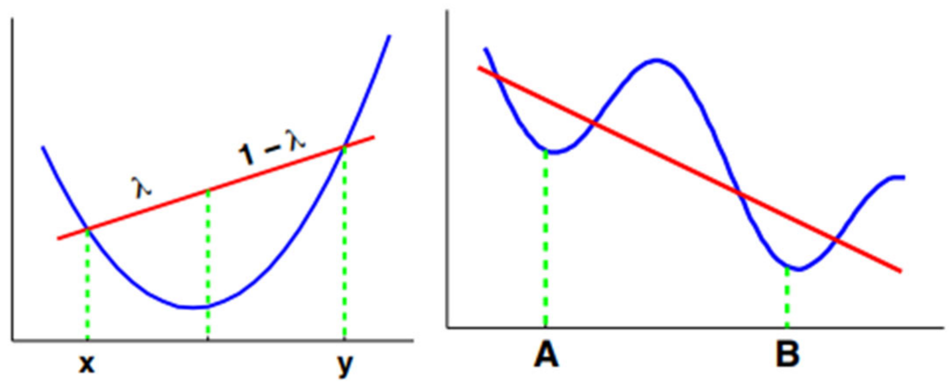
for any $0 \leqslant \theta \leqslant 1$

- Function $f : \mathbf{R}^n \to \mathbf{R}$

is convex if for any x,y and $0 \leqslant \theta \leqslant 1$ we have

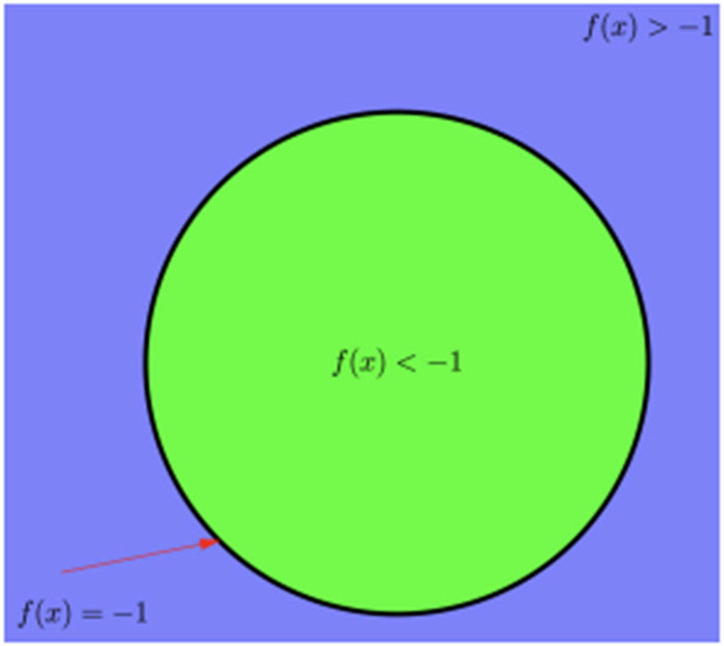
$$f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta)f(y).$$





Convexity

- A connection between convex sets and convex functions comes from looking at level sets
- Recall that a level set (aka contour line) for the value c is a points x such that f(x) = c.
- A sublevel set for value c is the set of points x such that f(x) ≤ c
- Sublevel sets of convex functions are convex.



Convexity

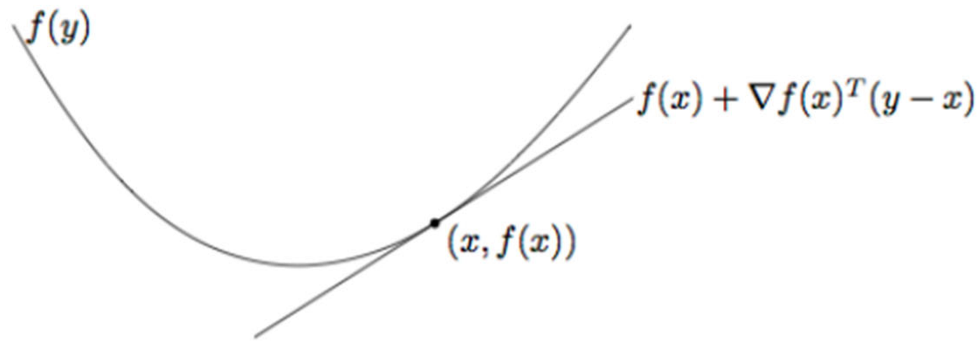
- Suppose $f : \mathbf{R}^d \to \mathbf{R}$ is differentiable. We can compute the gradient through each of the d partial derivatives. Can we predict f(y) from f(x) and $\nabla f(x)$?
- While convex functions are not linear functions, they behave like linear functions in a one-sided sense. The Taylor expansion near x is

$$f(y) \approx f(x) + \nabla f(x)^T (y - x)$$

- By convexity we have

$$f(y) \geqslant f(x) + \nabla f(x)^T (y - x)$$

- Therefore the linear approximation near x determine by the gradient is a global under-estimator of f



Convexity

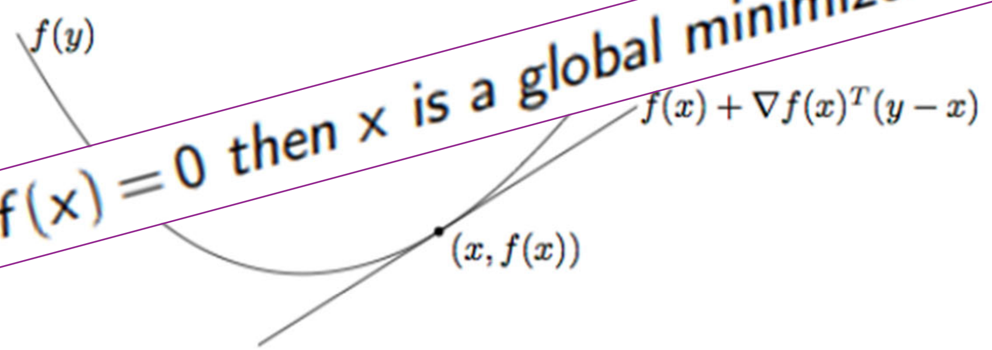
- Suppose $f:\mathbf{R}^d \to \mathbf{R}$ is differentiable. We can compute the gradient through each of the d partial derivatives. Can we predict f(y) from f(x) and $\nabla f(x)$?
- While convex functions are not linear functions, they behave like linear functions in a one-sided sense. The Taylor expansion near x is

$$f(y) \approx f(x) + \nabla f(x)^T (y-x)$$

- By convexity we have

$$f(y) \geqslant f(x) + \nabla f(x)^T (y-x)$$

- Therefore the linear approximation near x determine by the gradient is a global under-estimator of f



If $\nabla f(x) = 0$ then x is a global minimizer of f.

Convexity

- A function is strictly convex when the line segment connecting two points on the graph (aka secant line) lies strictly above the graph
- So when a function is convex, if we have a local minimum, then we know it's a global minimum
- Moreover with strict convexity the global minimum is unique

Examples of Convex Functions

$x \mapsto ax + b$ is both convex and concave on $\mathbf{R}$ for all $a, b \in \mathbf{R}$

$x \mapsto |x|^p$ for $p \geqslant 1$ is convex on $\mathbf{R}$

$x \mapsto e^{ax}$ is convex on $\mathbf{R}$ for all $a \in \mathbf{R}$

Every norm on $\mathbf{R}^n$ is convex (e.g. $\|x\|_1$ and $\|x\|_2$)

Max: $(x_1, \ldots, x_n) \mapsto \max\{x_1 \ldots, x_n\}$ is convex on $\mathbf{R}^n$

Optimization Problem

- We can study a more general minimization problem by incorporating equality constraints
- Note that equality constraints are not actually that different because

$$h(x) = 0$$

if and only if

$$h(x) \geqslant 0 \text{ AND } h(x) \leqslant 0$$

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots p, \end{array}$$

$$\boxed{\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & h(x) = 0 \end{array}}$$

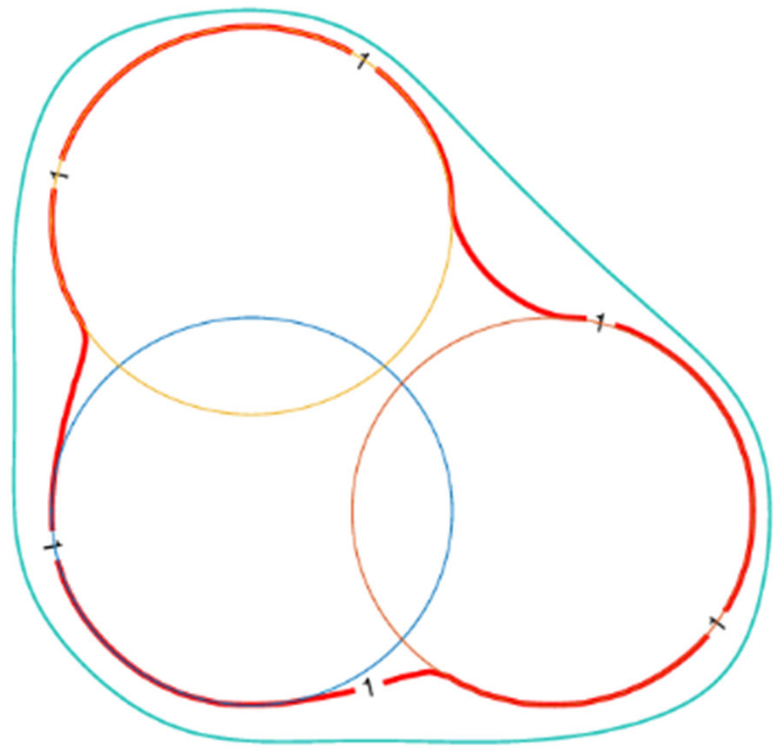
$$\boxed{\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & h(x) \leqslant 0 \\ & -h(x) \leqslant 0 \end{array}}$$

Optimization Problem

- Set of points satisfying the constraints is called the feasible set.
- Note that the intersection of convex sets is convex. So convex constraint functions give us convex feasible set.
- Optimal value p^* is

$$p^* = \inf\{f_0(x) \mid x \text{ satisfies all constraints}\}$$

- Optimal point x^* is a feasible point such that $f_0(x^*) = p^*$



Duality between Optimization Problems

- We can switch from the constraint form to the penalization form by forming the Lagrangian.

$$L(x,\lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x)$$

- Note that the additional variables are called dual variables (aka Lagrange multipliers)
- The primal form of the optimization problem is

$$p^* = \inf_x \sup_{\lambda \succeq 0} L(x,\lambda)$$

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \ldots, m \end{array}$$

$$\begin{aligned} \sup_{\lambda \succeq 0} L(x,\lambda) &= \sup_{\lambda \succeq 0} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right) \\ &= \begin{cases} f_0(x) & \text{when } f_i(x) \leqslant 0 \text{ all } i \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

Duality between Optimization Problems

- We can switch from the constraint form to the penalization form by forming the Lagrangian.

$$L(x,\lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x)$$

- Note that the additional variables are called dual variables (aka Lagrange multipliers)
- The primal form of the optimization problem is

$$p^* = \inf_x \sup_{\lambda \succeq 0} L(x,\lambda)$$

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \ldots, m \end{array}$$

$$\begin{aligned} \sup_{\lambda \succeq 0} L(x,\lambda) &= \sup_{\lambda \succeq 0} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right) \\ &= \begin{cases} f_0(x) & \text{when } f_i(x) \leqslant 0 \text{ all } i \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

Duality between Optimization Problems

- The dual form of the optimization problem is

$$d^* = \sup_{\lambda \succeq 0} \inf_x L(x, \lambda)$$

- Note that we may not have equality between the primal form and the dual form with conditions called constraint qualifications.
- However, we have weak duality

$$p^* \geqslant d^*$$

even for problems without convexity assumptions

$$\boxed{\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0, \quad i = 1, \dots, m \end{array}}$$

$$\begin{aligned} \sup_{\lambda \succeq 0} L(x, \lambda) &= \sup_{\lambda \succeq 0} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right) \\ &= \begin{cases} f_0(x) & \text{when } f_i(x) \leqslant 0 \text{ all } i \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

Duality between Optimization Problems

- The dual form of the optimization problem is

$$d^* = \sup_{\lambda \succeq 0} \inf_x L(x, \lambda)$$

- Note that we may not have equality between the primal form and the dual form with conditions called constraint qualifications.
- However, we have weak duality

$$p^* \geqslant d^*$$

even for problems without convexity assumptions

$$p^* = \inf_x \sup_{\lambda \succeq 0} \left[f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right]$$
$$\geqslant \sup_{\lambda \succeq 0, \nu} \inf_x \left[f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right] = d^*$$

- The equality between primal and dual problems is called strong duality. Otherwise we have a duality gap $p^* - d^*$

Duality between Optimization Problems

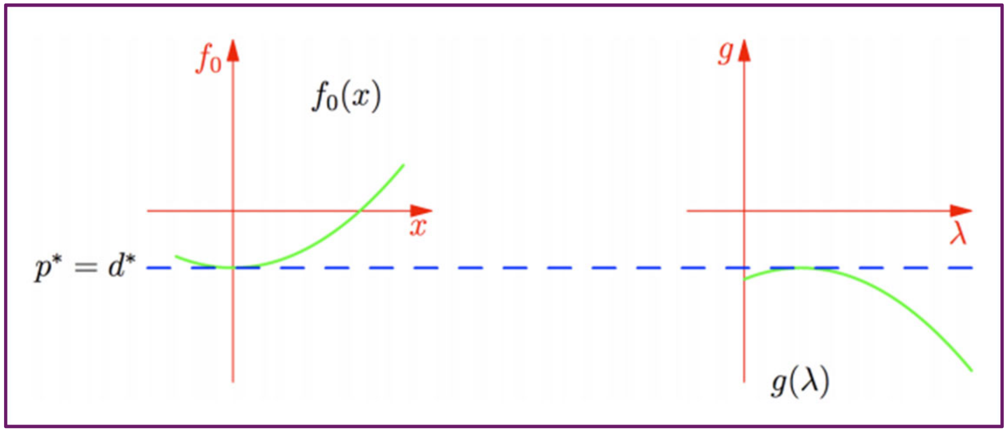
- The dual form of the optimization problem is

$$d^* = \sup_{\lambda \succeq 0} \inf_x L(x, \lambda)$$

- Note that we may not have equality between the primal form and the dual form with conditions called constraint qualifications.
- However, we have weak duality

$$p^* \geqslant d^*$$

even for problems without convexity assumptions



- The equality between primal and dual problems is called strong duality. Otherwise we have a duality gap p^* - d^*

Duality between Optimization Problems

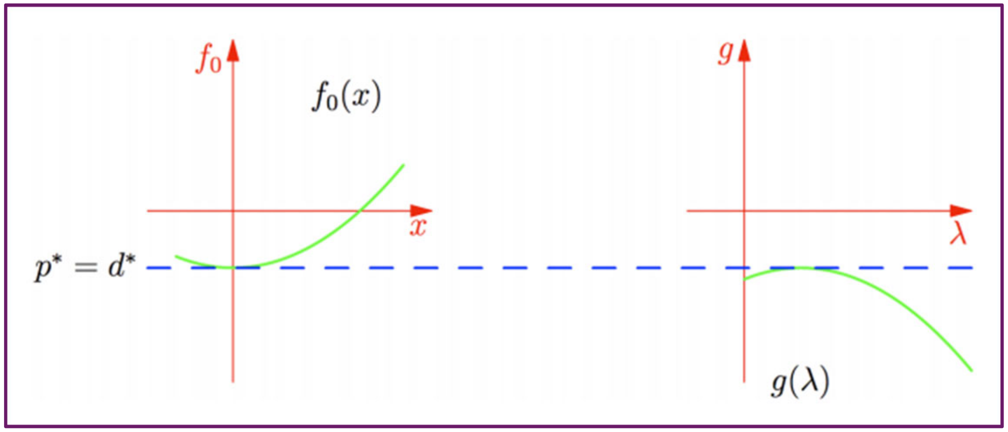
- The dual form of the optimization problem is

$$d^* = \sup_{\lambda \succeq 0} \inf_x L(x, \lambda)$$

- Note that we may not have equality between the primal form and the dual form with conditions called constraint qualifications.
- However, we have weak duality

$$p^* \geqslant d^*$$

even for problems without convexity assumptions



- The equality between primal and dual problems is called strong duality. Otherwise we have a duality gap $p^* - d^*$
- The dual function is always concave

$$g(\lambda) = \inf_x L(x, \lambda) = \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right)$$

Duality between Optimization Problems

- The dual form of the optimization problem is

$$d^* = \sup_{\lambda \succeq 0} \inf_x L(x, \lambda)$$

- Note that we may not have equality between the primal form and the dual form with conditions called constraint qualifications.
- However, we have weak duality

$$p^* \geqslant d^*$$

even for problems without convexity assumptions

- For convex optimization problems where
  - f_0 is a convex function
  - f_i is a convex function

we have Slater's sufficient conditions for strong duality

$$f_i(x) < 0 \text{ for } i = 1, \ldots, m$$

This must hold for some x.

- Note that for affine functions f_i the inequality doe not have to be strict.

$$f_i(x) \leqslant 0$$

Complementary Slackness

- Weak duality implies the dual form allows us to search for the best lower bound for the primal problem.
- We need to constrain the dual variables to be positive

$$\lambda \succeq 0$$

Otherwise the dual variables are not dual feasible.

- The dual form can be more simple or more informative yielding insights into dual problem

$$\boxed{\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda \succeq 0 \end{array}}$$

Complementary Slackness

- Weak duality implies the dual form allows us to search for the best lower bound for the primal problem.
- We need to constrain the dual variables to be positive

$$\lambda \succeq 0$$

Otherwise the dual variables are not dual feasible.

- The dual form can be more simple or more informative yielding insights into dual problem

$$\boxed{\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda \succeq 0 \end{array}}$$

- Assuming we have strong duality $p^* = d^*$, we can connect the primal variables and the dual variables

$$\begin{aligned} f_0(x^*) &= g(\lambda^*) = \inf_x L(x, \lambda^*) \\ &\leqslant L(x^*, \lambda^*) \\ &= f_0(x^*) + \sum_{i=1}^{m} \underbrace{\lambda_i^* f_i(x^*)}_{\leqslant 0} \\ &\leqslant f_0(x^*). \end{aligned}$$

Complementary Slackness

- Here x^* is the primal optimal and λ^* is the dual optimal
- Each term in the sum

$$\sum_{i=1} \lambda_i^* f_i(x^*)$$

must be 0 because of the primal and dual constraints dictating the sign of the terms

- Therefore

$$\boxed{\lambda_i^* f_i(x^*) = 0, \quad i = 1, \ldots, m}$$

- We call the relationship complementary slackness

- Note that $L(x^*, \lambda^*) = \inf_x L(x, \lambda^*)$ implies
$$\nabla_x L(x^*, \lambda^*) = 0$$

for $x \mapsto L(x, \lambda^*)$ differentiable.

- Assuming we have strong duality $p^* = d^*$, we can connect the primal variables and the dual variables

$$\begin{aligned} f_0(x^*) &= g(\lambda^*) = \inf_x L(x, \lambda^*) \\ &\leqslant L(x^*, \lambda^*) \\ &= f_0(x^*) + \sum_{i=1}^{m} \underbrace{\lambda_i^* f_i(x^*)}_{\leqslant 0} \\ &\leqslant f_0(x^*). \end{aligned}$$

Kuhn Tucker Conditions

- Assume we have a convex optimization problem where
  - f_0 is a convex function
  - f_i is a convex function
- For example suppose we have checked Slater's sufficient conditions for $p^* = d^*$

$$f_i(x) < 0 \text{ for } i = 1, \dots, m$$

- Assume that $x \mapsto L(x, \lambda^*)$ is differentiable
- We want to determine optimal primal variables and optimal dual variables by checking conditions

- First Order Condition

$$\nabla_x L(x^*, \lambda^*) = 0$$

- Dual Feasible

$$\lambda^* \succeq 0$$

- Primal Feasible

$$f_i(x^*) \leqslant 0$$

- Complementary Slackness

$$\boxed{\lambda_i^* f_i(x^*) = 0, \quad i = 1, \dots, m}$$

Exercise

- Suppose that graders and students compete to minimize or maximize points on assignments.
- Assume that each assignment has five questions.
- The graders decide to score only one question from each assignment. Graders will set the number of points lost to $+\infty$ for any omitted problem.
- The students must decide on allocation of time for each problem to avoid losing points.

Each row is a student strategy

$$A = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 8 & 8 & 1 & 8 & 8 \\ +\infty & +\infty & +\infty & 0 & +\infty \end{bmatrix}$$

Each column is a grader strategy

Exercise

- Assuming that
  - Student is forced to submit the homework without knowing the graders' choice of problem
  - Graders are allowed to decide which problem to grade after having seen the student's submission
- The number of points lost will be

$$p^* = \min_i \max_j a_{ij}$$

- Here students pick strategy first and graders pick strategy second.

$$A = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 8 & 8 & 1 & 8 & 8 \\ +\infty & +\infty & +\infty & 0 & +\infty \end{bmatrix}$$

Exercise

- Assuming that
  - Graders announce the problem in advance
  - Students get to decide on allocation of time for that problem
- The number of points lost will be

$$d^* = \max_j \min_i a_{ij}$$

- Here graders pick strategy first and students pick strategy second.

$$A = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 8 & 8 & 1 & 8 & 8 \\ +\infty & +\infty & +\infty & 0 & +\infty \end{bmatrix}$$

Exercise

- Show that for any matrix

$$A = (a_{ij}) \in \mathbb{R}^{m \times n}$$

we always have

$$\boxed{\max_j \min_i a_{ij} = d^* \leq p^* = \min_i \max_j a_{ij}}$$

- Check the inequality through showing

$$d^* = a_{i_d j_d} \leq a_{i_p j_d} \leq a_{i_p j_p} = p^*$$

Calculate p*

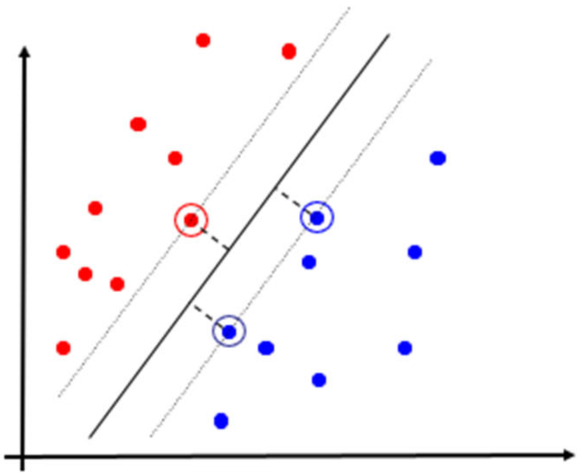
$$A = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 8 & 8 & 1 & 8 & 8 \\ +\infty & +\infty & +\infty & 0 & +\infty \end{bmatrix}$$

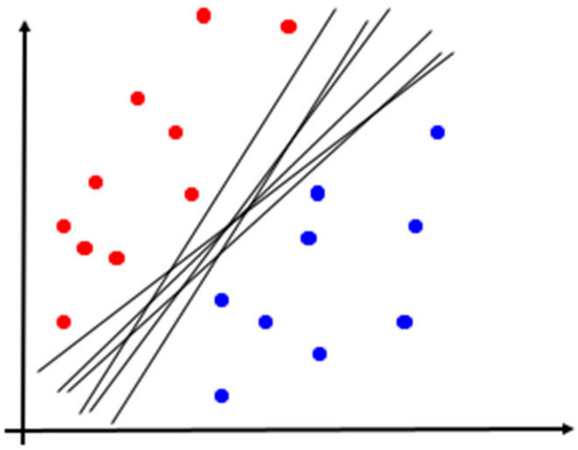
Calculate d*

Perceptron

- Remember some of the properties of perceptron algorithm
- Advantages
  - Error Bound
  - Online Algorithm
- Disadvantages
  - Many Decision Boundaries
  - Separable Data

Distance from the decision boundary should reflect confidence





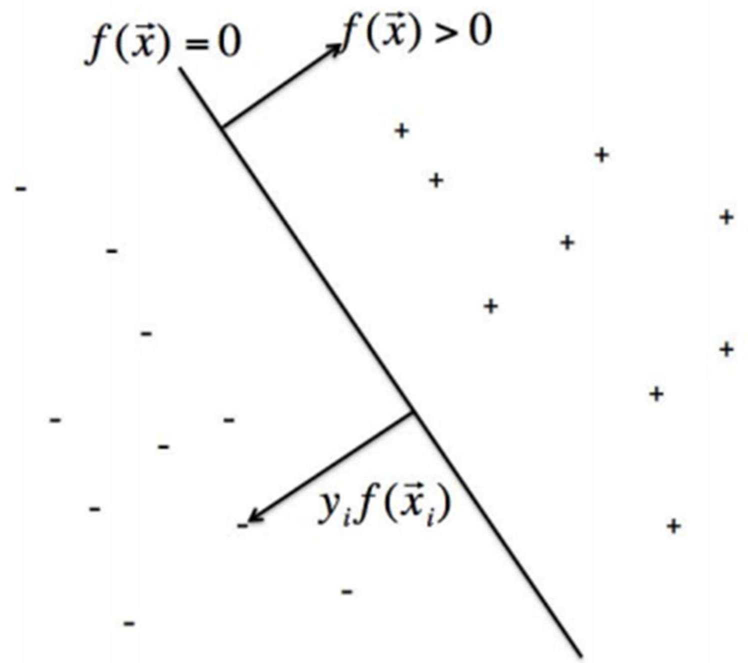
We want to make the distance from the decision boundary large

Margin

- Recall that margin intuitively means signed distance from decision boundary
- For hypothesis f, we could define the functional margin as

$$y\, f(x)$$

- The decision boundary was the level set of f for value 0.
- The decision boundary splits into two half-spaces depending on the sign of the margin
  - Positive
  - Negative



Margin

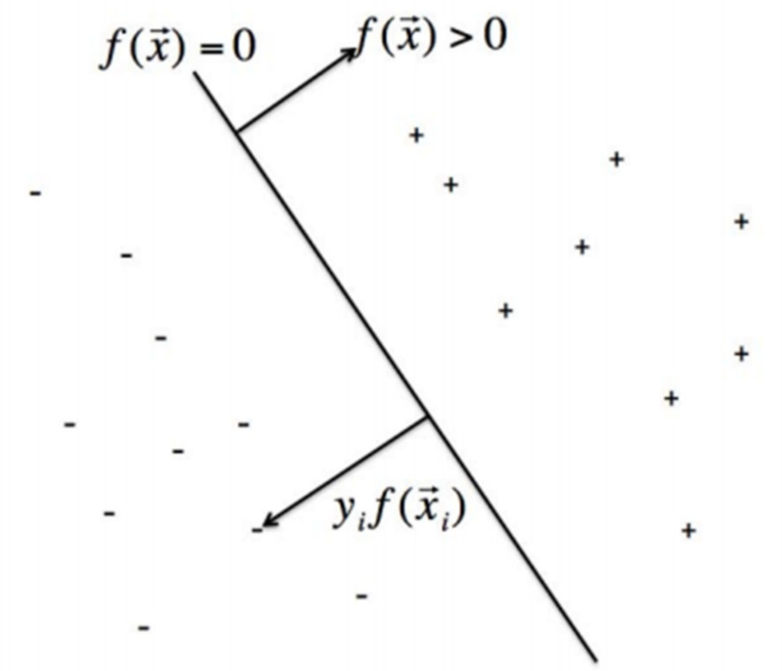
Focus on linear hypotheses

$$f(\mathbf{x}) = \sum_{j=1}^{n} w^{(j)} x^{(j)} + w_0$$

- Recall that margin intuitively means signed distance from decision boundary
- For hypothesis f, we could define the functional margin as

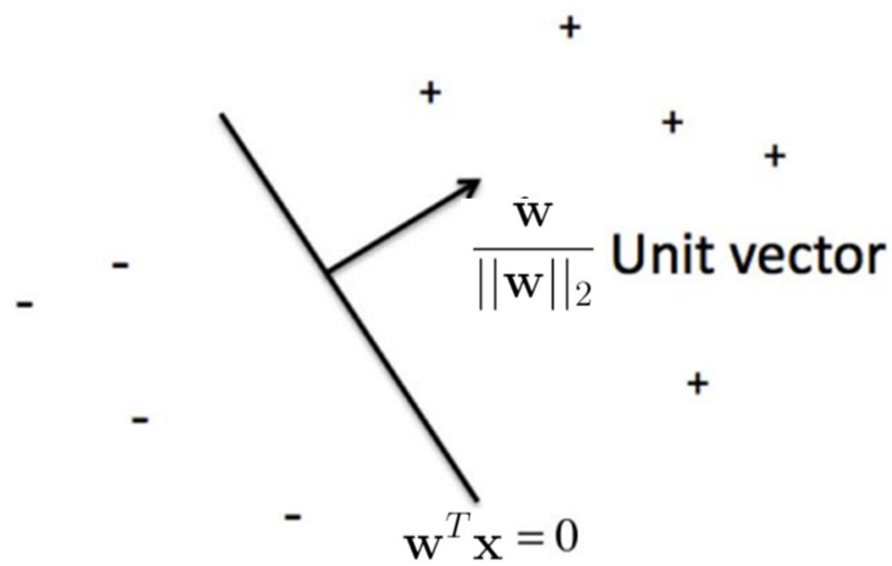
$$y\ f(x)$$

- The decision boundary was the level set of f for value 0.
- The decision boundary splits into two half-spaces depending on the sign of the margin
  - Positive
  - Negative



Support Vector Machine

- Note that scale the hypothesis f by a large number would increase margin for correct classifications
- We must recognize the scaling issue in maximizing the minimum distance of the training points from the decision boundary
- Remember that the weights determine a plane through the origin. The offset w_0 shifts the plane away from the origin



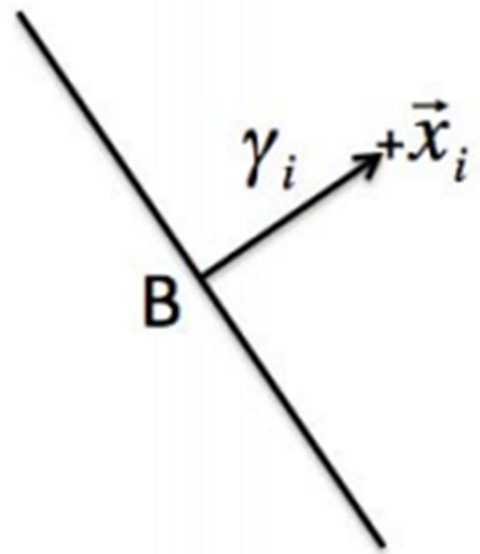
Support Vector Machine

- Recall projection of a point x_i onto the plane determine by w shifted by w_0
- Suppose the nearest point is B. Denote the signed distance by γ_i
- Note that B is

$$\mathbf{B} = \mathbf{x}_i - \gamma_i \frac{\mathbf{w}}{||\mathbf{w}||_2}$$

- Since B lies on the decision boundary, we have

$$w^T B + w_0 = 0$$



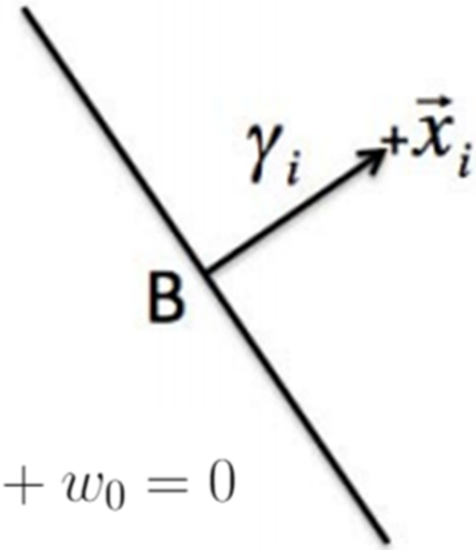
Support Vector Machine

- Recall projection of a point x_i onto the plane determine by w shifted by w_0
- Suppose the nearest point is B. Denote the signed distance by γ_i
- Note that B is

$$\mathbf{B} = \mathbf{x}_i - \gamma_i \frac{\mathbf{w}}{||\mathbf{w}||_2}$$

- Since B lies on the decision boundary, we have

$$w^T B + w_0 = 0$$



$$\mathrm{w}^T \left(\mathbf{x}_i - \gamma_i \frac{\mathbf{w}}{||\mathbf{w}||_2} \right) + w_0 = 0$$

$$\mathrm{w}^T \mathbf{x}_i - \gamma_i \frac{||\mathbf{w}||_2^2}{||\mathbf{w}||_2} + w_0 = 0$$

$$\gamma_i = \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

Support Vector Machine

- Therefore to choose the hypothesis so the training points are far away from the decision boundary, we need to study maximize the minimum geometric margin

$$\gamma_i = \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

$$\max_f \max_\gamma \gamma \text{ subject to } y_i f(x_i) \geq \gamma \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- Therefore to choose the hypothesis so the training points are far away from the decision boundary, we need to study maximize the minimum geometric margin

$$\gamma_i = \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

$$\max_f \max_\gamma \gamma \text{ subject to } y_i f(x_i) \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma, \mathbf{w}, w_0} \gamma \text{ subject to } y_i \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2} \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma, \mathbf{w}, w_0} \gamma \text{ subject to } y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) \geq \gamma ||\mathbf{w}||_2 \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- Therefore to choose the hypothesis so the training points are far away from the decision boundary, we need to study maximize the minimum geometric margin

$$\gamma_i = \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

- Note that we can scale w and w_0. By convention we set

$$||\mathbf{w}||_2 = \frac{1}{\gamma}$$

$$\max_f \max_\gamma \gamma \text{ subject to } y_i f(x_i) \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma, \mathbf{w}, w_0} \gamma \text{ subject to } y_i \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2} \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma, \mathbf{w}, w_0} \gamma \text{ subject to } y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) \geq \gamma ||\mathbf{w}||_2 \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- Therefore to choose the hypothesis so the training points are far away from the decision boundary, we need to study maximize the minimum geometric margin

$$\gamma_i = \frac{\mathbf{w}^T\mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

- Note that we can scale w and w_0. By convention we set

$$||\mathbf{w}||_2 = \frac{1}{\gamma}$$

$$\max_f \max_\gamma \gamma \text{ subject to } y_i f(x_i) \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma,\mathbf{w},w_0} \gamma \text{ subject to } y_i \frac{\mathbf{w}^T\mathbf{x}_i + w_0}{||\mathbf{w}||_2} \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma,\mathbf{w},w_0} \gamma \text{ subject to } y_i \left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq \gamma||\mathbf{w}||_2 \text{ for } i = 1, \ldots, m$$

$$\max_{\mathbf{w},w_0} \frac{1}{||\mathbf{w}||_2} \text{ subject to } y_i \left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq 1 \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- Therefore to choose the hypothesis so the training points are far away from the decision boundary, we need to study maximize the minimum geometric margin

$$\gamma_i = \frac{\mathbf{w}^T\mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

- Note that we can scale w and w_0. By convention we set

$$||\mathbf{w}||_2 = \frac{1}{\gamma}$$

$$\max_f \max_\gamma \gamma \text{ subject to } y_i f(x_i) \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma,\mathbf{w},w_0} \gamma \text{ subject to } y_i \frac{\mathbf{w}^T\mathbf{x}_i + w_0}{||\mathbf{w}||_2} \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma,\mathbf{w},w_0} \gamma \text{ subject to } y_i \left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq \gamma ||\mathbf{w}||_2 \text{ for } i = 1, \ldots, m$$

$$\max_{\mathbf{w},w_0} \frac{1}{||\mathbf{w}||_2} \text{ subject to } y_i \left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq 1 \text{ for } i = 1, \ldots, m$$

$$\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||_2^2 \text{ subject to } y_i \left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq 1 \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- Therefore to choose the hypothesis so the training points are far away from the decision boundary, we need to study maximize the minimum geometric margin

$$\gamma_i = \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2}$$

- Note that we can scale w and w_0. By convention we set

$$||\mathbf{w}||_2 = \frac{1}{\gamma}$$

$$\max_f \max_\gamma \gamma \text{ subject to } y_i f(x_i) \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma, \mathbf{w}, w_0} \gamma \text{ subject to } y_i \frac{\mathbf{w}^T \mathbf{x}_i + w_0}{||\mathbf{w}||_2} \geq \gamma \text{ for } i = 1, \ldots, m$$

$$\max_{\gamma, \mathbf{w}, w_0} \gamma \text{ subject to } y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) \geq \gamma ||\mathbf{w}||_2 \text{ for } i = 1, \ldots, m$$

$$\max_{\mathbf{w}, w_0} \frac{1}{||\mathbf{w}||_2} \text{ subject to } y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) \geq 1 \text{ for } i = 1, \ldots, m$$

$$\min_{\mathbf{w}, w_0} \frac{1}{2} ||\mathbf{w}||_2^2 \text{ subject to } y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) \geq 1 \text{ for } i = 1, \ldots, m$$

$$\min_{\mathbf{w}, w_0} \frac{1}{2} ||\mathbf{w}||_2^2 \text{ subject to } 0 \geq 1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- We can form the Lagrangian
- Note that we have a convex optimization problem with affine constraints
- Strict feasibility requires that the training data can be separated with a linear decision boundary

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)$$

Support Vector Machine

- We can form the Lagrangian
- Note that we have a convex optimization problem with affine constraints
- Strict feasibility requires that the training data can be separated with a linear decision boundary

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m}\alpha_i\left(1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right)\right)$$

Kuhn-Tucker Conditions

$$\nabla_{\mathbf{w}} L\left(\mathbf{w}, w_0, \alpha\right) = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i$$

Support Vector Machine

- We can form the Lagrangian
- Note that we have a convex optimization problem with affine constraints
- Strict feasibility requires that the training data can be separated with a linear decision boundary

$$L(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)$$

Kuhn-Tucker Conditions

$$\nabla_{\mathbf{w}} L(\mathbf{w}, w_0, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial \lambda_0} L(\mathbf{w}, w_0, \alpha) = -\sum_{i=1}^{m} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0$$

Support Vector Machine

- We can form the Lagrangian
- Note that we have a convex optimization problem with affine constraints
- Strict feasibility requires that the training data can be separated with a linear decision boundary

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)$$

Kuhn-Tucker Conditions

$$\nabla_{\mathbf{w}} L\left(\mathbf{w}, w_0, \alpha\right) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial \lambda_0} L\left(\mathbf{w}, w_0, \alpha\right) = -\sum_{i=1}^{m} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0$$

$\alpha_i \geq 0$ and $-y_i\left(\mathbf{w}^T \mathbf{x}_i + w_0\right) + 1 \leq 0$

Support Vector Machine

- We can form the Lagrangian
- Note that we have a convex optimization problem with affine constraints
- Strict feasibility requires that the training data can be separated with a linear decision boundary

$$L(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)$$

Kuhn-Tucker Conditions

$$\nabla_{\mathbf{w}} L(\mathbf{w}, w_0, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial \lambda_0} L(\mathbf{w}, w_0, \alpha) = -\sum_{i=1}^{m} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0$$

$\alpha_i \geq 0$ and $-y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) + 1 \leq 0 \quad \text{for } i = 1, \ldots, m$

$$\alpha_i \left(-y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right) + 1\right) = 0 \text{ for } i = 1, \ldots, m$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$L(\mathbf{w}, w_0, \alpha) = \underbrace{\frac{1}{2}\sum_{j=1}^{n} w^{(j)2}}_{\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i} + \underbrace{\sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right)\right)}_{\sum_{i=1}^{m} \alpha_i y_i = 0}$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)$$

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}||\mathbf{w}||_2^2 + \mathbf{w}^T \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m} \alpha_i$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)$$

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$L\left(\mathbf{w}, w_0, \alpha\right) = \frac{1}{2}||\mathbf{w}||_2^2 + \mathbf{w}^T \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2}||\mathbf{w}||_2^2 - ||\mathbf{w}||_2^2 - w_0 \sum_{i=1}^{m}(y_i w_0) + \sum_{i=1}^{m} \alpha_i$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$L\left(\mathbf{w}, w_0, \alpha\right) = \underbrace{\frac{1}{2}\sum_{j=1}^{n} w^{(j)2}}_{\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i} + \underbrace{\sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\mathbf{w}^T \mathbf{x}_i + w_0\right)\right)}_{\sum_{i=1}^{m} \alpha_i y_i = 0}$$

$$\begin{aligned} L\left(\mathbf{w}, w_0, \alpha\right) &= \frac{1}{2}||\mathbf{w}||_2^2 + \mathbf{w}^T \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m} \alpha_i \\ &= \frac{1}{2}||\mathbf{w}||_2^2 - ||\mathbf{w}||_2^2 - w_0 \sum_{i=1}^{m}(y_i w_0) + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + 0 + \sum_{i=1}^{m} \alpha_i \end{aligned}$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$-\frac{1}{2}\sum_{j=1}^{n} w^{(j)2} = -\frac{1}{2}\sum_{j=1}^{n}\left(\sum_{i=1}^{m}\alpha_i y_i x_i^{(j)}\right)^2$$

$$\begin{aligned} L(\mathbf{w}, w_0, \alpha) &= \frac{1}{2}||\mathbf{w}||_2^2 + \mathbf{w}^T\sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}\alpha_i \\ &= \frac{1}{2}||\mathbf{w}||_2^2 - ||\mathbf{w}||_2^2 - w_0\sum_{i=1}^{m}(y_i w_0) + \sum_{i=1}^{m}\alpha_i \\ &= -\frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + 0 + \sum_{i=1}^{m}\alpha_i \end{aligned}$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$
\begin{aligned}
-\frac{1}{2}\sum_{j=1}^{n} w^{(j)2} &= -\frac{1}{2}\sum_{j=1}^{n}\left(\sum_{i=1}^{m}\alpha_i y_i x_i^{(j)}\right)^2 \\
&= -\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{k=1}^{m}\alpha_i\alpha_k y_i y_k x_i^{(j)} x_k^{(j)}
\end{aligned}
$$

$$
\begin{aligned}
L\left(\mathbf{w}, w_0, \alpha\right) &= \frac{1}{2}||\mathbf{w}||_2^2 + \mathbf{w}^T\sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}\alpha_i \\
&= \frac{1}{2}||\mathbf{w}||_2^2 - ||\mathbf{w}||_2^2 - w_0\sum_{i=1}^{m}(y_i w_0) + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + 0 + \sum_{i=1}^{m}\alpha_i
\end{aligned}
$$

Support Vector Machine

- Substituting the expressions in the Kuhn Tucker conditions arising from the derivative of the Lagrangian, we can simplify the Lagrangian.
- We obtain a quadratic programming problem in the dual variables α

$$
\begin{aligned}
-\frac{1}{2}\sum_{j=1}^{m} w^{(j)2} &= -\frac{1}{2}\sum_{j=1}^{m}\left(\sum_{i=1}^{m}\alpha_i y_i x_i^{(j)}\right)^2 \\
&= -\frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{m}\sum_{k=1}^{m}\alpha_i\alpha_k y_i y_k x_i^{(j)} x_k^{(j)} \\
&= -\frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{m}\alpha_i\alpha_k y_i y_k \mathbf{x}_i^T\mathbf{x}_k.
\end{aligned}
$$

$$
\begin{aligned}
L(\mathbf{w}, w_0, \alpha) &= \frac{1}{2}||\mathbf{w}||_2^2 + \mathbf{w}^T\sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}(-\alpha_i y_i \mathbf{x}_i) + \sum_{i=1}^{m}\alpha_i \\
&= \frac{1}{2}||\mathbf{w}||_2^2 - ||\mathbf{w}||_2^2 - w_0\sum_{i=1}^{m}(y_i w_0) + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{j=1}^{n} w^{(j)2} + 0 + \sum_{i=1}^{m}\alpha_i
\end{aligned}
$$

Support Vector Machine

- We can solve the quadratic programming problem with a package like CVXOPT.
- Another approach called Sequential Minimal Optimization applies coordinate descent to pairs of dual variables.

Dual of Hard Margin SVM

$$\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha})$$

where

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,k} \alpha_i \alpha_k y_i y_k \mathbf{x}_i^T \mathbf{x}_k$$

subject to constrains

$$\begin{cases} \alpha_i \geq 0 \quad i = 1 \ldots m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

Support Vector Machine

- We can solve the quadratic programming problem with a package like CVXOPT.
- Another approach called Sequential Minimal Optimization applies coordinate descent to pairs of dual variables.

Dual of Hard Margin SVM

Note that we have not used the other two Kuhn Tucker conditions…and we have not computed w_0!

$$\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha})$$

where

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,k} \alpha_i \alpha_k y_i y_k \mathbf{x}_i^T \mathbf{x}_k$$

subject to constrains

$$\begin{cases} \alpha_i \geq 0 \quad i = 1 \ldots m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

Support Vectors

- Consider complementary slackness
- Note that the second and fourth situations are not possible by primal feasibility and dual feasibility.

$$\alpha_i^* \left(1 - y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right)\right) = 0$$

$$\downarrow$$

$$\begin{cases} \alpha_i^* > 0 \Rightarrow y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) = 1 \\ \alpha_i^* < 0 \\ \alpha_i^* = 0 \Rightarrow 1 - y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) < 0 \\ \alpha_i^* = 0 \Rightarrow 1 - y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) > 0 \end{cases}$$

Support Vectors

- Consider complementary slackness
- Note that the second and fourth situations are not possible by primal feasibility and dual feasibility.
- So for the hypothesis with optimal (scaled) weights

$$f^*(x) = \mathbf{w}^{*T}\mathbf{x} + w_0^*$$

$$\begin{cases} \alpha_i^* > 0 & \Rightarrow \; y_i f^*(\mathbf{x}_i) = \text{scaled margin}_i = 1 \\ 1 < y_i f^*(\mathbf{x}_i) & \Rightarrow \; \alpha_i^* = 0 \end{cases}$$

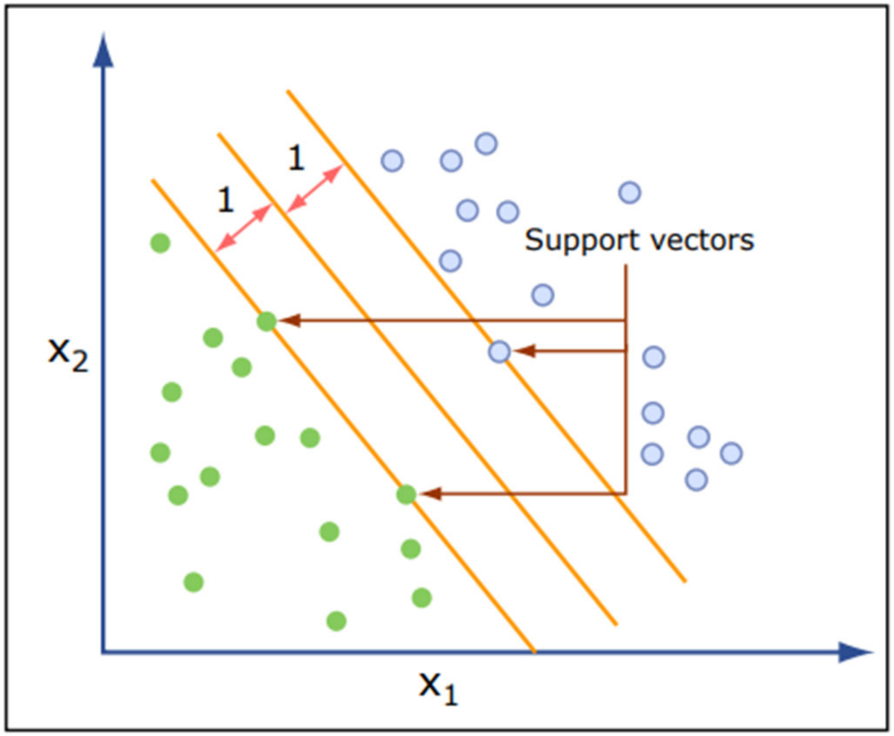
$$\begin{cases} \alpha_i^* > 0 \Rightarrow y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) = 1 \\ \cancel{\alpha_i^* < 0} \\ \alpha_i^* = 0 \Rightarrow 1 - y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) < 0 \\ \cancel{\alpha_i^* = 0 \Rightarrow 1 - y_i \left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) > 0} \end{cases}$$

Support Vectors

- Consider complementary slackness
- Note that the second and fourth situations are not possible by primal feasibility and dual feasibility.
- So for the hypothesis with optimal (scaled) weights

$$f^*(x) = \mathbf{w}^{*T}\mathbf{x} + w_0^*$$

$$\begin{cases} \alpha_i^* > 0 & \Rightarrow \quad y_i f^*(\mathbf{x}_i) = \text{scaled margin}_i = 1 \\ 1 < y_i f^*(\mathbf{x}_i) & \Rightarrow \quad \alpha_i^* = 0 \end{cases}$$



Offset Term

- For a support vector we have

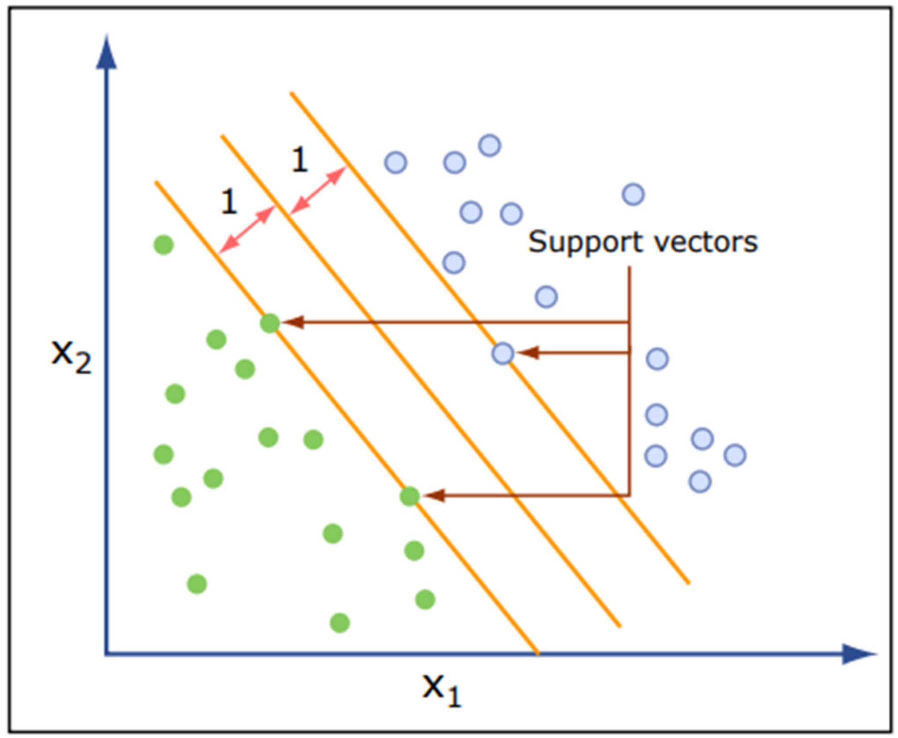
$$y_i\left(\mathbf{w}^{*T}\mathbf{x}_i + w_0^*\right) = 1$$

- For a $y_i = 1$ we obtain

$$w_0^* = 1 - \mathbf{w}^{*T}\mathbf{x}_i$$

- So we would compute the dual variables α to get w before computing w_0

$$\begin{cases} \alpha_i^* > 0 & \Rightarrow \quad y_i f^*(\mathbf{x}_i) = \text{scaled margin}_i = 1 \\ 1 < y_i f^*(\mathbf{x}_i) & \Rightarrow \quad \alpha_i^* = 0 \end{cases}$$

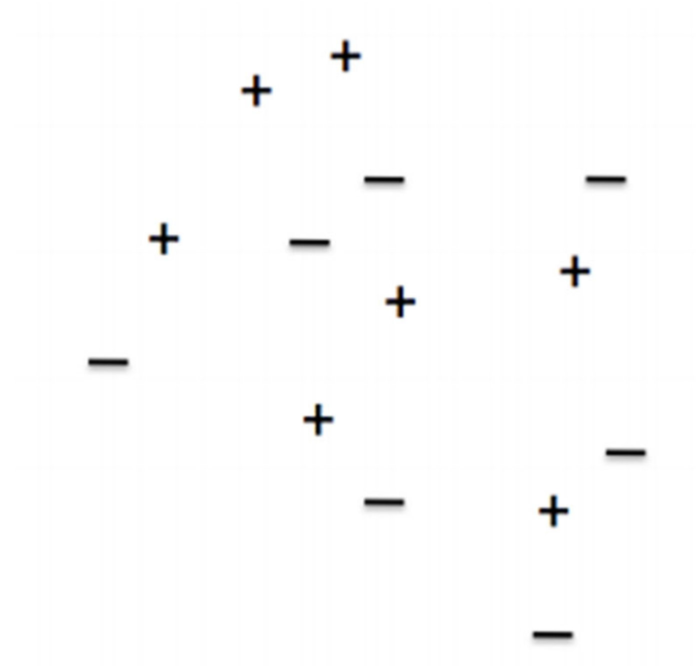


Relaxing the Constraint

- We cannot separate some training sets with a linear decision boundary
- We can relax the constraint by adding a slack variable that captures the violation of the margin constraint.

$$\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{m} \xi_i$$

$$\text{subject to } \begin{cases} y_i\left(w^T\mathbf{x}_i + w_0\right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

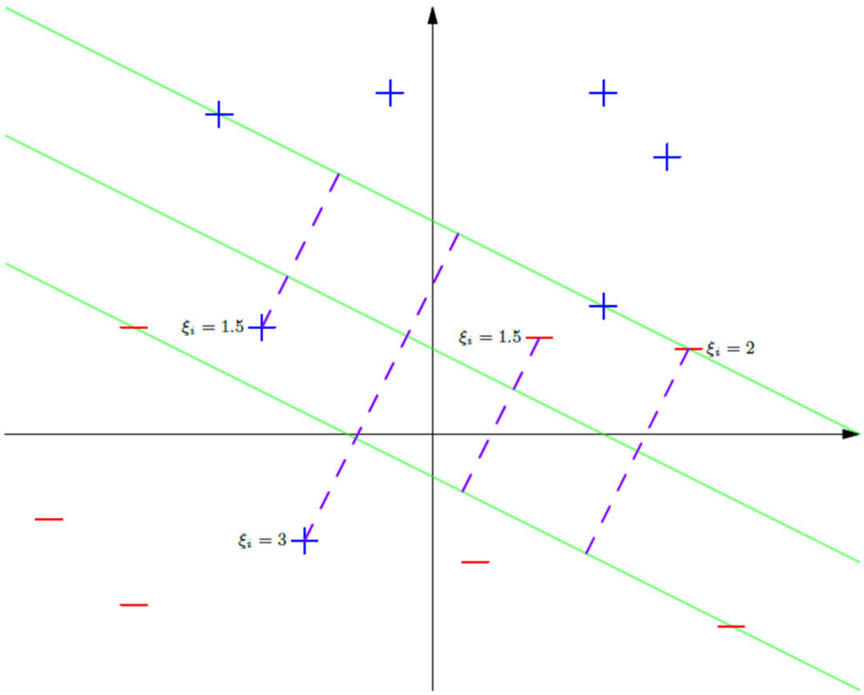


Relaxing the Constraint

- We cannot separate some training sets with a linear decision boundary
- We can relax the constraint by adding a slack variable that captures the violation of the margin constraint.

$$\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{m} \xi_i$$

$$\text{subject to} \begin{cases} y_i\left(w^T\mathbf{x}_i + w_0\right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$



Relaxing the Constraint

- Note that we can combine the two constraints because

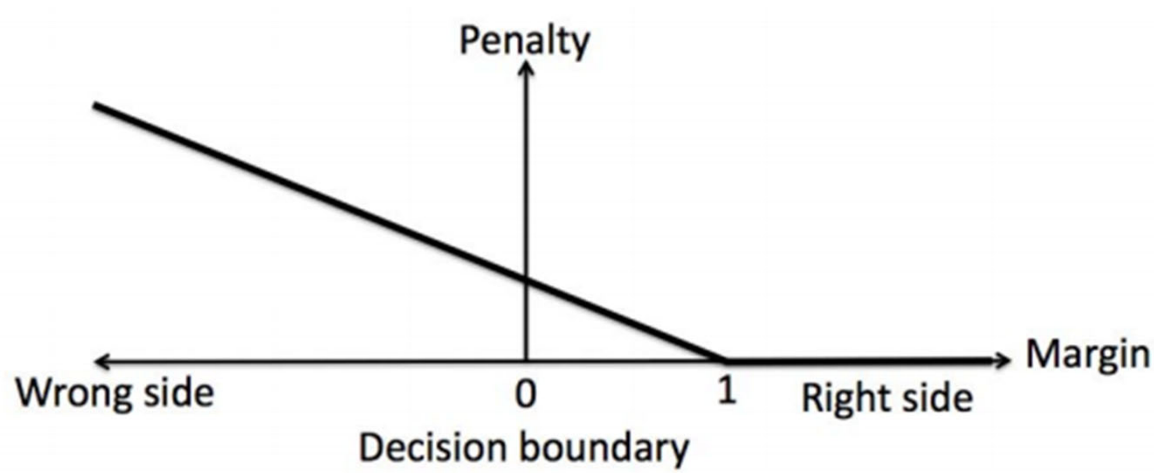
$$\begin{cases} y_i \left(w^T \mathbf{x}_i + w_0 \right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

if and only if

$$\xi_i \geq \max \left\{ 0, 1 - y_i \left(w^T \mathbf{x}_i + w_0 \right) \right\}$$

- Therefore we substitute into the objective function

$$\min_{\mathbf{w}, w_0} \frac{1}{2} ||\mathbf{w}||_2^2 + C \sum_{i=1}^{m} \max \left\{ 0, 1 - y_i \left(w^T \mathbf{x}_i + w_0 \right) \right\}$$



Support Vector Machine

- We can solve the quadratic programming problem with a package like CVXOPT.
- Another approach called Sequential Minimal Optimization applies coordinate descent to pairs of dual variables.

Dual of Soft Margin SVM

$$\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha})$$

where

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,k} \alpha_i \alpha_k y_i y_k \mathbf{x}_i^T \mathbf{x}_k$$

subject to constrains

$$\begin{cases} 0 \leq \alpha_i \leq C & i = 1 \ldots m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

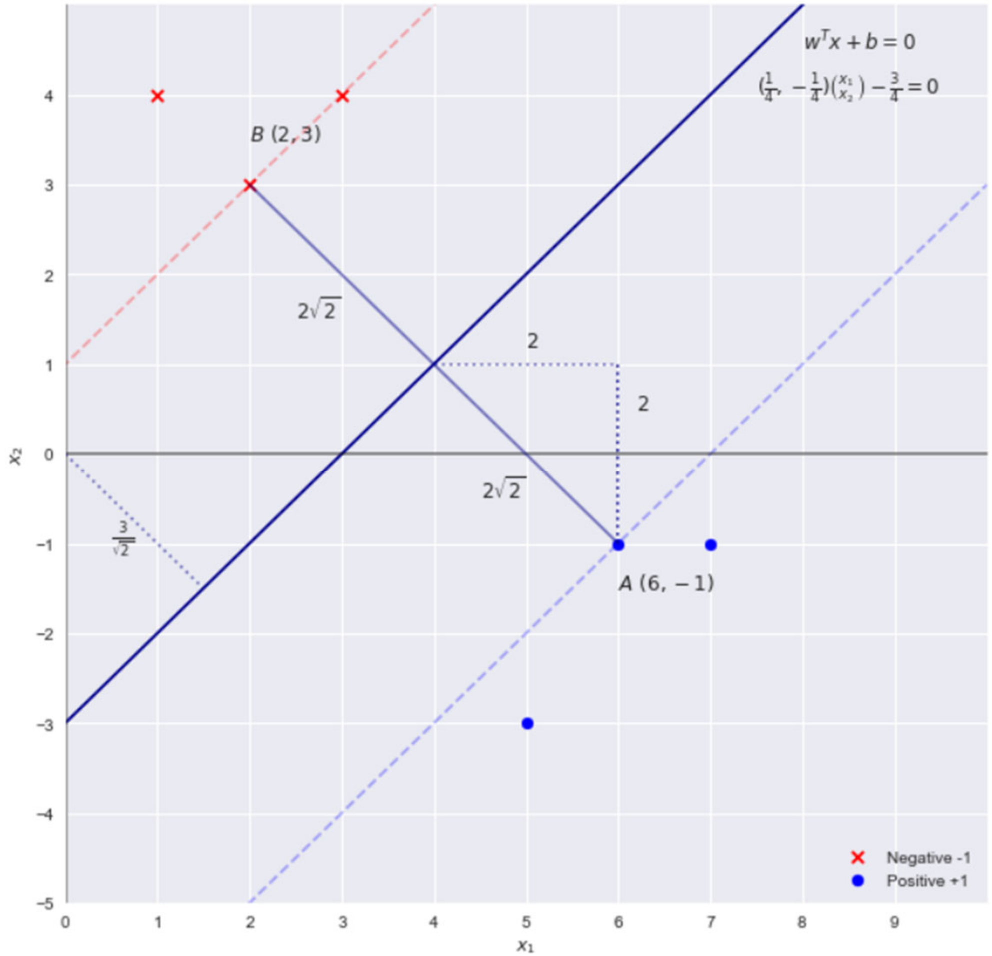
Exercise

```
import numpy as np
from sklearn.svm import SVC

X = np.array([[3,4],[1,4],[2,3],[6,-1],[7,-1],[5,-3]] )
y = np.array([-1,-1, -1, 1, 1 , 1 ])

clf = SVC(C = 1e5, kernel = 'linear')
clf.fit(X, y)
```

```
SVC(C=100000.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

```
clf.support_vectors_
```

```
array([[ 2.,  3.],
       [ 6., -1.]])
```



Summary

- Convexity
  - Sets, Functions
- Duality
  - Min-Max Inequality
  - Complementary Slackness
- Support Vector Machines
  - Hard Margin, Soft Margin
- Understanding Support Vector Machines through Duality

- Goals
  - How does the hinge loss increase margins? Why would large margins help us?
  - What are some advantages of the dual formulation of a minimization problem? What insights can we gain from the SVM dual problem?
  - What is a support vector? Why is SVM sparse in the data?

Questions

Kernel Methods

- Questions on Piazza?
  - Please provide your feedback
- Question for You!

The objective depends on dot products. Could we replace with other products?

$$\sup_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \ldots, n.$$