$\LaTeX$ command declarations here.

EECS 545: Machine Learning
Lecture 10: Support Vector Machines

- Instructor: **Jacob Abernethy**
- Date: February 10, 2015

Lecture Exposition Credit: Ben, Saket, & Valli

Outline

- Prerequisite
- Maximum Margin Classifier
  - Problem Formulation
  - Linear Separability
  - Optimal Soft Margin Hyperplane (OSMH)
- Duality
  - Concepts
  - KKT Conditions
- Optimal Soft Margin Hyperplane
  - Dual Problem Formulation
  - Support Vectors
  - Kernelization and SVM

Reading List

- Required:
  - **[PRML]**, §7.1: Maximum Margin Classifiers
- Optional:
  - **[CS229]**, Lecture Notes 03: Support Vector Machines & Kernels (http://cs229.stanford.edu/notes/cs229-notes3.pdf)

> In this lecture, we will introduce a classifier with heuristical idea, which is finding a separating hyperplane such that the distance from any datapoint to it is maximized. This classifier is called *maximum margin classifier*. The parameter of this classifier can be obtained by solving a simple optimization problem. The disadvantage of maximum margin classifier is that it doesn't work for dataset that is not linearly separable. To deal with this, we will do some slackness and convert original optimal hard margin hyperplane problem into *optimal soft margin hyperplane (OSMH)* problem. Instead of solving OSMH problem directly, we will show how to solve it by solving its dual problem. This can be advantageous because sometimes dual problem can be easier to solve than original problem. Don't be afraid if you don't have much knowledge about convex duality, some basics of duality will be reviewed. With these knowledge, we will show how to formulate the dual problem and how to obtain primal solution out of dual solution. Then, some analysis about support vectors will be proposed both analytically and geometrically. Finally, how to apply kernel trick to our classifier is shown. This is critical because kernel can map original feature into some higher dimensional feature space just like we talked about in last lecture.

Preliminaries
Vapnik's Principle:

> "When solving a problem of interest, do not solve a more general problem as an intermediate step."
>
> "Don't solve a harder problem than you have to"

Review: Linear Classifiers

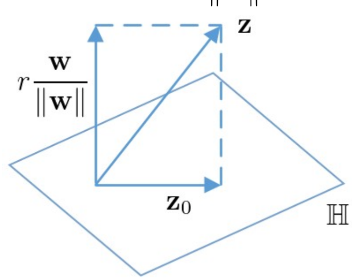
- Linear classifiers make decisions based on a linear (or more generally affine) combination of features.
  - *Generative*: Require estimation of (conditional) densities or mass functions.
    - GDA, Naive Bayes
  - *Discriminative*: Often much easier to just determine the "decision boundary."
    - Logistic Regression, Perceptron
- Based on *Vapnik's Principle*, we will focus on a discriminative classifier **Support Vector Machine (SVM)** in this lecture.

Preliminaries: Hyperplane

- **Hyperplane** is an affine subspace one dimension fewer than its ambient space.
  - The hyperplanes of a 2-D space are 1-D lines.
  - The hyperplanes of a 3-D space are 2-D planes.
- Mathematically, a **hyperplane** is of the form
$$\mathbb{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\}$$
where $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ and d is the number of features.

Preliminaries: Point-Plane Distance

- Given a hyperplane $\mathbb{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\}$ and a point $\mathbf{z} \notin \mathbb{H}$, what is the point-plane distance from $\mathbf{z}$ to $\mathbb{H}$?
- We can write $\mathbf{z}$ as:
$$\mathbf{z} = \mathbf{z}_0 + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$
- We have decomposed $\mathbf{z}$ into two components:
$$\mathbf{z}_0 \in \mathbb{H} \qquad r\frac{\mathbf{w}}{\|\mathbf{w}\|} \perp \mathbb{H}$$



- So the distance is then given by $|r|$!

Preliminaries: Point-Plane Distance

- Calculating $|r|$:
$$\begin{aligned}\mathbf{w}^T\mathbf{z} + b &= \mathbf{w}^T\left(\mathbf{z}_0 + r\cdot\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) + b \\ &= \underbrace{\mathbf{w}^T\mathbf{z}_0 + b}_{=0} + \mathbf{w}^T\left(r\cdot\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) \\ &= r\frac{\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|} \\ &= r\|\mathbf{w}\|\end{aligned}$$
Therefore, point-plane distance from point $\mathbf{z}$ to plane $\{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\}$ is
$$\boxed{|r| = \frac{|\mathbf{w}^T\mathbf{z} + b|}{\|\mathbf{w}\|}}$$

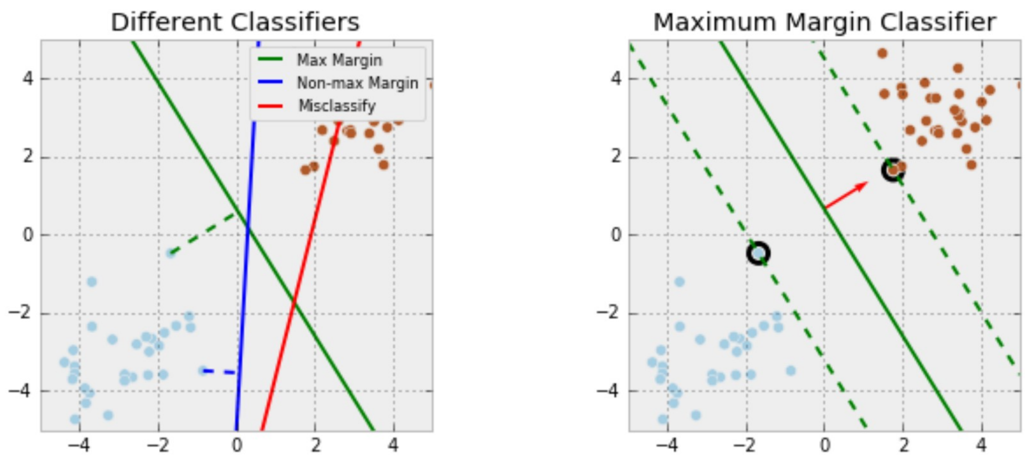
Preliminaries

- **Separating Hyperplanes**
  - Provide a way of solving 2-class classification problems.
  - **Idea:** divide the vector space $\mathbb{R}^d$ where d is the number of features into 2 "decision regions" with a $\mathbb{R}^{d-1}$ subspace (a hyperplane).
    - Eg. Logistic Regression, Perceptron, LDA
  - As with other linear classifiers, classification could be achieved by
$$y = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$
    Note: We may use $\mathbf{x}$ and $\phi(\mathbf{x})$ interchangeably to denote features.
- **(Functional) Margin**
  - The distance from a separating hyperplane to the *closest* datapoint of *any* class.
$$\rho = \rho(\mathbf{w}, b) = \min_{i=1,\dots,n} \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|}$$
    where $\mathbf{x}_i$ is the ith datapoint from the training set.

Maximum Margin Classifier

Maximum Margin Classifier

- **Max. Margin Classifiers:** separate data by looking for the hyperplane that maximizes the margin.



- The length of dotted segment in left plot is margin ρ.
- Properties
  - tends to guarantee better generalization performance.
  - more robust to noise
  - misclassification unlikely with a wide margin between classes.

Finding the Max-Margin Hyperplane

- For dataset $\{\mathbf{x}_i, t_i\}_{i=1}^n$, maximum margin separating hyperplane is the solution of
$$\begin{aligned} \underset{\mathbf{w},b}{\text{maximize}} \quad & \min_{i=1,\dots,n} \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|} \\ \text{subject to} \quad & t_i(\mathbf{w}^T\mathbf{x}_i + b) > 0 \quad \forall i \end{aligned}$$
  of which the constraint ensures every training data is correctly classified
  - Note that $t_i \in \{+1, -1\}$ is the label of ith training data
- This problem guarantees optimal hyperplane, but the solution $\mathbf{w}$ and b is **not** unique :
  - we could scale both $\mathbf{w}$ and b by arbitrary scalar without affecting $\mathbb{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\}$
  - we have infinite sets of solutions

Ensuring Uniqueness of Solution

- For the optimal hyperplane $\mathbf{H}$ and *one* set of $\mathbf{w}$ and b, let
$$m = \min_{i=1,\dots,n} \left|\mathbf{w}^T\mathbf{x}_i + b\right|$$
- If we scale $\mathbf{w}$ and b by $\frac{1}{m}$, we could get a new and unique set of $\mathbf{w}$ and b such that
  - $\min_{i=1,\dots,n} \left|\mathbf{w}^T\mathbf{x}_i + b\right| = 1$ and margin becomes $\frac{1}{\|\mathbf{w}\|}$
  - $t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ for some i and $t_i(\mathbf{w}^T\mathbf{x}_i + b) > 1$ for other i's
- So, conversely, if we restrict $t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ for some i and $t_i(\mathbf{w}^T\mathbf{x}_i + b) > 1$ for other i's, we have
  - $\min_{i=1,\dots,n} \left|\mathbf{w}^T\mathbf{x}_i + b\right| = 1$ and margin becomes $\frac{1}{\|\mathbf{w}\|}$
  - problem will have *unique* solution $\mathbf{w}$ and b
- Original problem can be converted into

$$\begin{array}{ll} \underset{\mathbf{w},b}{\text{maximize}} & \min\limits_{i=1,\dots,n} \dfrac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|} \\ \text{subject to} & t_i(\mathbf{w}^T x_i + b) > 0 \quad \forall i \end{array} \implies \begin{array}{lll} \underset{\mathbf{w},b}{\text{maximize}} & \dfrac{1}{\|\mathbf{w}\|} & \\ \text{subject to} & t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1 & \text{for some } i \\ & t_i(\mathbf{w}^T\mathbf{x}_i + b) > 1 & \text{for other } i \end{array}$$

Restatement of Optimization Problem

- Simplifying further, we have

$$\begin{array}{ll} \underset{\mathbf{w},b}{\text{maximize}} & \dfrac{1}{\|\mathbf{w}\|} \\ \text{subject to} & t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1 \text{ for some } i \\ & t_i(\mathbf{w}^T\mathbf{x}_i + b) > 1 \text{ for other } i \end{array} \implies \begin{array}{ll} \underset{\mathbf{w},b}{\text{minimize}} & \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i \end{array}$$

Linear Separability

- Two classes of data are said to be **linearly separable** if there exists a hyperplane that separates them without any errors.
- So far, we have looked at primarily linearly separable data where a single hyperplane will do for classification.
- We can extend on this notion to a multiclass scenario by considering data to be linearly separable if there exists a set of hyperplanes that can classify each class of examples from the rest (again without errors).
- **BUT**, how to deal with data that **aren't** linearly separable?
  - Use "slack" variables that allow for misclassification and penalize misclassification.
    - This is the protagonist of this lecture.
    - Hyperplane obtained in this way is called **optimal soft-margin hyperplane (OSMH)**
  - Extend linear classifiers with kernels.

Optimal Soft-Margin Hyperplane (OSMH)

- To deal with non-linearly separable case, we could introduce slack variables:

$$\begin{array}{ll} \underset{\mathbf{w},b}{\text{minimize}} & \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i \end{array} \implies \begin{array}{ll} \underset{\mathbf{w},b,\xi}{\text{minimize}} & \dfrac{1}{2}\|\mathbf{w}\|^2 + \dfrac{C}{n}\sum_{i=1}^n \xi_i \\ \text{subject to} & t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{array}$$

- New term $\frac{C}{n}\sum_{i=1}^n \xi_i$ penalizes errors and accounts for the influence of outliers through a constant $C \geq 0$ (0 would lead us back to the hard margin case) and $\xi = [\xi_1, \dots, \xi_n]$ are the slack variables.
- **Motivation:**
  - The **objective function** ensures margin is large *and* the margin violations are small
  - The **first set of constraints** ensures classifier is doing well
    - similar to the prev. max-margin constraint, except we now allow for slack
  - The **second set of constraints** ensure slack variables are non-negative.
    - keeps the optimization problem from *"diverging"*
- Instead of solving this problem directly, we prefer to solve its **dual problem**.
  - Sometimes, dual problem is easier to solve than original problem
- Next, we will review basics of duality

Review: Duality

Lagrangian

- Consider a **constrained optimization problem**

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, n \end{aligned}$$

  - **Feasible set** is defined as $C \triangleq \{\mathbf{x}\,|g_i(\mathbf{x}) \le 0, h_j(\mathbf{x}) = 0,\ i = 1, \ldots, m,\ j = 1, \ldots, n\}$. C is convex if $g_i(\mathbf{x})$ is convex and $h_j(\mathbf{x})$ is affine
- The Lagrangian is then given by

$$L(\mathbf{x}, \alpha, \beta) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \beta_j h_j(\mathbf{x})$$

  Here, $\alpha \in \mathbb{R}^m$ and $\beta \in R^n$ are the **Lagrange Multipliers / Dual Variables**

Lagrangian Primal

- For better visualization, we reiterate the original problem and Lagrangian in this slide

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \le 0 \quad h_j(\mathbf{x}) = 0 \end{aligned} \qquad L(\mathbf{x}, \alpha, \beta) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \beta_j h_j(\mathbf{x})$$

- The **primal objective** is defined as

$$L_P(\mathbf{x}) \triangleq \max_{\alpha, \beta : \alpha_i \ge 0} L(\mathbf{x}, \alpha, \beta) = \begin{cases} f(x) & \text{if } x \in C \\ +\infty & \text{if } x \notin C \end{cases}$$

  The second equality holds because to maximize $L(\mathbf{x}, \alpha, \beta)$:
  - for $\mathbf{x} \in C$ (i.e. $g_i(\mathbf{x}) \le 0$ and $h_j(\mathbf{x}) = 0$), letting $\alpha = 0, \beta = 0$ gives maxima $f(\mathbf{x})$
  - for $\mathbf{x} \notin C$ (i.e. $g_i(\mathbf{x}) > 0$ or $h_j(\mathbf{x}) \ne 0$), letting $\alpha \to +\infty, \beta = 0$ then $L(\mathbf{x}, \alpha, \beta) \to +\infty$
- The **primal optimization problem** is defined as

$$\min_{\mathbf{x}}\ L_P(\mathbf{x}) = \min_{\mathbf{x}} \max_{\alpha, \beta : \alpha_i \ge 0} L(\mathbf{x}, \alpha, \beta) = \min_{\mathbf{x}} \begin{cases} f(x) & \text{if } x \in C \\ +\infty & \text{if } x \notin C \end{cases} = \min_{\mathbf{x} \in C}\ f(\mathbf{x})$$

  which is equivalent to the original optimization problem!

Lagrangian Dual

- The **primal optimization problem**

$$\min_{\mathbf{x}}\ L_P(\mathbf{x}) = \min_{\mathbf{x}} \max_{\alpha, \beta : \alpha_i \ge 0} L(\mathbf{x}, \alpha, \beta)$$

  is just original problem and of no interest to us.
- **BUT**, swapping the inner and outer optimization, we get **dual optimization problem**

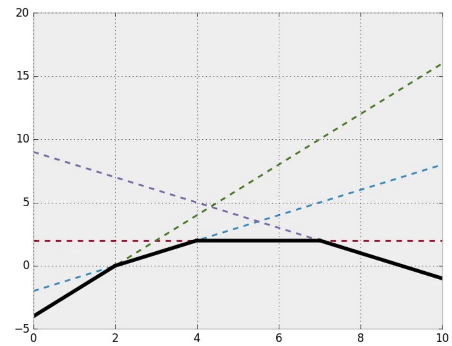
$$\max_{\alpha, \beta : \alpha_i \ge 0} \min_{\mathbf{x}} L(\mathbf{x}, \alpha, \beta)$$

- The **dual objective** is defined as

$$L_D(\alpha, \beta) \triangleq \min_{\mathbf{x}} L(\mathbf{x}, \alpha, \beta)$$

Remark

- Lagrangian$L(\mathbf{x}, \alpha, \beta) = f(\mathbf{x}) + \sum_{i=1}^m \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^n \beta_j h_j(\mathbf{x})$ is affine respect to α and β
- Therefore, dual objective $L_D(\alpha, \beta) = \min_{\mathbf{x}} L(\mathbf{x}, \alpha, \beta)$ is a piece-wise minimum of affine functions and it's concave.
- Why Concave? Bold lines in the plots are piece-wise minimum of affine functions (1D case) and it's obviously concave



- Since $L_D(\alpha, \beta)$ is concave, the maximization in dual problem $\max_{\alpha, \beta: \alpha_i \geq 0} L_D(\alpha, \beta)$ can be achieved.

Strong and Weak Duality

- Let $\mathbf{x}^*$ and p^* denote the solution and optimal value of **primal**/original problem
$$p^* = \min_{\mathbf{x}} L_p(\mathbf{x}) = L_p(\mathbf{x}^*)$$
- Let α^*, β^* and d^* denote the solution and optimal value of **dual** problem
$$d^* = \max_{\alpha, \beta: \alpha_i \geq 0} L_D(\alpha, \beta) = L_D(\alpha^*, \beta^*)$$
- **Weak duality** (always true): $d^* \leq p^*$
- **Strong duality** (under some conditions): $p^* = d^*$
- Strong duality is surely more interesting, which allows us to solve original problem by solving its dual!
- Even if we can solve for α^*, β^* and d^*, some questions are still left open
  - How to obtain $\mathbf{x}^*$ out of α^* and β^* (Necessary conditions of strong duality)?
  - When does strong duality hold (Sufficient conditions of strong duality)?

A Quick Summary

- Here is a table summarizing all the concepts we just covered

	Primal	**Dual**
Original Problem	$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x})$ $\text{subject to} \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m$ $h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, n$	
Lagrangian	$L(\mathbf{x}, \alpha, \beta) = f(\mathbf{x}) + \sum_{i=1}^m \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^n \beta_j h_j(\mathbf{x})$	
Objective	$L_P(\mathbf{x}) \triangleq \max_{\alpha, \beta: \alpha_i \geq 0} L(\mathbf{x}, \alpha, \beta)$	$L_D(\alpha, \beta) \triangleq \min_{\mathbf{x}} L(\mathbf{x}, \alpha, \beta)$
Problem	$\min_{\mathbf{x}} \; L_P(\mathbf{x})$	$\max_{\alpha, \beta: \alpha_i \geq 0} L_D(\alpha, \beta)$
Solution	$\mathbf{x}^*$	α^*, β^*
Optimal Value	p^*	d^*
Weak Duality	$p^* \geq d^*$	
Strong Duality	$p^* = d^*$	

Necessary Conditions of Strong Duality—KKT Conditions

- If **strong duality** holds, i.e. $p^* = d^*$, then **Karush-Kuhn-Tucker (KKT) Conditions** hold:
  - Stationarity
$$\nabla_{\mathbf{x}} L(\mathbf{x}, \alpha^*, \beta^*)\,|_{\mathbf{x}=\mathbf{x}^*} = \nabla_{\mathbf{x}} f(\mathbf{x}^*) + \sum_i \alpha_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*) + \sum_j \beta_j^* \nabla_{\mathbf{x}} h_j(x^*) = 0$$
  - Primal Feasibility
$$\forall\, i, g_i(\mathbf{x}^*) \le 0 \qquad \forall\, j, h_j(\mathbf{x}^*) = 0$$
  - Dual Feasibility
$$\forall\, i, \alpha_i^* \ge 0$$
  - Complimentary Slackness
$$\forall\, i, \alpha_i^* g_i(\mathbf{x}^*) = 0$$
- Proof for Stationarity and Complimentary Slackness is in the notes!
- KKT conditions enable us to simplify dual problem and obtain $\mathbf{x}^*$ out of α^* and β^*

> **Remark**
>
> - Proof for KKT conditions
>   - We have
> $$\begin{aligned} f(\mathbf{x}^*) = p^* = d^* &\qquad \text{(By Strong duality)} \\ &= L_D(\alpha^*, \beta^*) \\ &= \min_{\mathbf{x}}\; f(\mathbf{x}) + \textstyle\sum_i \alpha_i^* g_i(\mathbf{x}) + \sum_j \beta_j^* h_j(\mathbf{x}) \\ &\le f(\mathbf{x}^*) + \sum_i \alpha_i^* g_i(\mathbf{x}^*) + \sum_j \beta_j^* h_j(\mathbf{x}^*) \\ &\le f(\mathbf{x}^*) \qquad \text{(Since } g_i(\mathbf{x}^*) \le 0, h_j(\mathbf{x}^*) = 0) \end{aligned}$$
>   The first and last term form $f(\mathbf{x}^*) \le f(\mathbf{x}^*)$ which indicates all inequalities are actually **equalities**!

> - Therefore, we have
> $$\begin{aligned} &\min_{\mathbf{x}}\; f(\mathbf{x}) + \textstyle\sum_i \alpha_i^* g_i(\mathbf{x}) + \sum_j \beta_j^* h_j(\mathbf{x}) \\ &= f(\mathbf{x}^*) + \textstyle\sum_i \alpha_i^* g_i(\mathbf{x}^*) + \sum_j \beta_j^* h_j(\mathbf{x}^*) \\ &= f(\mathbf{x}^*) \end{aligned}$$
>   - The equality of the last two lines indicates $\forall i, \alpha_i^* g_i(\mathbf{x}^*) = 0$. So complimentary slackness condition is proved.
>   - The first equality implies $\mathbf{x}^*$ is a minimizer of $L(\mathbf{x}, \alpha^*, \beta^*)$ $\text{w.r.t. } \mathbf{x}$. Therefore, $\nabla_{\mathbf{x}} L(\mathbf{x}, \alpha^*, \beta^*) = 0$. So stationarity condition is proved.

Sufficient Conditions of Strong Duality

- **Strong duality** holds if **ANY** of the conditions below holds
  - **KKT conditions** hold for primal solution $\mathbf{x}^*$ and dual solution α^* and β^*
  - Original problem is **convex optimization problem** ($f(\mathbf{x})$ and $g_i(\mathbf{x})$ are convex and $h_j(\mathbf{x})$ are affine).
  - **Slater's Condition**: $\exists \mathbf{x}$ s.t. $g_i(\mathbf{x}) < 0$ and $h_j(\mathbf{x}) = 0$

Back to SVM

The OSMH Optimization Problem and Lagrangian

- Recall the OSMH problem is

$$
\begin{aligned}
\underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\
\text{subject to} \quad & -\big(t_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i\big) \le 0 \quad \forall i \\
& -\xi_i \le 0 \quad \forall i
\end{aligned}
$$

- **Strong duality** holds because OSMH optimization is a **convex optimization problem**!
  - objective function is quadratic and inequality constraints are affine (and hence convex).
- So **KKT conditions** hold !
- Next we will show how to formulate its **dual problem** and solve for $\mathbf{w}^*$ and b^* by solving for dual variable α^* and β^*
- The **Lagrangian** is given by (Note that **primal variables** are $\{\mathbf{w}, b, \xi\}$.)

$$
\begin{aligned}
& L(\mathbf{w}, b, \xi, \alpha, \beta) \\
=& 0.5\|\mathbf{w}\|^2 + C/n\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left[t_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{n}\beta_i\xi_i \\
=& 0.5\|\mathbf{w}\|^2 - \mathbf{w}^T\sum_{i=1}^{n}\alpha_i t_i\mathbf{x}_i - \sum_{i=1}^{n}\alpha_i t_i b + \sum_{i=1}^{n}\left(C/n - \alpha_i - \beta_i\right)\xi_i + \sum_{i=1}^{n}\alpha_i
\end{aligned}
$$

OSMH: Dual Objective

- **Stationarity KKT condition** $\nabla_{\mathbf{x}} L(\mathbf{x}, \alpha^*, \beta^*)\,|_{\mathbf{x}=\mathbf{x}^*} = 0$ says optimal solution $\{\alpha^*, \beta^*\}$ should satisfy

$$
\partial L\,/\,\partial b = 0 \Rightarrow \boxed{\sum_{i=1}^{n}\alpha_i t_i = 0}
$$

$$
\partial L\,/\,\partial \xi_i = 0 \Rightarrow \boxed{C/n - \alpha_i - \beta_i = 0}
$$

- **Dual objective** is given by

$$
\begin{aligned}
L_D(\alpha, \beta) &= \min_{\mathbf{w},b,\xi}\; L(\mathbf{w}, b, \xi, \alpha, \beta) \\
&= \min_{\mathbf{w},b,\xi} 0.5\|\mathbf{w}\|^2 - \mathbf{w}^T\sum_{i=1}^{n}\alpha_i t_i\mathbf{x}_i - \sum_{i=1}^{n}\alpha_i t_i b + \sum_{i=1}^{n}\left(C/n - \alpha_i - \beta_i\right)\xi_i + \sum_{i=1}^{n}\alpha_i \\
&= \min_{\mathbf{w}} 0.5\|\mathbf{w}\|^2 - \mathbf{w}^T\sum_{i=1}^{n}\alpha_i t_i\mathbf{x}_i + \sum_{i=1}^{n}\alpha_i \\
&= -0.5\left\|\sum_{i=1}^{n}\alpha_i t_i\mathbf{x}_i\right\|^2 + \sum_{i=1}^{n}\alpha_i \\
&= \boxed{-0.5\sum_{i,j=1}^{n}\alpha_i\alpha_j t_i t_j\mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i}
\end{aligned}
$$

- 3rd equality holds because we plug in stationarity conditions
- 4th equality holds because $\mathbf{w} = \sum_{i=1}^{n}\alpha_i t_i\mathbf{x}_i$ is the solution to quadratic minimization in the 3rd line.

OSMH: Dual Problem

- Let's wrap it up! **Dual problem** is given by

$$
\begin{aligned}
\underset{\alpha,\beta}{\text{maximize}} \quad & -0.5\sum_{i,j=1}^{n}\alpha_i\alpha_j t_i t_j\mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i \\
\text{subject to} \quad & \alpha_i \ge 0, \beta_i \ge 0 \quad \forall i \\
& \sum_{i=1}^{n}\alpha_i t_i = 0 \\
& C/n - \alpha_i - \beta_i = 0 \quad \forall i
\end{aligned}
$$

  - Inequality constraints are due to non-negativeness of dual variables
  - Equality constraints are due to stationarity conditions
- NOTE: **Dual variable** α and β are both for *inequality* constraints! This is why $\beta_i \ge 0$ is also required.
  - Remember β was for equality constraints in previous slides? Sorry for this misnomer!
- Eliminating β, we get the final quadratic programming problem

$$
\boxed{
\begin{aligned}
\underset{\alpha,\beta}{\text{maximize}} \quad & -0.5\sum_{i,j=1}^{n}\alpha_i\alpha_j t_i t_j\mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i \\
\text{subject to} \quad & 0 \le \alpha_i \le C/n \quad \forall i \\
& \sum_{i=1}^{n}\alpha_i t_i = 0
\end{aligned}
}
$$

OSMH: Solution $\mathbf{w}^*$ and b^*

- Obtain $\mathbf{w}^*$
  - Applying **stationarity KKT condition** to $\mathbf{w}$, we have
$$\partial L \,/\, \partial \mathbf{w} = 0 \Rightarrow \boxed{\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* t_i \mathbf{x}_i}$$
- Obtain b^*
  - Recall we have constraints in last slide
$$\alpha_i \geq 0,\ \beta_i \geq 0,\ \alpha_i + \beta_i = C/n$$
    So for any $0 < \alpha_i^* < C/n$, we must have $\beta_i^* > 0$
  - Applying **complimentary slackness KKT condition**, we have
$$\beta_i^* \xi_i^* = 0 \qquad \alpha_i^*(1 - \xi_i^* - t_i({\mathbf{w}^*}^T \mathbf{x}_i + b^*)) = 0$$
    So for any $0 < \alpha_i^* < C/n$, we further have
$$\xi_i^* = 0 \qquad 1 - \xi_i^* - t_i({\mathbf{w}^*}^T \mathbf{x}_i + b^*) = 0$$
    which implies $t_i({\mathbf{w}^*}^T \mathbf{x}_i + b^*) = 1$
  - Since $t_i \in \{\pm 1\}$, we have ${\mathbf{w}^*}^T \mathbf{x}_i + b^* = t_i$. Therefore, for any $0 < \alpha_i^* < C/n$, we could solve for b^*:
$$\boxed{b^* = t_i - {\mathbf{w}^*}^T \mathbf{x}_i}$$

OSMH: Support Vectors

- Applying **stationarity KKT condition** to $\mathbf{w}$, we know optimal $\mathbf{w}^*$ and α^* should satisfy
$$\partial L \,/\, \partial \mathbf{w} = 0 \Rightarrow \boxed{\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* t_i \mathbf{x}_i}$$
- So, the optimal normal vector of separating hyperplane is a *linear combination* of datapoints!
- Applying **complimentary slackness KKT condition**, we have
$$\alpha_i^*(1 - \xi_i^* - t_i({\mathbf{w}^*}^T \mathbf{x}_i + b^*)) = 0$$
  - If $\mathbf{x}_i$ satisfies $t_i({\mathbf{w}^*}^T \mathbf{x}_i + b^*) = 1 - \xi_i^*$,
    - then α^* could be nonzero and $\mathbf{x}_i$ will contribute to $\mathbf{w}^*$
    - we call $\mathbf{x}_i$ **support vector (SV)**
  - If $\mathbf{x}_i$ cannot satisfy $t_i({\mathbf{w}^*}^T \mathbf{x}_i + b^*) = 1 - \xi_i^*$
    - then α^* must be 0 and $\mathbf{x}_i$ has no effect on $\mathbf{w}^*$
    - $\mathbf{x}_i$ is **NOT** a SV
- The above means $\mathbf{w}^*$ depends **ONLY** on support vectors! This is why we call **support vector machine**.
- Now let's analyze what datapoints can be support vectors geometrically

OSMH: Geometric Interpretation of SV

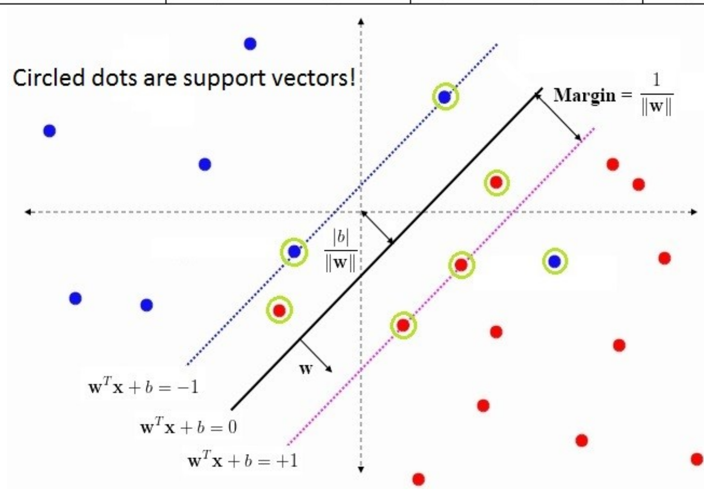
- Recall original OSMH problem is
$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & 0.5\|\mathbf{w}\|^2 + C/n \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & t_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned}$$
  - For optimal ξ_i^*, at least one of
$$t_i(\mathbf{w}^T \mathbf{x}_i + b) = 1 - \xi_i^*$$
$$\xi_i^* = 0$$
    mush hold
  - Because if $t_i(\mathbf{w}^T \mathbf{x}_i + b) > 1 - \xi_i$ and $\xi > 0$ we can reduce ξ_i to get lower objective value without violating constraints!

Remark

- Since for each i, at least one equality in
$$t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i^* \qquad \xi_i^* \geq 0$$
must hold
- So if $\xi > 0$, we have
$$\xi_i = 1 - t_i(\mathbf{w}^T\mathbf{x}_i + b)$$
- Therefore, ξ_i could take the value of either $1 - t_i(\mathbf{w}^T\mathbf{x}_i + b)$ or 0, which has provided another approach to solve original problem:
$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & 0.5\|\mathbf{w}\|^2 + C/n\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \implies \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad 0.5\|\mathbf{w}\|^2 + C/n\sum_{i=1}^n \max\left(0, 1 - t_i(\mathbf{w}^T\mathbf{x}_i\right.$$
- This is an unconstrained problem! And we could obtain solution using gradient descent! (More precisely, using subgradient due to the $\max(\cdot)$)
- You encounter this in your HW :)

- We already know
  - For particular i, **at least one equality** of the following two inequalities must hold
$$t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i^* \qquad \xi_i^* \geq 0$$
  - If first equality holds, then $\mathbf{x}_i$ is SV. Otherwise, it's not SV
- Based on above results, for data $\mathbf{x}_i$, we have

Location		Two Constraints		Whether SV
$\mathbf{x}_i$ is **outside the margin**	$t_i(\mathbf{w}^{*T}\mathbf{x}_i + b^*) > 1$	$t_i(\mathbf{w}^T\mathbf{x}_i + b) > 1 - \xi_i^*$	$\xi_i^* = 0$	NOT SV
$\mathbf{x}_i$ is **on the margin**	$t_i(\mathbf{w}^{*T}\mathbf{x}_i + b^*) = 1$	$t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1 - \xi_i^*$	$\xi_i^* = 0$	SV
$\mathbf{x}_i$ is **within the margin**	$0 \leq t_i(\mathbf{w}^{*T}\mathbf{x}_i + b^*) < 1$	$t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1 - \xi_i^*$	$1 \geq \xi_i^* > 0$	SV
$\mathbf{x}_i$ is **misclassified**	$t_i(\mathbf{w}^{*T}\mathbf{x}_i + b^*) < 0$	$t_i(\mathbf{w}^T\mathbf{x}_i + b) = 1 - \xi_i^*$	$\xi_i^* > 1$	SV



OSMH: Support Vector Machines

- The dual problem and final classifier only involve the data via inner products.
  - We can apply the **kernel trick** and kernelize the OSMH problem.
  - The resulting classifier is known as a **Support Vector Machine**.
- Let $k(\cdot,\cdot)$ be an inner product kernel
- The dual problem is given by

$$\begin{aligned} \underset{\alpha,\beta}{\text{maximize}} \quad & -0.5\sum_{i,j=1}^{n}\alpha_i\alpha_j t_i t_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \le \alpha_i \le C/n \quad \forall i \\ & \sum_{i=1}^{n}\alpha_i t_i = 0 \end{aligned}$$

$$\xrightarrow{\text{Kernelization}} \boxed{\begin{aligned} \underset{\alpha,\beta}{\text{maximize}} \quad & -0.5\sum_{i,j=1}^{n}\alpha_i\alpha_j t_i t_j k(\mathbf{x}_i,\mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \le \alpha_i \le C/n \quad \forall i \\ & \sum_{i=1}^{n}\alpha_i t_i = 0 \end{aligned}}$$

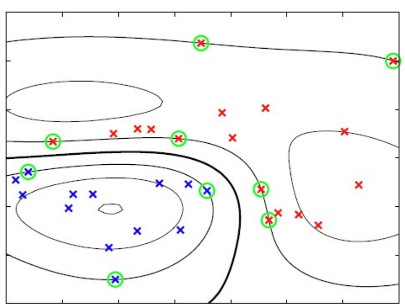
- The solution and final classifier is given by

$$\begin{aligned} \mathbf{w}^* &= \sum_{i=1}^{n}\alpha_i^* t_i \mathbf{x}_i \\ b^* &= t_j - \mathbf{w}^{*T}\mathbf{x}_j \\ &= t_j - \sum_{i=1}^{n}\alpha_i^* t_i \mathbf{x}_i^T\mathbf{x}_j \\ y &= \text{sign}\left(\mathbf{w}^{*T}\mathbf{x} + b^*\right) \\ &= \text{sign}\left(\sum_{i=1}^{n}\alpha_i^* t_i \mathbf{x}_i^T\mathbf{x} + b^*\right) \end{aligned} \xrightarrow{\text{Kernelization}} \boxed{\begin{aligned} \mathbf{w}^* &= \sum_{i=1}^{n}\alpha_i^* t_i \mathbf{x}_i \\ b^* &= t_j - \mathbf{w}^{*T}\mathbf{x}_j \\ &= t_j - \sum_{i=1}^{n}\alpha_i^* t_i k(\mathbf{x}_i,\mathbf{x}_j) \\ y &= \text{sign}\left(\mathbf{w}^{*T}\mathbf{x} + b^*\right) \\ &= \text{sign}\left(\sum_{i=1}^{n}\alpha_i^* t_i k(\mathbf{x}_i,\mathbf{x}) + b^*\right) \end{aligned}}$$

  of which index j satisfies $0 < \alpha_j^* < C/n$

SVM: Kernels

- Choice of kernels
  - Gaussian or polynomial kernels are used quite often
- Choice of Kernel Parameters
  - Ex: Gaussian Kernel: $k(\mathbf{x},\mathbf{z}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}\right)$. As a heuristic, the Bandwidth (σ) can be chosen to be the distance between neighboring points whose labels will likely affect the prediction of the query point.
- Example of SVM using Gaussian Kernel



- Bold line is the separating hyperplane
- Different contours indicate different values of $\mathbf{w}^{*T}\mathbf{x} + b^*$

Remark

- How to solve for the SVM dual?
  - "Chunking Algorithm"
    - Start with a random subset of the data and keep iteratively adding examples which violate the optimality conditions.
    - Problem: QP problem scales with the number of SVs.
    - Most SVM problems were solved with such algorithms in expensive QP solver softwares prior to SMO (see below).
  - Sequential Minimal Optimization
    - Divide the Dual problem into smaller sub-problems each of which consists of 2 of the linear equality constraint Lagrange multipliers (α's).
    - Find a Lagrange multiplier α_1 that violates the KKT conditions.
    - Pick a second multiplier α_2 and optimize the pair (α_1, α_2) using **coordinate ascent**.
    - Repeat the previous 2 steps until convergence (the KKT conditions are satisfied within a user-defined tolerance).
  - See Platt (1998) for details.