CSCI567 Machine Learning (Spring 2018)

Michael Shindler

Lecture on February 26, 2018

Outline

1. Administration
2. Linear Programming
3. Review of last lecture
4. SVM Examples

Outline

Administrative stuff

- Quiz 1 Grading in process
- Homework 1 Grading is coming along
- Please remember we have a large class.

Outline

Acknowledgement

- Much of this section comes from:
  Algorithm Design and Applications
  by Michael Goodrich and Roberto Tamassia
  Chapter 26: Linear Programming

Example Optimization Problem

Web server company wants to buy new servers.

Standard Model

- \$400
- 300W power
- Two shelves of rack
- Handles 1000 hits/min

Cutting-edge model

- \$1600
- 500W power
- One shelf
- 2000 hits/min

Budget:

- \$36,800
- 44 shelves of space
- 12,200W power

Goal: maximize the number of hits we serve per minute

The approach: linear programming

- Introduce variables x_1 and x_2
  (the number of servers of each model we buy)
- The number of hits per minute we get is:

$$1000x_1 + 2000x_2$$

- The budget places three limitations on us:

The approach: linear programming

- Introduce variables x_1 and x_2
  (the number of servers of each model we buy)
- The number of hits per minute we get is:

$$1000x_1 + 2000x_2$$

- The budget places three limitations on us:
  - The financial budget:

$$400x_1 + 1600x_2 \leq 36800$$

  - The number of shelves available:

$$2x_1 + x_2 \leq 44$$

  - Power used collectively

$$300x_1 + 500x_2 \leq 12200$$

Summarize the optimization problem

$$
\begin{aligned}
\text{maximize:} \quad & z = 1000x_1 + 2000x_2 \\
\text{subject to:} \quad & 400x_1 + 1600x_2 \leq 36800 \\
& 2x_1 + x_2 \leq 44 \\
& 300x_1 + 500x_2 \leq 12200 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Various algorithms exist to solve the problem

Maximum Flow as a Linear Program

- Given a flow network with source, sink, edge capacities
- Flow through an edge must be at most capacity of edge.
- Flow into a vertex must equal flow out
  (Exceptions: source, sink)

Maximum Flow as a Linear Program

- Given a flow network with source, sink, edge capacities
- Flow through an edge must be at most capacity of edge.
- Flow into a vertex must equal flow out
  (Exceptions: source, sink)

maximize:	$\sum_{e \in E^+(s)} f_e$	where s is the source.
subject to:	$0 \leq f_e \leq c_e$	for all edges e
	$\sum_{e \in E^-(v)} f_e = \sum_{e \in E^+(v)} f_e$	for all vertices v except the source and sink.

Standard form

A linear program is in *standard* form if it is an optimization problem in the following form:

$$
\begin{aligned}
\text{maximize:} \quad & z = \sum_{i \in V} c_i x_i \\
\text{subject to:} \quad & \sum_{j \in V} a_{ij} x_j \leq b_i \text{ for } i \in C \\
& x_i \geq 0 \text{ for } i \in V
\end{aligned}
$$

Converting to standard form

The form …	could also be written as …
minimize $f(x_1, \ldots, x_n)$	maximize $-f(x_1, \ldots, x_n)$
$f(x_1, \ldots, x_n) \geq y$	$-f(x_1, \ldots, x_n) \leq -y$
$f(x_1, \ldots, x_n) = y$	$f(x_1, \ldots, x_n) \leq y$ $f(x_1, \ldots, x_n) \geq y$

Matrix Notation

A linear function can be expressed as a dot product:

$$\sum_{i=1}^{n} a_x x_i = \boldsymbol{a} \cdot \boldsymbol{x}$$

We can write the standard form more compactly:

$$\begin{array}{ll} \text{maximize:} & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{subject to:} & \boldsymbol{a_1} \cdot \boldsymbol{x} \leq b_1 \\ & \boldsymbol{a_2} \cdot \boldsymbol{x} \leq b_2 \\ & \vdots \\ & \boldsymbol{a_m} \cdot \boldsymbol{x} \leq b_m \end{array}$$

Matrix Notation

A linear function can be expressed as a dot product:

$$\sum_{i=1}^{n} a_x x_i = \boldsymbol{a} \cdot \boldsymbol{x}$$

We can write the standard form more compactly:

$$\begin{array}{ll} \text{maximize:} & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{subject to:} & \boldsymbol{a_1} \cdot \boldsymbol{x} \le b_1 \\ & \boldsymbol{a_2} \cdot \boldsymbol{x} \le b_2 \\ & \vdots \\ & \boldsymbol{a_m} \cdot \boldsymbol{x} \le b_m \end{array}$$

Or even more compactly:

$$\begin{array}{ll} \text{maximize:} & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{subject to:} & A\boldsymbol{x} \le \boldsymbol{b} \end{array}$$

Slack Form

- Rewrite each inequality as an equivalent equality
- This introduces new *slack variables*
  - These are all nonnegative
  - These measure difference in original inequality
- We say it is in slack form if:

maximize:	$z = c_* + \sum_{j \in F} c_j x_j$
subject to:	$x_i = b_i - \sum_{j \in F} a_{ij} x_j$, for $i \in B$
	$x_i \geq 0$ for $1 \leq i \leq m + n$

- Sets B and F partition x_i into basic and free.

Duality

Given a linear program in standard form, a **dual LP**:

- is a minimization problem
- interchanges the roles of $\boldsymbol{b}$ and $\boldsymbol{c}$
- interchanges the roles of B and F.

The original is the **primal**.

Primal:

maximize:	$z = \boldsymbol{c} \cdot \boldsymbol{x}$
subject to:	$A\boldsymbol{x} \leq \boldsymbol{b}$
	$\boldsymbol{x} \geq \boldsymbol{0}$

Dual:

minimize:	$z = \boldsymbol{b} \cdot \boldsymbol{y}$
subject to:	$A^T\boldsymbol{y} \geq \boldsymbol{c}$
	$\boldsymbol{y} \geq \boldsymbol{0}$

Duality: The Web Server Problem

$$\begin{aligned} \text{maximize:} \quad & z = 1000x_1 + 2000x_2 \\ \text{subject to:} \quad & 400x_1 + 1600x_2 \leq 36800 \\ & 2x_1 + x_2 \leq 44 \\ & 300x_1 + 500x_2 \leq 12200 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Write the equivalent dual problem.

Duality: The Web Server Problem

$$
\begin{aligned}
\text{maximize:} \quad & z = 1000x_1 + 2000x_2 \\
\text{subject to:} \quad & 400x_1 + 1600x_2 \leq 36800 \\
& 2x_1 + x_2 \leq 44 \\
& 300x_1 + 500x_2 \leq 12200 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Write the equivalent dual problem.

$$
\begin{aligned}
\text{minimize:} \quad & z = 36800y_1 + 44y_2 + 12200y_3 \\
\text{subject to:} \quad & 400y_1 + 2y_2 + 300y_3 \geq 1000 \\
& 1600y_1 + y_2 + 500y_3 \geq 2000 \\
& y_1, y_2, y_3 \geq 0
\end{aligned}
$$

Maximum Flow as a Linear Program

- Given a flow network with source, sink, edge capacities
- Flow through an edge must be at most capacity of edge.
- Flow into a vertex must equal flow out (Exceptions: source, sink)

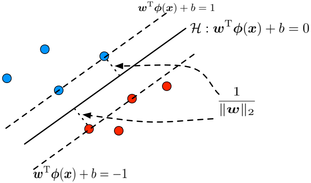
maximize:	$\sum_{e \in E^+(s)} f_e$	where s is the source.
subject to:	$0 \leq f_e \leq c_e$	for all edges e
	$\sum_{e \in E^-(v)} f_e = \sum_{e \in E^+(v)} f_e$	for all vertices v except the source and sink.

What is the dual?

Outline

Support Vector Machines

Interpretation: maximize the margin



- For separable data

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad \forall \;\; n$$

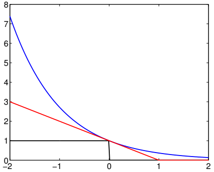
- For non-separable data

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad \forall \;\; n$$
$$\xi_n \geq 0, \quad \forall \; n$$

where C is our tradeoff (hyper)parameter.

Support Vector Machines

Interpretation: minimize loss



$$\ell^{\text{HINGE}}(f(\boldsymbol{x}), y) = \max(0, 1 - yf(\boldsymbol{x}))$$

- Minimize loss on all data

$$\min_{\boldsymbol{w},b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

- equivalently

$$\begin{aligned} \min_{\boldsymbol{w},b,\{\xi_n\}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \le \xi_n, \quad \forall\ n \\ & \xi_n \ge 0, \quad \forall\ n \end{aligned}$$

where all ξ_n are called *slack variables*.

Primal and dual

Primal

$$\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\text{s.t.} \quad 1 - y_n[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad \forall\, n$$

$$\xi_n \geq 0, \quad \forall\, n$$

Dual

$$\max_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\, n$$

$$\sum_n \alpha_n y_n = 0$$

Why we seek dual formulation

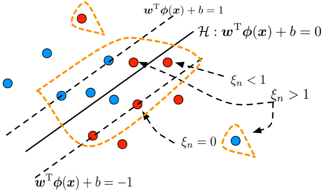
- We can kernelize the method by using kernel function in place of inner products
- We can discover interesting structures in solution: *support vectors*

Geometric interpretation of support vectors

Some α_n will become zero

$$\max_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$
$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n$$
$$\sum_n \alpha_n y_n = 0$$

Nonzero α_n is called support vector

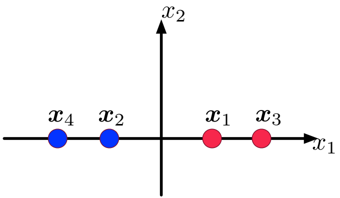


Support vectors are those being circled with the orange line. Removing them will change the solution.

Outline

The following toy problem

idx	x_1	x_2	y
$\boldsymbol{x}_1$	1	0	1
$\boldsymbol{x}_2$	-1	0	-1
$\boldsymbol{x}_3$	2	0	1
$\boldsymbol{x}_4$	-2	0	-1



Let us use linear kernel to solve the problem

$$k(\boldsymbol{x}_m, \boldsymbol{x}_n) = \boldsymbol{x}_m^{\mathrm{T}} \boldsymbol{x}_n$$

in other words, $\phi(\boldsymbol{x}) = \boldsymbol{x}$.

Guess the solution

- Decision boundary by SVM

$$x_1 = 0$$

  ie, the vertical axis
- Support vectors: $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

What is the dual formulation?

idx	x_1	x_2	y
$\boldsymbol{x}_1$	1	0	1
$\boldsymbol{x}_2$	-1	0	-1
$\boldsymbol{x}_3$	2	0	1
$\boldsymbol{x}_4$	-2	0	-1

Kernel matrix $\boldsymbol{x}_m^{\mathrm{T}}\boldsymbol{x}_n$

$$\boldsymbol{K} = \begin{pmatrix} 1 & -1 & 2 & -2 \\ -1 & 1 & -2 & 2 \\ 2 & -2 & 4 & -4 \\ -2 & 2 & -4 & 4 \end{pmatrix}$$

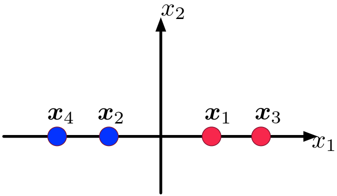
Dual formulation, by setting $C = +\infty$

$$\max_{\boldsymbol{\alpha}} \quad \sum_{n=1}^{4} \alpha_n - \frac{1}{2} \sum_{m=1,n=1}^{4} y_m y_n \alpha_m \alpha_n K_{mn}$$

$$\begin{aligned} \text{s.t.} \quad & 0 \leq \alpha_1 \leq +\infty \\ & 0 \leq \alpha_2 \leq +\infty \\ & 0 \leq \alpha_3 \leq +\infty \\ & 0 \leq \alpha_4 \leq +\infty \\ & \alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3 + \alpha_4 y_4 = 0 \end{aligned}$$

Simplify a bit

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{m=1,n=1}^{4} y_m y_n \alpha_m \alpha_n K_{mn} - \sum_{n=1}^{4} \alpha_n \\
\text{s.t.} \quad & 0 \leq \alpha_1 \\
& 0 \leq \alpha_2 \\
& 0 \leq \alpha_3 \\
& 0 \leq \alpha_4 \\
& \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0
\end{aligned}
$$



Intuition (due to symmetry

$$\alpha_1 = \alpha_2 \text{ and } \alpha_3 = \alpha_4$$

Note that the linear equality in the constraint is automatically satisfied now.

Putting the value of the kernel matrix in

$$\begin{aligned} \min_{\alpha_1, \alpha_3} \quad & 2(\alpha_1^2 + 4\alpha_1\alpha_3 + 4\alpha_3^2 - \alpha_1 - \alpha_3) \\ \text{s.t.} \quad & 0 \leq \alpha_1 \\ & 0 \leq \alpha_3 \end{aligned}$$

The objective function is (after removing the prefactor of 2)

$$\left(\alpha_1 + 2\alpha_3 - \frac{1}{2}\right)^2 - \frac{1}{4} + \alpha_3 \geq \alpha_3 - \frac{1}{4}$$

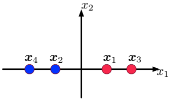
How to solve α_1 and α_3?

Since α_3 is always nonnegative, thus, to minimize the objective function, we have to set

$$\alpha_3 = 0$$

and set

$$\alpha_1 = \frac{1}{2}$$



We have shown now

$$\alpha_1 = \alpha_2 = 1/2, \qquad \alpha_3 = \alpha_4 = 0$$

- Namely, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are support vectors
- $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ are removable without changing solution - obviously from the graph!
- $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ contribute equally – intuitively true too!

$$\boldsymbol{w} = \sum_n \alpha_n y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = (1\ 0)^T$$

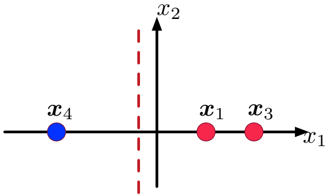
Thus, the decision boundary $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 0$ is

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = x_1 = 0$$

(I will leave out as an exercise to show $b = 0$).

Importance of support vectors

If we remove them, say x_2



and obviously the optimal decision boundary changes (to the dashed line)

Demo of SVM

- Binary classification problem
- Nonlinear kernel

$$k(\boldsymbol{x}_m, \boldsymbol{x}_n) = e^{-\|\boldsymbol{x}_m - \boldsymbol{x}_n\|_2^2 / 2\sigma^2}$$