Support Vector Machines - extra materials (S.S.)

Dr. Huiping Cao

Extra materials (just for fun)

- The following several slides explain the Lagrange formulation.
- I include them in case any of you is interested in understanding the details of solving this optimization problem.

Lagrange formulation

- Lagrangian for the optimization problem (take into account the constraints by rewriting the objective function).
  - $\mathcal{L}_P(\mathbf{w}, b, \lambda) = \frac{\|\mathbf{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i (y^{(i)}(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b) - 1)$

minimize w.r.t. **w** and b and maximize w.r.t. each $\lambda_i \geq 0$ where λ_is are Lagrange multiplier

- To minimize the Lagrangian, take the derivative of $\mathcal{L}_P(\mathbf{w}, b, \lambda)$ w.r.t. **w** and b and set them to 0:

$$\frac{\partial \mathcal{L}_P}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i y^{(i)} \mathbf{x}^{(i)}$$

$$\frac{\partial \mathcal{L}_P}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N} \lambda_i y^{(i)} = 0$$

Substituting: $\mathcal{L}_P(\mathbf{w}, b, \lambda)$ to $\mathcal{L}_D(\lambda)$

- Solving $\mathcal{L}_P(\mathbf{w}, b, \lambda)$ is still difficult because it solves a large number of parameters **w**, b, and λ_i.
- **Idea**: Transform Lagrangian into a function of the Lagrange multipliers only by substituting **w** and b in $\mathcal{L}_P(\mathbf{w}, b, \lambda)$, we get (the dual problem)
  - $\mathcal{L}_D(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\intercal \mathbf{x}^{(j)}$
  - See next slide to see the details of getting $\mathcal{L}_D(\lambda)$.
- It is very nice to have $\mathcal{L}_D(\lambda)$ because it is a simple quadratic form in the vector λ_i.

Substituting details, get the dual problem $\mathcal{L}_D(\lambda)$

$$\frac{\| \mathbf{w} \|^2}{2} = \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} = \frac{1}{2}\left(\sum_{i=1}^{N} \lambda_i y^{(i)} \left(\mathbf{x}^{(i)}\right)^\intercal\right) \cdot \left(\sum_{j=1}^{N} \lambda_j y^{(j)} \mathbf{x}^{(j)}\right) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\intercal \mathbf{x}^{(j)}$$

$$-\sum_{i}^{N} \lambda_i \left(y^{(i)}\left(\mathbf{w}^\intercal\mathbf{x}^{(i)} + b\right) - 1\right) = -\sum_{i=1}^{N} \lambda_i y^{(i)} \mathbf{w}^\intercal \mathbf{x}^{(i)} - \sum_{i=1}^{N} \lambda_i y^{(i)} b + \sum_{i=1}^{N} \lambda_i$$

$$= -\sum_{i=1}^{N} \lambda_i y^{(i)} \left(\sum_{j=1}^{N} \lambda_j y^{(j)} (\mathbf{x}^{(j)})^\intercal\right) \mathbf{x}^{(i)} + \sum_{i=1}^{N} \lambda_i$$

$$= \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\intercal \mathbf{x}^{(j)}$$

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\intercal \mathbf{x}^{(j)}$$

Finalized $\mathcal{L}_D(\lambda)$

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^{\intercal} \mathbf{x}^{(j)}$$

Maximize w.r.t. λ

Subject to

$\lambda_i \geq 0$ for $i = 1,2,\ldots,N$

and $\sum_{i=1}^{N} \lambda_i y^{(i)} = 0$

Solve $\mathcal{L}_D(\lambda)$ using quadratic programming (QP). We get all the λ_i.

Solve $\mathcal{L}_D(\lambda)$ – QP

- We are maximizing $\mathcal{L}_D(\lambda)$

$$max_\lambda\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y^{(i)}y^{(j)}\left(\mathbf{x}^{(i)}\right)^\intercal\mathbf{x}^{(j)}\right)$$

- Subject to constraints
  - $\lambda_i \geq 0$ for $i = 1,2,\ldots,N$
  - and $\sum_{i=1}^{N}\lambda_i y^{(i)} = 0$
- Translate the objective to minimization because QP packages generally come with minimization.

$$min_\lambda\left(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y^{(i)}y^{(j)}\left(\mathbf{x}^{(i)}\right)^\intercal\mathbf{x}^{(j)} - \sum_{i=1}^{N}\lambda_i\right)$$

Solve $\mathcal{L}_D(\lambda)$ – QP

$$min_\lambda\left(\frac{1}{2}\lambda^\intercal\begin{bmatrix} y_1y_1\mathbf{x}_1{}^\intercal\mathbf{x}_1 & y_1y_2\mathbf{x}_1{}^\intercal\mathbf{x}_2 & \dots & y_1y_N\mathbf{x}_1{}^\intercal\mathbf{x}_N \\ y_2y_1\mathbf{x}_2{}^\intercal\mathbf{x}_1 & y_2y_2\mathbf{x}_2{}^\intercal\mathbf{x}_2 & \dots & y_2y_N\mathbf{x}_2{}^\intercal\mathbf{x}_N \\ \dots & \dots & \dots & \dots \\ y_Ny_1\mathbf{x}_N{}^\intercal\mathbf{x}_1 & y_Ny_2\mathbf{x}_N{}^\intercal\mathbf{x}_2 & \dots & y_Ny_N\mathbf{x}_N{}^\intercal\mathbf{x}_N \end{bmatrix}\lambda + (-1)^\intercal\lambda\right)$$

Subject to

$$\mathbf{y}^\intercal\lambda = 0$$

$$0 \le \lambda \le \infty$$

Let Q represent the matrix with the quadratic coefficients $min_\lambda\left(\frac{1}{2}\lambda^\intercal Q\lambda + (-1)^\intercal\lambda\right)$ subject to $\mathbf{y}^\intercal\lambda = 0$ and $\lambda \ge 0$.

Lagrange multiplier

- QP solves $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_N$ where most of them are zeros.
- Karush-Kuhn-Tucker (KKT) conditions
  - $\lambda_i \geq 0$
  - The constraint (zero form with extreme value)
    - $\lambda_i\left(y^{(i)}\left(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b\right) - 1\right) = 0$
    - Either λ_i is zero
    - or $\left(y^{(i)}\left(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b\right) - 1\right) = 0$
- **Support vector $\mathbf{x}^{(i)}$**: $y^{(i)}\left(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b\right) - 1 = 0$ and $\lambda_i > 0$
- Training instances that do not reside along these hyperplanes have $\lambda_i = 0$.

Quadratic programming packages – Octave

- Solve the quadratic program

$$min_{\mathbf{x}}(0.5\mathbf{x}^\top * H * \mathbf{x} + \mathbf{x}^\top * q)$$

- Subject to

$$\begin{cases} A * \mathbf{x} = & b & \\ lb \leq & \mathbf{x} & \leq ub \\ A_{lb} \leq & A_{in} * \mathbf{x} & \leq A_{ub} \end{cases}$$

Quadratic programming packages - MATLAB

- Optimization toolbox in MATLAB

$$min_{\mathbf{x}}(\frac{1}{2}\mathbf{x}^{\top}H\mathbf{x} + \mathbf{f}^{\top}\mathbf{x})$$

- Such that

$$\begin{cases} A \cdot \mathbf{x} & \leq b, \\ Aeq \cdot \mathbf{x} & = beq, \\ lb & \leq \boldsymbol{x} \leq ub. \end{cases}$$

Get **w** and b

- **w** and b depend on support vectors $\mathbf{x}^{(i)}$ and its class label $y^{(i)}$.
- **w** value: $\mathbf{w} = \sum_{i=1}^{N} \lambda_i y^{(i)} \, \mathbf{x}^{(i)}$
- b value: $b = y^{(i)} - \mathbf{w}^\intercal \mathbf{x}^{(i)}$
- Idea:
  - Given a support vector ($\mathbf{x}^{(i)}$, $y^{(i)}$),we have $y^{(i)}\left(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b\right) - 1 = 0$
  - Multiply $y^{(i)}$ on both sides, we get $(\mathrm{y}^{(i)})^2\left(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b\right) - \mathrm{y}^{(i)} = 0$
  - $(\mathrm{y}^{(i)})^2 = 1$ because $\mathrm{y}^{(i)} = 1$ or $\mathrm{y}^{(i)} = -1$
  - Then, $\left(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b\right) - \mathrm{y}^{(i)} = 0$

Get w and b – Example

- Solve λ using quadratic programming packages
- $\mathbf{w}^{\top} = (w_1, w_2)$

$$w_1 = \sum_{i=1}^{2} \lambda_i y^{(i)} {x_1}^{(i)} = 100 * 1 * 0.4 + 100 * (-1) * 0.5 = -10$$

$$w_2 = \sum_{i=1}^{2} \lambda_i y^{(i)} {x_2}^{(i)} = 100 * 1 * 0.5 + 100 * (-1) * 0.6 = -10$$

$$b = 1 - \mathbf{w}^{\top}\mathbf{x}^{(1)} = 1 - \left((-10) * 0.4 + (-10) * (0.5)\right) = 10$$

$x_1^{(i)}$	$x_2^{(i)}$	$y^{(i)}$	λ_i
0.4	0.5	1	100
0.5	0.6	-1	100
0.9	0.4	-1	0
0.7	0.9	-1	0
0.17	0.05	1	0
0.4	0.35	1	0
0.9	0.8	-1	0
0.2	0	1	0

Prediction

- Given a test data point $\mathbf{z}$, we can calculate
  - $y_z = sign(\mathbf{w}^\top \mathbf{z} + b) = sign\left(\left(\sum_{i=1}^{N} \lambda_i y^{(i)} (\mathbf{x}^{(i)})^\top\right)\mathbf{z} + b\right)$
- If $y_z = 1$, the test instance is classified as positive class
- If $y_z = -1$, the test instance is classified as negative class

Kernel SVM

- In particular, in the quadratic programming (QP) task, the SVM model replaces the dot product $\left(\mathbf{x}^{(i)}\right)^{\intercal}\mathbf{x}^{(j)}$ with $\phi\left(\mathbf{x}^{(i)}\right)^{\intercal}\phi(\mathbf{x}^{(j)})$.
- Thus,

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^{\intercal}\, \mathbf{x}^{(j)}$$

$$\downarrow$$

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} (\mathbf{z}^{(i)})^{\intercal}\, \mathbf{z}^{(j)}$$