Compsci 571 HW3

Yilin Gao (yg95)

February 10, 2018

1 Separability

Proof:

To show that two convex hulls do not intersect, we could show any point in one convex hull is not in the other convex hull, or equivalently, any point in one convex hull is not the same as any point in the other convext hull.

Suppose $\mathbf{x}_i = \sum_n \alpha_{ni}\mathbf{x}_n$ is a point in the convex hull of $\{\mathbf{x}_n\}$, with $\alpha_{ni} \geq 0$ for $\forall i$ and $\sum_n \alpha_{ni} = 1$, and $\mathbf{x}_j = \sum_m \alpha_{mj}\mathbf{x'}_m$ is a point in the convex hull of $\{\mathbf{x'}_m\}$, with $\alpha_{mj} \geq 0$ for $\forall j$ and $\sum_m \alpha_{mj} = 1$. We need to prove $\mathbf{x}_i \neq \mathbf{x'}_j$ for $\forall i$ and $\forall j$.

We prove with contradiction. Assume for a given i, there exists some j such that $\mathbf{x}_i = \mathbf{x'}_j$. Then $\mathbf{w}^T\mathbf{x}_i = \mathbf{w}^T\mathbf{x'}_j$, and $\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x'}_j = 0$.

$$\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x'}_j = \mathbf{w}^T(\sum_n \alpha_{ni}\mathbf{x}_n) - \mathbf{w}^T(\sum_m \alpha_{mj}\mathbf{x'}_m) = \sum_n \alpha_{ni}(\mathbf{w}^T\mathbf{x}_n) - \sum_m \alpha_{mj}(\mathbf{w}^T\mathbf{x'}_m)$$

$$= [\sum_n \alpha_{ni}(\mathbf{w}^T\mathbf{x}_n) + w_0] - [\sum_m \alpha_{mj}(\mathbf{w}^T\mathbf{x'}_m) + w_0]$$

Because $\sum_n \alpha_{ni} = 1$ and $\sum_m \alpha_{mj} = 1$:

$$= [\sum_n \alpha_{ni}(\mathbf{w}^T\mathbf{x}_n) + \sum_n \alpha_{ni}w_0] - [\sum_m \alpha_{mj}(\mathbf{w}^T\mathbf{x'}_m) + \sum_m \alpha_{mj}w_0]$$

$$= \sum_n \alpha_{ni}(\mathbf{w}^T\mathbf{x}_n + w_0) - \sum_m \alpha_{mj}(\mathbf{w}^T\mathbf{x'}_m + w_0)$$

If the two sets of points $\{\mathbf{x}_i\}$ and $\{\mathbf{x'}_j\}$are linearly seprable, $\mathbf{w}^T\mathbf{x}_n + w_0 > 0$ for $\forall n$, and $\mathbf{w}^T\mathbf{x'}_m + w_0 < 0$ for $\forall m$. And because $\alpha_{ni} \geq 0$ for $\forall ni$ and $\alpha_{mj} \geq 0$ for $\forall mj$, the above equation is a sum of n positive numbers minus a sum of m negative numbers, which is a positive number.

So $\mathbf{w}^T\mathbf{x}_i$ cannot be equal to $\mathbf{w}^T\mathbf{x'}_j$. Contradiction.

So for a given i, there is no j such that $\mathbf{x}_i = \mathbf{x'}_j$. So the two convex hulls do not intersect. □

2 Logistic Regression and Gradient Descent

(a) *Proof:*

$$\sigma'(a) = -\frac{1}{(1+e^{-a})^2}(-e^{-a}) = \frac{e^{-a}}{(1+e^{-a})^2}$$

$$\sigma(a)(1-\sigma(a)) = \frac{1}{1+e^{-a}}(1-\frac{1}{1+e^{-a}}) = \frac{1}{1+e^{-a}}\frac{e^{-a}}{1+e^{-a}} = \frac{e^{-a}}{(1+e^{-a})^2}$$

$$\therefore \sigma'(a) = \sigma(a)(1-\sigma(a)) \quad \square$$

(b)

$$\frac{\partial L_{\mathbf{w}}}{\partial w_j} = \frac{\partial}{\partial w_j}\sum_{i=1}^{n}\{-y^{(i)}\log[\sigma(\mathbf{w}^T\mathbf{x}^{(i)})] - (1-y^{(i)})\log[1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})]\}$$

$$= \sum_{i=1}^{n}\{-y^{(i)}\frac{\sigma'(\mathbf{w}^T\mathbf{x}^{(i)})}{\sigma(\mathbf{w}^T\mathbf{x}^{(i)})}x_j^{(i)} - (1-y^{(i)})\frac{-\sigma'(\mathbf{w}^T\mathbf{x}^{(i)})}{1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})}x_j^{(i)}\}$$

Because $\sigma'(\mathbf{w}^T\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})(1-\sigma(\mathbf{w}^T\mathbf{x}))$,

$$= \sum_{i=1}^{n}\{-y^{(i)}(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)}))x_j^{(i)} + (1-y^{(i)})\sigma(\mathbf{w}^T\mathbf{x}^{(i)})x_j^{(i)}\}$$

$$= \sum_{i=1}^{n}\{x_j^{(i)}(\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\}$$

(c) *Proof:*

To prove $L_{\mathbf{w}}$ is convex in $\mathbf{w}$, we need to prove that for any $\mathbf{w}_1$ and $\mathbf{w}_2$ and $\theta \in [0,1]$, $L(\theta\mathbf{w}_1+(1-\theta)\mathbf{w}_2) \leq \theta L(\mathbf{w}_1) + (1-\theta)L(\mathbf{w}_2)$.

Use $\mathbf{w}^*$ to represent $\theta\mathbf{w}_1 + (1-\theta)\mathbf{w}_2$.

For the left-hand side:

$$LHS = L(\mathbf{w}^*) = \sum_{i=1}^{n}\{-y^{(i)}\log[\sigma(\mathbf{w}^{*T}\mathbf{x})] - (1-y^{(i)})\log[1-\sigma(\mathbf{w}^{*T}\mathbf{x})]\}$$

Because $\sigma(\mathbf{w}^{*T}\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^{*T}\mathbf{x}}} = \frac{1}{1+e^{-\theta\mathbf{w}_1^T\mathbf{x}} * e^{-(1-\theta)\mathbf{w}_2^T\mathbf{x}}}$, define $A \triangleq 1 + e^{-\theta\mathbf{w}_1^T\mathbf{x}} * e^{-(1-\theta)\mathbf{w}_2^T\mathbf{x}}$,

$$= \sum_{i=1}^{n}\{y^{(i)}\log A - (1-y^{(i)})(\log{(1-A)} - \log A)\}$$

$$= \sum_{i=1}^{n}\{y^{(i)}\log A + (1-y^{(i)})(\mathbf{w}^{*T}\mathbf{x} + \log A)\}$$

$$= \sum_{i=1}^{n}\{(1-y^{(i)})\mathbf{w}^{*T}\mathbf{x} + \log A)\}$$

For the right hand side:

$$RHS = \theta L(\mathbf{w}_1) + (1-\theta)L(\mathbf{w}_2)$$

$$= \theta\sum_{i=1}^{n}\{-y^{(i)}\log[\sigma(\mathbf{w}_1^T\mathbf{x})] - (1-y^{(i)})\log[1-\sigma(\mathbf{w}_1^T\mathbf{x})]\} + (1-\theta)\sum_{i=1}^{n}\{-y^{(i)}\log[\sigma(\mathbf{w}_2^T\mathbf{x})] - (1-y^{(i)})\log[1-\sigma(\mathbf{w}_2^T\mathbf{x})]\}$$

$$= \sum_{i=1}^{n}\{-y^{(i)}[\theta\log[\sigma(\mathbf{w}_1^T\mathbf{x})] + (1-\theta)\log[\sigma(\mathbf{w}_2^T\mathbf{x})]] - (1-y^{(i)})[\theta\log[1-\sigma(\mathbf{w}_1^T\mathbf{x})] + (1-\theta)\log[1-\sigma(\mathbf{w}_2^T\mathbf{x})]]\}$$

Define $B \triangleq (1+e^{-\mathbf{w}_1^T\mathbf{x}})^{\theta}(1+e^{-\mathbf{w}_2^T\mathbf{x}})^{1-\theta}$,

$$= \sum_{i=1}^{n} \{y^{(i)} \log B + (1 - y^{(i)})(\mathbf{w}^{*T}\mathbf{x} + \log B)\}$$

$$= \sum_{i=1}^{n} \{(1 - y^{(i)})\mathbf{w}^{*T}\mathbf{x} + \log B)\}$$

So $RHS - LHS = n(\log B - \log A)$,

$$\log B - \log A = \log \frac{(1 + e^{-\mathbf{w}_1^T\mathbf{x}})^\theta (1 + e^{-\mathbf{w}_2^T\mathbf{x}})^{1-\theta}}{1 + e^{-\theta\mathbf{w}_1^T\mathbf{x}} * e^{-(1-\theta)\mathbf{w}_2^T\mathbf{x}}}$$

Because for $\theta \in [0,1]$, $(1+x_1)^\theta(1+x_2)^{1-\theta} \geq 1+x_1^\theta x_2^{1-\theta}$, so $(1+e^{-\mathbf{w}_1^T\mathbf{x}})^\theta(1+e^{-\mathbf{w}_2^T\mathbf{x}})^{1-\theta} \geq 1+e^{-\theta\mathbf{w}_1^T\mathbf{x}} * e^{-(1-\theta)\mathbf{w}_2^T\mathbf{x}}$, $RHS - LHS = n(\log B - \log A) \geq 0$.

So for any $\mathbf{w}_1$ and $\mathbf{w}_2$ and $\theta \in [0,1]$, $L(\theta\mathbf{w}_1 + (1-\theta)\mathbf{w}_2) \leq \theta L(\mathbf{w}_1) + (1-\theta)L(\mathbf{w}_2)$. $L_\mathbf{w}$ is convex in $\mathbf{w}$. □

(d) See the Jupyter notebook.

3 Boosting

(a) From the class notes on Boosting, we have proved that if the weak learning assumption holds, at tth step, AdaBoost's training misclassification error is ≥ 0 and $\leq e^{-2\gamma_{WLA}^2 T}$. According to the weak learning assumption, $\gamma_{WLA} > 0$, so $-2\gamma_{WLA}^2 < 0$. So when $T \to \infty$, $-2\gamma_{WLA}^2 T \to -\infty$, and $e^{-2\gamma_{WLA}^2 T} \to 0$. When the lower bound is 0, and the upper bound approaches 0, AdaBoost's training error also approaches 0. □

(b) Use the boosting statistical view to derive the algorithm for weighted AdaBoost.

Define the AdaBoost output model as $f(\mathbf{x}) = \sum_{j=1}^{p} \lambda_{T,j} h_j(\mathbf{x})$, which is a linear combination of p weak classifiers. The training misclassification error is $R^{train}(\lambda) = \sum_{i=1}^{n} w_i e^{-(\mathbf{M}\lambda)_i}$. Assign $\lambda_1 = \mathbf{0}$ at the beginning, which means every training point is misclassified at the beginning.

Do the following in each tth ($t \in [1, T]$) step:

First, use coordinate descent to find the direction j_t with largest reduction in $R^{train}(\lambda)$:

$$j_t \in \underset{j}{\text{argmax}}[-\frac{\partial R^{train}(\lambda_t + \alpha\mathbf{e}_j)}{\partial\alpha}|_{\alpha=0}]$$

$$= \underset{j}{\text{argmax}}[-\frac{\partial}{\partial\alpha}[\sum_{i=1}^{n} w_i e^{-(\mathbf{M}(\lambda_t + \alpha\mathbf{e}_j))_i}]|_{\alpha=0}]$$

$$= \underset{j}{\text{argmax}}[\sum_{i=1}^{n} M_{ij} w_i e^{-(\mathbf{M}\lambda_t)_i}]$$

Define a discrete probability distribution as $d_{t,i}^* = w_i e^{-(\mathbf{M}\lambda_t)_i}/Z_t$, where $Z_t = \sum_{i=1}^{n} w_i e^{-(\mathbf{M}\lambda_t)_i}$ so that $\sum_{i=1}^{n} d_{t,i}^* = 1$.

So

$$j_t \in \underset{j}{\text{argmax}}[\sum_{i \in M_{ij}=1} d_{t,i}^* - \sum_{i \in M_{ij}=-1} d_{t,i}^*]$$

$$= \underset{j}{\text{argmax}}[1 - \sum_{i \in M_{ij}=-1} d_{t,i}^* - \sum_{i \in M_{ij}=-1} d_{t,i}^*]$$

$$= \underset{j}{\operatorname{argmin}}[\sum_{i \in M_{ij}=-1} d^*_{t,i}]$$

Define $\sum_{i \in M_{ij}=-1} d^*_{t,i}$ as d^*_-, and $\sum_{i \in M_{ij}=1} d^*_{t,i}$ as d^*_+. We have $d^*_+ + d^*_- = 1$.
After get j_t, $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \alpha \mathbf{e}_{jt}$.
Second, take an α_t long step at the coordinate j_t to reach the minimal value of $R^{train}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})$:

$$0 = \frac{\partial R^{train}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})}{\partial \alpha}|_{\alpha_t}$$

$$= \sum_{i=1}^{n} M_{ij} w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_i - \alpha_t M_{ij_t}}$$

$$= e^{-\alpha_t} \sum_{i \in M_{ij}=1} w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_i} - e^{\alpha_t} \sum_{i \in M_{ij}=-1} w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_i}$$

$$= e^{-\alpha_t} \sum_{i \in M_{ij}=1} w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_i}/Z_t - e^{\alpha_t} \sum_{i \in M_{ij}=-1} w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_i}/Z_t$$

$$= e^{-\alpha_t} \sum_{i \in M_{ij}=1} d^*_{t,i} - e^{\alpha_t} \sum_{i \in M_{ij}=-1} d^*$$

$$= e^{-\alpha_t} d^*_+ - e^{\alpha_t} d^*_-$$

This leads to

$$\alpha_t = \frac{1}{2} \ln(\frac{1 - d^*_-}{d^*_-})$$

And we could know in the next round $d^*_{t+1,i} = w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_{t+1})_i}/Z_{t+1}$, where $Z_{t+1} = \sum_{i=1}^{n} w_i e^{-(\mathbf{M}\boldsymbol{\lambda}_{t+1})_i}$.
So $d^*_{t+1,i} = w_i e^{-(\mathbf{M}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{jt})_i)}/Z_{t+1} = w_i e^{(-\mathbf{M}\boldsymbol{\lambda}_t)_i} e^{-\alpha_t M_{ij_t}}/Z_{t+1} = d^*_{t,i} e^{-\alpha_t M_{ij_t}} \frac{Z_t}{Z_{t+1}} \propto d^*_{t,i} e^{-\alpha_t M_{ij_t}} = d^*_{t,i} e^{-\alpha_t y_i h_{j_t}(\mathbf{x}_i)}$. And we could always adjust the normalization factor to make $\sum_{i=1}^{n} d^*_{t+1,i} = 1$.

And for the final output $f(x)$, because $\lambda_{T,j} = \sum_{t=1}^{T} \alpha_t \mathbf{1}_{[j_t=j]}$, we could have $f(\mathbf{x}) = \sum_{j=1}^{p} \lambda_{T,j} h_j(\mathbf{x}) = \sum_{j=1}^{p} \sum_{t=1}^{T} \alpha_t \mathbf{1}_{[j_t=j]} h_j(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t \sum_{j=1}^{p} h_j(\mathbf{x}) \mathbf{1}_{[j_t=j]} = \sum_{t=1}^{T} \alpha_t h_{j_t}(\mathbf{x})$.
So the chosen j_t in each step can also be expressed as $\operatorname{argmin}_j[\sum_{i=1}^{n} d^*_{t,i} \mathbf{1}_{[y_i \neq f(\mathbf{x}_i)]}] = \operatorname{argmin}_j[\sum_{i=1}^{n} d^*_{t,i} \mathbf{1}_{[y_i \neq h_j(\mathbf{x}_i)]}]$.
So the weighted AdaBoost algorithm is:

```
Data: training data $\{(\mathbf{x}_i, y_i)_{i=1}^{n}\}$, number of iterations $T$, some weak classifiers $h_j(\mathbf{x})_{j=1}^{p}$
Initialize $d^*_{1,i} = w_i$ for $i = 1 \dots n$ (actually this is the only difference from the normal
  AdaBoost algorithm);
for $t = 1 \dots T$ do
    $j_t = argmin_j[\sum_{i=1}^{n} d^*_{t,i} \mathbf{1}_{[y_i \neq h_j(\mathbf{x}_i)]}]$ ;
    $d^*_- = \sum_{i=1}^{n} d^*_{t,i} \mathbf{1}_{[y_i \neq h_{j_t}(\mathbf{x}_i)]}$ ;
    $\alpha_t = \frac{1}{2} \ln(\frac{1-d^*_-}{d^*_-})$ ;
    $d^*_{t+1,i} = d^*_{t,i} e^{-\alpha_t y_i h_{j_t}(\mathbf{x}_i)}/Z_{t+1}$ such that $\sum_{i=1}^{n} d^*_{t+1,i} = 1$ ;
end
Result: $H(\mathbf{x}) = sign(\sum_{t=1}^{T} \alpha_t h_{j_t}(\mathbf{x}))$
```

(c) See the Jupyter notebook.