Universidade do Minho (DMA) Machine Learning:: Fundamental and Application

maio 2019 Teste Modelo

Nome ____

Número ____________________ Curso ____________________

Seja uma base de dados $D = (x^n, y^n)_{n=1}^N$ onde $x^n \in \mathbb{R}^2$ e $y^n \in \{-1, 1\}$. Pretendemos criar um classificador de tipo preceptron mas usando uma função de activação regularizada

$$\rho(s; \mu) = \frac{s}{\mu + |s|},$$

onde $\mu \geq 0$ é um parâmetro. O objetivo é estudar o novo predictor associado e construir dois algoritmos de aprendizagem. Recordamos a notação $\widetilde{x} = (1, x_1, x_2)^T$

Parte A.

1) Justificar que se $\rho(s; 0)$ é a função sinal $\text{sng}(s)$.

2) Mostrar que $|\rho(s; \mu) - \text{sng}(s)| = \frac{\mu}{\mu+|s|}$. Deduzir que para qualquer $s \neq 0$

$$\lim_{\mu \to 0^+} \rho(s; \mu) = \text{sng}(s).$$

3) Calcular $\rho'(s; \mu)$ a derivada em ordem a s.

4) Seja $\widetilde{w} = (w_0, w_1, w_2)^T$. Definimos o predictor como

$$\widehat{y}(x; \widetilde{w}) = \rho(s; \mu), \quad \text{com} \quad s = w_0 + w_1 x_1 + w_2 x_2 = (\widetilde{w})^T \widetilde{x}.$$

Calcular o gradiente do predictor $\nabla_{\widetilde{w}} \widehat{y}(x; \widetilde{w})$.

Parte B.

Introduzimos a função erro como

$$E(\widetilde{w}; D) = \sum_{n=1}^{N} \frac{1}{2} (\widehat{y}^n - y^n)^2, \quad \widehat{y}^n = \widehat{y}(x^n; \widetilde{w}).$$

1) Calcular o gradiente $\nabla_{\widetilde{w}} E(\widetilde{w}; D)$.

2) Consideramos a base de dados elementar $\{x^n, y^n\}$ com $\{(1, 0), -1\}$, $\{(0, 1), -1\}$, $\{(-1, 0), 1\}$, $\{(0, -1), 1\}$. Determinar o vetor $\widetilde{w}$ que minimiza o erro $E(\widetilde{w}; D)$ e calcular o erro em função de μ.

3) Mostrar, neste último caso, que $\lim_{\mu \to 0^+} E(\widetilde{w}; D) = 0$.

4) Propor um algoritmo de aprendizagem baseado no método do gradiente estocástico. Identificar a fórmula que permite calcular $\widetilde{w}(t + 1)$ em função de $\widetilde{w}(t)$ e do elemento escolhido (x^m, y^m).

5) Como o método do gradiente é dependente do parâmetro μ. O que se passa se $\mu \to 0^+$?

Parte C.

Nesta parte parte, pretendemos adaptar o algoritmo do perceptron no caso regularizado usando a função $\rho(s, \mu)$.

1) Recordar o algoritmo de aprendizagem do perceptron.

2) Propor uma adaptação usando a função de activação $\rho(s, \mu)$ em substituição da função sng.

3) Consideramos a nova função erro

$$E(\widetilde{w}; D) = \sum_{n=1}^{N} \max(-\widehat{y}^n y^n, 0).$$

Justificar que $E(\widetilde{w}; D) = 0$ corresponde a uma classificação correta. Neste caso, explicar porque o erro é independente de μ.

4) (Mais difícil). Justificar porque, no caso do algoritmo perceptron, o valor de μ não tem muito impacto no algoritmo de aprendizagem (ao contrário da Parte B).

Correção

Parte A.

1) Se $s < 0$, temos $-1 = \text{sng}(s) = \frac{s}{|s|} = \rho(s;0)$. Se $s > 0$, temos $1 = \text{sng}(s) = \frac{s}{|s|} = \rho(s;0)$. Logo $\text{sng}(s) = \rho(s;0)$.

2) Temos

$$\rho(s,\mu) - \text{sng}(s) = \frac{s}{\mu + |s|} - \frac{s}{|s|} = \frac{s|s| - (\mu + |s|)s}{|s|(\mu + |s|)} = \frac{-\mu s}{|s|(\mu + |s|)}.$$

Logo $|\rho(s;\mu) - \text{sng}(s)| = \dfrac{|s|\mu}{|s|(\mu + |s|)} = \dfrac{\mu}{\mu + |s|}$.

Em conclusão, seja $s \neq 0$, temos

$$\lim_{\mu \to 0^+} |\rho(s;\mu) - \text{sng}(s)| = \frac{\mu}{\mu + |s|} = \frac{0}{|s|} = 0.$$

A função de ativação $\rho(s;\mu)$ converge simplesmente para a função de ativação $\text{sng}(s)$.

3) Temos

$$\rho'(s;\mu) = \frac{\mu}{(\mu + |s|)^2} > 0$$

4)

$$\nabla_{\widetilde{w}} \widehat{y}(x;\widetilde{w}) = \rho'(s;\mu) \nabla_{\widetilde{w}} (\widetilde{w})^T \widetilde{x}, \quad s = (\widetilde{w})^T \widetilde{x},$$

seja

$$\nabla_{\widetilde{w}} \widehat{y}(x;\widetilde{w}) = \frac{\mu \widetilde{x}}{(\mu + |(\widetilde{w})^T \widetilde{x}|)^2}. \qquad (*)$$

Parte B.

1) O gradiente é

$$\nabla_{\widetilde{w}} E(\widetilde{w};D) = \sum_{n=1}^{N} (\widehat{y}^n - y^n) \nabla_{\widetilde{w}} \widehat{y}(x^n;\widetilde{w}) = \sum_{n=1}^{N} \frac{\mu(\widehat{y}^n - y^n)\widetilde{x}^n}{(\mu + |(\widetilde{w})^T \widetilde{x}^n|)^2}$$

O gradient estoástico corresponde a usar apenas um elemento m da base de dados seja $\dfrac{\mu(\widehat{y}^m - y^m)\widetilde{x}^m}{(\mu + |(\widetilde{w})^T \widetilde{x}^m|)^2}$.

2) Devida à simetrias, temos $\widetilde{w} = (0, \alpha, \alpha)$ com $\alpha \in \mathbb{R}$ a determinar. Logo o erro obtido é

$$2E(\widetilde{w};D) = \left(\frac{\alpha}{\mu + |\alpha|} + 1\right)^2 + \left(\frac{\alpha}{\mu + |\alpha|} + 1\right)^2 + \left(-\frac{\alpha}{\mu + |\alpha|} - 1\right)^2 + \left(-\frac{\alpha}{\mu + |\alpha|} - 1\right)^2$$

seja

$$E(\widetilde{w};D) = 2\left(\frac{\alpha + \mu + |\alpha|}{\mu + |\alpha|}\right)^2$$

Deduzimos que $\alpha < 0$ para ter o erro mais baixo e obtemos $E(\widetilde{w};D) = \left(\dfrac{2\mu}{\mu + |\alpha|}\right)^2$. Não podemos definir um valor de α mas podemos observar que $|\alpha| >> \mu$, obtemos um erro pequeno.

3) Qualquer que seja $\alpha \neq 0$, temos

$$\lim_{\mu \to 0^+} E(\widetilde{w};D) = \lim_{\mu \to 0^+} \left(\frac{2\mu}{\mu + |\alpha|}\right)^2 = 0.$$

4) O algoritmo do gradiente corresponde aos passos seguintes

```
1. while(not converge)
2.          choose an element m
3.          compute o gradient Gm with formula (*)
4.          w(t+1)=w(t)-eta Gm
5. do
```

A fórmula de correção do gradiente é

$$\widetilde{w}(t+1) = \widetilde{w}(t) - \eta \frac{\mu(\widehat{y}^m - y^m)\widetilde{x}^m}{(\mu + |(\widetilde{w})^T \widetilde{x}^m|)^2}.$$

4) Se $\mu \to 0^+$, o gradiente se torna cada vez mais pequenos até desaparecer por razão do parâmetro multiplicativo μ. Logo o algoritmo acabo de funcionar. Precisamos que o parâmetro μ seja a volta da unidade (ou então compensar com um $\eta \approx \frac{1}{\mu}$).

Parte C.

1) O algoritmo consiste em escolher um elemento (x^m, y^m) da base de dados. Melhoramos o perceptron modificando os coeficientes de $\widetilde{w}$ como

$$\widetilde{w}(t+1) = \widetilde{w}(t) - \eta(\widehat{y}^m - y^m)\widetilde{x}^m$$

onde η é a taxa de aprendizagem.

2) Um algoritmo alternativo, derivando do perceptron, logo seria substituir $\text{sng}(s)$ para $\rho(s, \mu)$, seja

$$\widetilde{w}(t+1) = \widetilde{w}(t) - \eta(\rho(s^m; \mu) - y^m)\widetilde{x}^m, \quad s^m = (\widetilde{w})^T \widetilde{x}^m$$

Nota que esta formula converge para a formulado perceptron quando $\mu \to 0^+$, enquanto não temos esta propriedade no algoritmo da parte B. Temos dois algoritmos bem diferentes.

3) Se $E(\widetilde{w}; D) = 0$, significa que $\rho(s^m; \mu)y^m \geq 0$, logo $(\widetilde{w})^T \widetilde{x}^m y^m \geq 0$ e esta última relação é independente de μ.

4) No algoritmo do perceptron, o que é importante são os sinais respectivos entre $\widehat{y}^m$ e y^m independemente do μ. Quando s^m e y^m são de sinais opostos, a quantidade $|\rho(s^m; \mu) - y^m| \in [1, 2]$ independemente de μ. Logo, o algoritmo se comporta de uma maneira muito similar ao caso do perceptron. O caso onde s^m e y^m são de sinais iguais é um bocadinho diferente porque no caso do perceptron não há nenhuma correção enquanto no caso da função $\rho(s; \mu)$ temos uma pequena correção porque $|\rho(s^m; \mu) - y^m| \in]0, 1]$. Esta correção é cada vez mais pequena se μ é cada vez mais pequeno ($\rho(s; \mu) \to \text{sng}(s)$.