COMP90051 - Statistical Machine Learning

Lecturer: Ben Rubinstein
Workshop 4 solutions

August 23, 2019

1) **The L2 regularization term $\mathcal{L}_{\text{reg}}(\mathbf{w}) = \frac{1}{2}\lambda\mathbf{w}^T\mathbf{w}$ is commonly said to reduce overfitting. Give a brief justification why?**

The L2 regularization term $\frac{1}{2}||\mathbf{w}||^2$ provides pressure during the optimization procedure on the weights $\mathbf{w}$ to be as small as possible. If we separate the loss $\mathcal{L} = \mathcal{L}_{clf} + \frac{\lambda}{2}||\mathbf{w}||^2$ into a classification and L2 regularization term (ideally we would like to achieve this without compromising classification power, with this tradeoff controlled by the regularization parameter λ), then the gradient descent update term looks like:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla_{\mathbf{w}}\left(\mathcal{L}_{clf} + \frac{\lambda}{2}||\mathbf{w}||^2\right) \tag{1}$$

$$= \mathbf{w}\left(1 - \eta\lambda\right) - \eta\nabla_{\mathbf{w}}\mathcal{L}_{clf} \tag{2}$$

Hence the model weights decay toward zero at each step by the factor $(1 - \eta\lambda)$. If certain weight values are important in the sense that they decrease the classification error, then they should recover at each step. The hope is that weight values that are due to noise in the dataset will not recover similarly, allowing the weights to capture meaningful variation in data rather than noise.

2) **Why do we only include the weights $\mathbf{w}$ in the L2 regularization term $\mathcal{L}_{\text{reg}}(\mathbf{w}) = \frac{1}{2}\lambda\mathbf{w}^T\mathbf{w}$? i.e. the bias terms are excluded from regularization.**

The decision boundary for logistic regression, where $p(y = 1|\mathbf{x}) = p(y = 0|\mathbf{x})$ occurs at $\mathbf{w}^T\mathbf{x}+b$. There is no reason a priori that the decision boundary should occur close to the origin and so penalizing the bias b may hinder learning the appropriate separation of classes.

Furthermore, a small bias term may force the classifier to stay within the region close to the origin where the sigmoid is approximately linear if the weights are not adjusted properly. As we model $p(y = 1|\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + b\right) \approx \mathbf{w}^T\mathbf{x} + b$. So including the bias term helps the model to make more confident predictions.

3) **Given we model the conditional probability of label $y = 1$ to be $p(y = 1|\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + b\right)$, show that prediction is based on a linear decision rule given by the sign of logarithm of the ratio of probabilities:**

$$\log\frac{p(y = 1|\mathbf{x})}{p(y = 0|\mathbf{x})} = \mathbf{w}^T\mathbf{x} + b \tag{3}$$

This is why logistic regression is referred to as a log-linear model. What is the decision boundary for logistic regression?

The conditional probability for an instance to belong to the $y = 1$ class is

$$p(y=1|\mathbf{x}) = \frac{1}{1+\exp\left[-\left(\mathbf{w}^T\mathbf{x}+b\right)\right]} \tag{4}$$

As we are considering the binary case, $p(y=0|\mathbf{x}) = 1 - p(y=1|\mathbf{x})$. Hence the odds ratio is

$$o = \frac{p(y=1|\mathbf{x})}{p(y=0|\mathbf{x})} = \exp\left(\mathbf{w}^T\mathbf{x}+b\right) \tag{5}$$

We predict $y = 1$ if $o > 1$ and $y = 0$ otherwise. The decision boundary corresponds to the points where $p(y=1|\mathbf{x}) = p(y=0|\mathbf{x})$, i.e. the two outcomes are equally likely. By inspecting the odds ratio, this is the (hyper)plane with $\mathbf{w}^T\mathbf{x}+b = 0$.

4) **Consider extending logistic regression to the multiclass case where $y \in \{c_1, \dots c_m\}$. How would you model the conditional probability for class k: $p_k = p(y = c_k|\mathbf{x})$?**

We will drop the bias term for simplicity. We require $p_k \in [0,1]$ and $\sum_k p_k = 1$ by the law of total probability. Taking inspiration from the logistic regression case, we can achieve this by exponentiating the output of our classifier $f(\mathbf{x}) = W^T\boldsymbol{\Phi}(\mathbf{x}) = \left[\mathbf{w}_0^T\boldsymbol{\Phi}|\dots|\mathbf{w}_m^T\boldsymbol{\Phi}\right]$, where $\boldsymbol{\Phi} : \mathbb{R}^d \to \mathbb{R}^D$ is some possibly nonlinear transformation typically mapping the instance $\mathbf{x} \in \mathbb{R}^d$ to some higher-dimensional space, and $\mathbf{w} \in \mathbb{R}^D$, $W \in \mathbb{R}^{D\times m}$. More concretely, the matrix operation looks like:

$$W^T\boldsymbol{\Phi} = \begin{bmatrix} — & \mathbf{w}_0^T & — \\ — & \mathbf{w}_1^T & — \\ & \vdots & \\ — & \mathbf{w}_m^T & — \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}^{(1)} \\ \boldsymbol{\Phi}^{(2)} \\ \vdots \\ \boldsymbol{\Phi}^{(D)} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_0 \cdot \boldsymbol{\Phi} \\ \mathbf{w}_1 \cdot \boldsymbol{\Phi} \\ \vdots \\ \mathbf{w}_m \cdot \boldsymbol{\Phi} \end{bmatrix} \in \mathbb{R}^m \tag{6}$$

This will return a vector of length m. Each dimension of this vector should correspond to the unnormalized probability $\tilde{p}_k$. We then require normalization of the probability output, which can be achieved by dividing by $\sum_k \tilde{p}_k$. Hence we have:

$$p(y=k|\mathbf{x}) = \frac{\exp\left[\left(\mathbf{w}_k^T\Phi(\mathbf{x})\right)\right]}{\sum_n \exp\left[\left(\mathbf{w}_n^T\Phi(\mathbf{x})\right)\right]} \tag{7}$$

This is also known as a Boltzmann distribution. Since the exponential is monotonic, the class prediction is given by

$$\hat{y}(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, \mathbf{w}_k^T\Phi(\mathbf{x}) \tag{8}$$

This corresponds to picking the index of the entry of the largest entry in the RHS of Equation 6. Note that the softmax operation is a mere normalization operation and does not guarantee good calibration or more importantly, uncertainty in the Bayesian sense.