**Technical Notes for Logistic Regression**

Instructor: Jiayu Zhou

April 4, 2017

# 1 Logistic Regression

We limit our discussion on two class case, $\mathcal{C}_+$ and $\mathcal{C}_-$. The materials can be extended to multivariate logistic regression.

- Probabilistic discriminative model
  - Given a data point $\mathbf{x}$, it directly models class conditional probability $\Pr(\mathcal{C}_+|\mathbf{x})$ and $\Pr(\mathcal{C}_-|\mathbf{x})$. Once we figure out the relationship between the two we can generate predictions. Recall that in generative model we model $\Pr(\mathcal{D})$ instead.
  - Therefore we directly use a parameterized (general linear) function to model $\Pr(\mathcal{C}_+|\mathbf{x}) = f(\mathbf{x};\theta)$, where $\theta$ is the parameter to be determined. In the two class case, this is enough because $\Pr(\mathcal{C}_-|\mathbf{x}) = 1 - \Pr(\mathcal{C}_+|\mathbf{x})$.
- The sigmoid function
  - In logistic regression, we use the sigmoid function $\sigma(t) = \frac{1}{1+\exp(-t)}$, i.e.,

$$\Pr(\mathcal{C}_+|\mathbf{x}) = f(\mathbf{x};\theta = \mathbf{w}) = \sigma(\mathbf{x}^T\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{x}^T\mathbf{w})}$$

  - Range: $\sigma(0) = 0.5$, $\sigma(\infty) = \frac{1}{1+\exp(-\infty)} = 1$, and $\sigma(-\infty) = \frac{1}{1+\exp(\infty)} = 0$.
  - Differential $\frac{\partial\sigma(t)}{\partial t} = \sigma(t)(1-\sigma(t))$ :

$$\begin{aligned}\frac{\partial\sigma(t)}{\partial t} &= \frac{\partial\sigma(t)}{1+\exp(-t)} \cdot \frac{\partial(1+\exp(-t))}{\partial\exp(-t)} \cdot \frac{\partial\exp(-t)}{\partial t}\\ &= \frac{1}{1+\exp(-t)} \cdot \frac{\exp(-t)}{1+\exp(-t)}\\ &= \sigma(t)(1-\sigma(t))\end{aligned}$$

  - Property $1-\sigma(t) = \sigma(-t)$:

$$1-\sigma(t) = \frac{1+\exp(-t)}{1+\exp(-t)} - \frac{1}{1+\exp(-t)} = \frac{\exp(-t)}{1+\exp(-t)} = \frac{1}{1+\exp(t)} = \sigma(-t)$$

  - The sigmoid function is a special case of the logistic function $f(t;L,k,t_o) = \frac{L}{1+\exp(-k(t-t_o))}$, where $L$ is the magnitude, $k$ decides how steep is the curve, and $t_0$ is the midpoint. In sigmoid, $L=1, k=1, t_0=0$.
  - The sigmoid function is the inverse of logit function.
- Linear model with an intercept
  - Usually we need an intercept term in the linear model to offset the bias, i.e., $\mathbf{x}_i^T\mathbf{w} + c$ instead of $\mathbf{x}_i^T\mathbf{w}$.

- Typically this can be achieved by adding one dummy variable to the data and use the same algorithm. That is, $\hat{\mathbf{x}}_i = [\mathbf{x}_i; 1]$ and then we add a corresponding extra parameter $c$ in the model so that $\hat{\mathbf{w}} = [\mathbf{w}; c]$. Therefore this gives $\hat{\mathbf{x}}_i^T \hat{\mathbf{w}} = \mathbf{x}_i^T \mathbf{w} + c$.
- Alternatively, we can treat $c$ independently and perform a separate gradient descent. In this case, refer to the *block coordinate descent* technique, where $c$ and $\mathbf{w}$ are considered two blocks. This is useful when there are structured regularization applied on $c$ and $\mathbf{w}$ separately. For example, $\ell_1$ regularized logistic regression only applies $\ell_1$ penalty on $\mathbf{w}$ but not $c$.

## 2 Label Encoding and Logistic Loss

Given different label encoding scheme, we can derive different formulations for logistic regression, leading to different algorithms (because the gradients are different).

- Encoding labels as $\mathcal{C}_+ = +1$ and $\mathcal{C}_- = 0$
  - If we encode this way we can use Bernelle distribution. If random variable $X$ is a random variable with Bernelle distribution, then $\Pr(X = 1) = 1 - \Pr(X - 0) = p$, and probability mass function of the distribution can be written as $\Pr^k (1 - \Pr)^{1-k}$.
  - Likelihood. According to the Bernelle distribution, then we can write

$$\begin{aligned}\Pr(\mathbf{y}|\mathbf{X}; \mathbf{w}) &= \prod_i \Pr(y_i|\mathbf{x}_i; \mathbf{w}) = \prod_i \Pr(y_i|\mathbf{x}_i; \mathbf{w})^{y_i} (1 - \Pr(y_i|\mathbf{x}_i; \mathbf{w}))^{1-y_i} \\ &= \prod_i \sigma(y_i \mathbf{x}_i^T \mathbf{w})^{y_i} (1 - \sigma(y_i \mathbf{x}_i^T \mathbf{w}))^{1-y_i}\end{aligned}$$

  - Loss. Minimize the following:

$$\mathcal{L}_{1,0} = -\frac{1}{N} \ln \Pr(\mathbf{y}|\mathbf{X}; \mathbf{w}) = -\frac{1}{N} \sum \left( y_i \ln \sigma(\mathbf{x}_i^T \mathbf{w}) + (1 - y_i)(1 - \ln \sigma(\mathbf{x}_i^T \mathbf{w})) \right) \tag{1}$$

    The loss has the form of cross-entropy.
  - Gradient

$$\begin{aligned}\nabla_{\mathbf{w}} \mathcal{L}_{1,0} &= -\frac{1}{N} \sum_i \left( y_i \frac{\sigma(\mathbf{x}_i^T \mathbf{w})(1 - \sigma(\mathbf{x}_i^T \mathbf{w}))}{\sigma(\mathbf{x}_i^T \mathbf{w})} \mathbf{x}_i + (1 - y_i) \frac{-\sigma(\mathbf{x}_i^T \mathbf{w})(1 - \sigma(\mathbf{x}_i^T \mathbf{w}))}{1 - \sigma(\mathbf{x}_i^T \mathbf{w})} \mathbf{x}_i \right) \\ &= -\frac{1}{N} \sum_i \left( y_i (1 - \sigma(\mathbf{x}_i^T \mathbf{w})) \mathbf{x}_i + (y_i - 1) \sigma(\mathbf{x}_i^T \mathbf{w}) \mathbf{x}_i \right) \\ &= -\frac{1}{N} \sum_i (y_i - \sigma(\mathbf{x}_i^T \mathbf{w})) \mathbf{x}_i \qquad (2)\end{aligned}$$

- Encoding labels as $\mathcal{C}_+ = +1$ and $\mathcal{C}_- = -1$
  - Since $\Pr(\mathcal{C}_+|\mathbf{x}) = \sigma(\mathbf{x}^T \mathbf{w})$, we have $\Pr(\mathcal{C}_-|\mathbf{x}) = 1 - \sigma(\mathbf{x}^T \mathbf{w}) = \sigma(-\mathbf{x}^T \mathbf{w})$. Given the encoding, we have that $\Pr(y|\mathbf{x}) = \sigma(y \mathbf{x}^T \mathbf{w})$.
  - Likelihood:

$$\Pr(\mathbf{y}|\mathbf{X}; \mathbf{w}) = \prod_i \Pr(y_i|\mathbf{x}_i; \mathbf{w}) = \prod_i \sigma(y \mathbf{x}^T \mathbf{w})$$

- Loss. Minimize the following:

$$\mathcal{L}_{1,-1} = -\frac{1}{N}\ln \Pr(\mathbf{y}|\mathbf{X};\mathbf{w}) = -\frac{1}{N}\sum_i \ln\sigma(y_i\mathbf{x}_i^T\mathbf{w})$$
$$= -\frac{1}{N}\sum_i \ln\frac{1}{1+\exp\left(-y_i\mathbf{x}_i^T\mathbf{w}\right)} = \frac{1}{N}\sum_i \ln\left(1+\exp\left(-y_i\mathbf{x}_i^T\mathbf{w}\right)\right) \tag{3}$$

- Gradient:

$$\nabla_{\mathbf{w}}\mathcal{L}_{1,-1} = -\frac{1}{N}\sum_i \frac{\partial\ln\sigma(y_i\mathbf{x}_i^T\mathbf{w})}{\partial\sigma(y_i\mathbf{x}_i^T\mathbf{w})}\cdot\frac{\partial\sigma(y_i\mathbf{x}_i^T\mathbf{w})}{\partial y_i\mathbf{x}_i^T\mathbf{w}}\cdot\frac{\partial y_i\mathbf{x}_i^T\mathbf{w}}{\partial\mathbf{w}}$$
$$= -\frac{1}{N}\sum_i \frac{1}{\sigma(y_i\mathbf{x}_i^T\mathbf{w})}\cdot\sigma(y_i\mathbf{x}_i^T\mathbf{w})(1-\sigma(y_i\mathbf{x}_i^T\mathbf{w}))\cdot y_i\mathbf{x}_i$$
$$= -\frac{1}{N}\sum_i y_i\sigma(-y_i\mathbf{x}_i^T\mathbf{w})\mathbf{x}_i \tag{4}$$

## 3 Prediction

- In either encoding scheme, given a data point $\mathbf{x}$, the linear function $f(\mathbf{w};\theta) = \mathbf{x}^T\mathbf{w} \geq 0$ simply implies $\Pr(\mathcal{C}_+|\mathbf{x}) \geq \Pr(\mathcal{C}_-|\mathbf{x})$, and is enough to generate the prediction of $\mathbf{x}$ (i.e., $\mathcal{C}_+$), and vice versa.

$$\Pr(\mathcal{C}_+|\mathbf{x}) \geq \Pr(\mathcal{C}_-|\mathbf{x}) \Rightarrow \frac{1}{1+\exp(-\mathbf{x}^T\mathbf{w})} \geq \frac{1}{1+\exp(\mathbf{x}^T\mathbf{w})} \Rightarrow \exp(\mathbf{x}^T\mathbf{w}) \geq \exp(-\mathbf{x}^T\mathbf{w})$$
$$\Rightarrow \mathbf{x}^T\mathbf{w} \geq -\mathbf{x}^T\mathbf{w} \Rightarrow \mathbf{x}^T\mathbf{w} \geq 0$$

- Therefore after the logistic regression model $\mathbf{w}$ is learned on training data, simple linear function $\text{sign}(\mathbf{x}^T\mathbf{w})$ gives prediction results, no matter how $y$ is encoded.

## 4 Implementation Issues

- Vectorization
  - Many BLAS implementations will significantly accelerate the code if variables are properly vectorized.
  - For +1/-1 encoding, the gradient is given by Eq (4). A careful rewriting gives more insights on the vectorized computation:

    $$\nabla_{\mathbf{w}}\mathcal{L}_{1,-1} = -\frac{1}{N}\left(\sum_i \left(y_i\sigma(-y_i\mathbf{x}_i^T\mathbf{w})\right)\mathbf{x}_i\right) = -\frac{1}{N}\mathbf{X}^T\mathbf{a}$$

    where $\mathbf{a} \in \mathbb{R}^N$ and $a_i = y_i\sigma(-y_i\mathbf{x}_i^T\mathbf{w})$.
    - Since many BLAS supports vectorized exp computation, a straightforward implementation of the sigmoid function should be able to take vectorized inputs.
    - The input of sigmoid can be vectorized via $-\mathbf{y} \odot \mathbf{X}\mathbf{w}$
    - $\mathbf{a}$ can thus computed by $\mathbf{a} = \mathbf{y} \odot \sigma(-\mathbf{y} \odot \mathbf{X}\mathbf{w})$

    - The final gradient is given by $\mathbf{X}^T \left(\mathbf{y} \odot \sigma(-\mathbf{y} \odot \mathbf{X}\mathbf{w})\right)/(-N)$
    - PYTHON code:

```
vt = data.T.dot(np.multiply(label,
      sigmoid(np.multiply(-label, data.dot(weights))))) / float(-n)
```

  - For +1/0 encoding, the gradient in Eq (2) is also given by a linear comination of feature vectors:

$$\nabla_{\mathbf{w}}\mathcal{L}_{1,0} = \frac{1}{N}\sum_i \left(\sigma(\mathbf{x}_i^T\mathbf{w}) - y_i\right)\mathbf{x}_i = \frac{1}{N}\mathbf{X}^T\mathbf{b}$$

    where $b_i = \sigma(\mathbf{x}_i^T\mathbf{w}) - y_i$ can be computed similarity as above.

- Loss function overflow

  - The computation of the loss functions in Eqs. 3 and 1 involves the computation of log of sigmoid function, which involves $\exp(\mathbf{x}^T\mathbf{w})$ or $\exp(-y\mathbf{x}^T\mathbf{w})$.
  - Double-precision floating point numbers (i.e., 64-bit IEEE) supports an approximate domain of $a \in (-750, 750)$ when computing $\exp(a)$ before underflowing to 0 or overflow to $+\infty$.
  - Technically the inner product $\mathbf{x}^T\mathbf{w}$ is not bounded, and thus it is very possible that $\exp(-\mathbf{x}^T\mathbf{w})$ will underflow or overflow. While underflow causes less trouble, overflow can lead to serious numerical problems: although $\exp(-\mathbf{x}^T\mathbf{w})$ overflows, $\log\left(\frac{1}{1+\exp(-\mathbf{x}^T\mathbf{w})}\right)$ can still be a meaningful and useful value!
  - Remedy: use $-\log\left(\frac{1}{1+\exp(a)}\right) = \log(\exp(-b) + \exp(a-b)) + b$, where $b = \max(a, 0)$. Therefore the maximum value inside exp is 0. To see this, we have for **any** $b$:

$$\begin{aligned}
-\log\left(\frac{1}{1+\exp(a)}\right) &= \log(1+\exp(a)) = \log(\exp(-b+b) + \exp(a-b+b)) \\
&= \log(\exp(-b)\exp(b) + \exp(a-b)\exp(b)) \\
&= \log((\exp(-b) + \exp(a-b))\exp(b)) \\
&= \log(\exp(-b) + \exp(a-b)) + \log\exp(b) \\
&= \log(\exp(-b) + \exp(a-b)) + b
\end{aligned}$$

  - For example, to compute Eq. (3), we first compute vector $\mathbf{m} \in \mathbb{R}^n$, where $m_i = -y_i\mathbf{x}_i^T\mathbf{w}$. We then compute the maximum $k = \max_i\{m_i\}$. The loss function is computed by

$$\mathcal{L}_{1,-1} = -\frac{1}{N}\sum_i \ln\sigma(y_i\mathbf{x}_i^T\mathbf{w}) = \frac{1}{N}\left(\log(\exp(-k) + \exp(\mathbf{m}-k)) + k\right)$$