# Lec 11.1 Multiclass Logistic Regression

## 1. Multiclass Logistic Regression:

↳ label: $y \in \{1, 2, \cdots, c\}$

↳ Hypothesis:

// One line is not enough to seperate data

Let $\Omega = \{\underline{w}(1), \underline{w}(2) \cdots \underline{w}(c)\}$ be the weight factors for c classes.

Hypothesize that

$$\Pr\{y_n = i \mid \underline{x}_n\} = \frac{e^{\underline{w}(i)^T \underline{x}_n}}{\sum_{j=1}^{c} e^{\underline{w}(j)^T \underline{x}_n}}, \quad \text{for } i = 1, 2, \cdots c$$

"soft max function"

$$\triangleq \hat{P}(i \mid \underline{x}_n)$$ // notation

↳ Error Criteria:

$$e_n(\Omega) = -\log \hat{p}(y_n \mid \underline{x}_n)$$

↳ Gradient:

$$\nabla_\Omega e_n(\Omega) = \begin{bmatrix} \nabla_{\underline{w}(1)} e_n(\Omega) \\ \nabla_{\underline{w}(2)} e_n(\Omega) \\ \vdots \\ \nabla_{\underline{w}(c)} e_n(\Omega) \end{bmatrix}$$

↳ SGD update:

For iteration k, $\Omega_k = \{\underline{w}_k(1), \underline{w}_k(2), \cdots w_k(c)\}$

Pick sample $n \sim \text{uniform}\{1, 2, \cdots N\}$

For $i = 1, 2, \cdots, c$, compute $\nabla_{\underline{w}(i)} e_n(\Omega)$.

Then update according to $\underline{w}_{k+1} = \underline{w}_k(i) - \varepsilon_k \nabla_{w(i)} e_n(\Omega_k)$

↳ Compute $\nabla_{\underline{w}(i)}\, e_n(\Omega)$:

// Note, in here, nth example $(x_n, y_n)$ is chosen.

- Case 1: $i = y_n$,

$$\nabla_{\underline{w}(i)}\, e_n(\Omega) = \nabla_{\underline{w}(i)} \left[ -\log \frac{e^{\underline{w}(y_n)^T \underline{x}_n}}{\sum_{j=1}^{C} e^{\underline{w}(j)^T \underline{x}_n}} \right]$$

$$= \nabla_{\underline{w}(i)} \left[ -\underline{w}(y_n)^T \underline{x}_n + \log \left( \sum_{j=1}^{C} e^{w(j)^T \underline{x}_n} \right) \right]$$

分子部分 log之后，且 $\underline{w}(y_n)^T = \underline{w}(i)^T$

$$= -\underline{x}_n + \frac{1}{\sum_{j=1}^{C} e^{w(j)^T x_n}} \cdot \nabla_{\underline{w}(i)} \left[ \sum_{j=1}^{C} e^{w(j)^T \underline{x}_n} \right]$$

// Note: j is dummy var, i is not

$$= -\underline{x}_n + \frac{e^{w(j)^T \underline{x}_n}}{\sum_{j=1}^{C} e^{w(j)^T x_n}} \cdot x_n$$

- Case 2: $i \neq y_n$

$$\nabla_{\underline{w}(i)}\, e_n(\Omega) = \nabla_{\underline{w}(i)} \left[ -\log \frac{e^{\underline{w}(y_n)^T \underline{x}_n}}{\sum_{j=1}^{C} e^{\underline{w}(j)^T \underline{x}_n}} \right]$$

$$= \nabla_{\underline{w}(i)} \left[ -\underline{w}(y_n)^T \underline{x}_n + \log \left( \sum_{j=1}^{C} e^{w(j)^T \underline{x}_n} \right) \right]$$

$$= \frac{e^{w(j)^T \underline{x}_n}}{\sum_{j=1}^{C} e^{w(j)^T x_n}} \cdot x_n$$

2. Softmax Regression for $C=2$:

$$\hat{p}(1|\underline{x}) = \frac{e^{\underline{w}(1)^T \underline{x}}}{e^{\underline{w}(1)^T \underline{x}} + e^{\underline{w}(2)^T \underline{x}}} = \frac{e^{(\underline{w}(1)-\underline{w}(2))^T \underline{x}}}{e^{(\underline{w}(1)-\underline{w}(2))^T \underline{x}} + 1}$$

$$\hat{p}(2|\underline{x}) = \frac{e^{\underline{w}(2)^T \underline{x}}}{e^{\underline{w}(1)^T \underline{x}} + e^{\underline{w}(2)^T \underline{x}}} = 1 - \hat{p}(1|\underline{x})$$

This is the same as logistic Regression with $\underline{w} = \underline{w}(1) - \underline{w}(2)$

## Lec 11.2 GD / SGD for non-linear regression

1. Recall in the linear regression:

↳ The Ein we wan to minimize is:

$$\boxed{\begin{aligned} E_{in}(\underline{w}) &= \frac{1}{N}\sum_{n=1}^{N} e_n(\underline{w}) \quad \leftarrow \text{a convex function} \\ e_n(\underline{w}) &= (\underline{w}^T \underline{x}_n - y_n)^2 \\ \nabla e_n(\underline{w}) &= \nabla_{\underline{w}} (\underline{w}^T \underline{x}_n - y_n)^2 \\ &= 2(\underline{w}^T \underline{x}_n - y_n) \cdot \nabla_{\underline{w}} (\underline{w}^T \underline{x}_n - y_n) \\ &= 2(\underline{w}^T \underline{x}_n - y_n) \cdot \underline{x}_n \end{aligned}}$$

↳ GD / SGD: as $k \to \infty$, $\underline{w}_k$ converges to the least square sol, which is

$$\underline{w}_{LS} = (\underline{X}^T \underline{X})^{-1} \underline{X}^T \underline{y}$$

↑ A closed form formula.

↳ <u>Why we perfer GD / SGD</u> rater than use this closed form formula?

① Computation complexity

GD : $O(Nd)$   SGD: $O(d)$   Mini batch: $O(Md)$ per iteration

They're smaller comparing to Matrix multiplication

② Exact solution may not be desirable.

We only care about $E_{out}$ (test error), not $E_{in}$

⇒ In practice we can run fewer iterations

stop when error over the validation data is small