ML & Opt Lecture 6

- Practical considerations:
  - Train/valid/test splits
  - Cross-validation
  - Hyperparameter selection
  - Regularization (overfitting)

- Revisiting binary/multiclass logistic regression, linear separability
Logistic Regression

$x \in \mathbb{R}^d$ features used to predict $y \in \{0, 1\}$.

Assume that the log odds of $y$ being 1 is linear in $x$ ($\rho = P(y=1|x)$):

$$\ln \left( \frac{\rho}{1-\rho} \right) = \omega_0 + \sum_{i=1}^{d} \omega_i x_i = \omega_0 + \omega^T x$$

Equivalent:

$$\tilde{x} = \begin{bmatrix} x \end{bmatrix}^T$$

and

$$\tilde{\omega} = \begin{bmatrix} \omega \end{bmatrix}$$

Interpretation: probability of $y = 1$ goes up if $\omega_i > 0$ and $x_i$ increases; goes down if $\omega_i < 0$ and $x_i$ increases; and $\omega_0$ determines the baseline prob of $y = 1$. 
\[ \frac{p}{1-p} = \exp(\omega_0 + \omega^T x) \Rightarrow p = \frac{1}{1 + e^{-\omega_0 + \omega^T x}} \]

\[ p = \sigma(\omega_0 + \omega^T x) \text{ where } \sigma(t) = \frac{1}{1 + e^{-t}} \]

called the logit

\[ \sigma(t) \]

This is the same as saying

\[ y \sim \text{Bern}(\sigma(\omega_0 + \omega^T x)) \]

Goal: recover \( \omega_0 \) & \( \omega \) given training data so in the future when we observe \( x \) we can predict \( y \)
Notice we predict $y$ by choosing $c \in \mathbb{R}^+$ that maximizes

\[ p(y=c|x) = \begin{cases} 
\sigma(\omega^T x) & c=1 \\
-\sigma(\omega^T x) & c=-1 
\end{cases} \]

Notice

\[ 1 - \sigma(t) = 1 - \frac{1}{1 + e^{-t}} = \frac{1 + e^{-t}}{1 + e^{-t}} - \frac{1}{1 + e^{-t}} \]

\[ = \frac{e^{-t}}{1 + e^{-t}} = \frac{1}{e^{t} + 1} \approx \sigma(-t) \]

So

\[ 1 - \sigma(t) = \sigma(-t) \]

This implies

\[ p(y=c|x) = \begin{cases} 
\sigma(\omega^T x) & c=1 \\
\sigma(-\omega^T x) & c=-1 
\end{cases} = \sigma(-c\omega^T x) \]
Now we see that we can predict $y >$ given $x >$ with

$$y = \arg \max_{c \in \mathbb{Z}} \ p(c | x) = \arg \max_{c \in \mathbb{Z}} \ \sigma(-c \omega^T x)$$

Any $x$ for which $\omega_0 + \omega^T x = \ln(1)$ has

$$P(y = 1 | x) = P(y = -1 | x)$$

So we would take $y = -1$
Question: given training data, how to estimate $\omega_0, \omega$

Use MLE to find best separating hyperplane

$$L(\omega_0, \omega) = \frac{1}{n} \sum_{i=1}^{n} -\log p(y_i | x_i ; \omega_0, \omega)$$

NLL of the model parameters $\omega_0, \omega$ given the observed (training) data
Notice

\[ L(\omega_0, \omega) = \frac{1}{n} \sum_{i=1}^{n} -\log \sigma(y_i \omega^T x_i) \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \log \left( \frac{1}{\sigma(y_i \omega^T x_i)} \right) \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \log \left( 1 + e^{-y_i \omega^T x_i} \right) \]

take \ \phi(u, v) = \log \left( 1 + e^{-u v} \right) \ \text{called the logistic loss}

\[ = \frac{1}{n} \sum_{i=1}^{n} \phi(\omega^T x_i, y_i) \]
So fitting the model parameters for logistic regression using MLE is done by solving

$$\hat{\omega}_0, \hat{\omega} = \arg\min_{\omega_0 \in \mathbb{R}, \omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} L(\omega_0 + \omega^T x_i \geq y_i)$$

where $L(u, v) = \log(1 + e^{-uv})$ is the logistic loss fn.

Notice logistic regression always has one solution (when it has a finite solution)

Aside: if the training data is actually linearly separable, and $\omega_0, \omega$ separate the data, then $\alpha \omega_0, \alpha \omega$ also separate the data, and I can choose $\alpha \to \pm \infty$ so that $L(\alpha \omega_0, \alpha \omega) \to 0$
Multiclass Logistic Regression

Given features $x \in \mathbb{R}^d$, predict class $y \in \{k\} = \{1, \ldots, k\}$

Ex: given pixel image $x \in [256]^{m \times n}$ determine
the type of image

Model: assume the log-odds (logits or scores) of
each class, given $x$, relative to class $k$ are linear:

$$
\log \frac{P(Y = 1 \mid X = x)}{P(Y = k \mid X = x)} = b_1 + \omega_1^T x
$$

$$
\vdots
$$

$$
\log \frac{P(Y = k-1 \mid X = x)}{P(Y = k \mid X = x)} = b_{k-1} + \omega_{k-1}^T x
$$

Parameters:

$$
\omega_1, \ldots, \omega_{k-1}, b \in \mathbb{R}^{k-1}
$$

$$
\omega \in \mathbb{R}^{k-1 \times d}
$$
These $k-1$ equations plus the fact that $\sum_{i=1}^{k} P(y = i | x) = 1$ gives us

$$P(y = i | x = x) = \frac{\exp(b_i + \omega_i^T x)}{1 + \sum_{l=1}^{k-1} \exp(b_l + \omega_l^T x)}$$

$i = 1, \ldots, k-1$

$$P(y = k | x = x) = \frac{1}{1 + \sum_{l=1}^{k-1} \exp(b_l + \omega_l^T x)}$$
For ease of use we introduce non-zero logits for class \( k \), i.e. note that

\[
P(Y = k | X) = \frac{1}{\sum_{i=1}^{k-1} \exp(b_i + \omega_i^T x)} \cdot \frac{\exp(b_k + \omega_k^T x)}{\exp(b_k + \omega_k^T x)}
\]

\[
= \frac{\exp(b_k + \omega_k^T x)}{\exp(b_k + \omega_k^T x) + \sum_{i=1}^{k-1} \exp((b_i + b_k) + (\omega_i + \omega_k)^T x)}
\]

Similarly we have

\[
P(Y = i | X) = \frac{\exp((b_i + b_k) + (\omega_k + \omega_i)^T x)}{\exp(b_k + \omega_k^T x) + \sum_{i=1}^{k-1} \exp((b_i + b_k) + (\omega_i + \omega_k)^T x)}
\]
We relabel the shifted biases and weights

\[ b_k \leftarrow b_k \]
\[ \omega_k \leftarrow \omega_k \]
\[ b_i \leftarrow b_i + b_k \] for \( i \neq k \)
\[ \omega_i \leftarrow \omega_i + \omega_k \]

Then we have

\[ P(Y = i \mid X) = \frac{\exp(b_i + \omega_i^T x)}{\sum_{i=1}^{k} \exp(b_i + \omega_i^T x)} \]

for \( i = 1, \ldots, k \)

where \( \text{softmax}(v) = \frac{e^v}{1^T e^v} \)

\[ = \text{softmax}(b + \omega^T x)_i \]
Recall $\sigma : \mathbb{R} \to [0, 1]$.

logits $\xrightarrow{\text{probability}} P = 1$

Now for multiclass logistic regression

$\text{softmax} : \mathbb{R}^k \xrightarrow{} \text{set of probability vectors over the } k \text{ classes for } k \text{ classes}$

softmax is a continuous approximation of the function that maps a vector $v$ to a vector of zeros and ones where there is a one corresponding to the largest entry in $v$, and a zero everywhere else.

Ex:

\[
\begin{bmatrix}
3 & 0 \\
1 & 0
\end{bmatrix}
\xrightarrow{\text{softmax}}
\begin{bmatrix}
1 \\
2.54 \times 10^{-13}
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.9 \\
0.05
\end{bmatrix}
\xrightarrow{\text{softmax}}
\begin{bmatrix}
0.539 \\
0.230
\end{bmatrix}
\]
Given that my data follows the label dist

\[ P(y=i \mid x) = \text{softmax}(w x + b)_i \]

how do I predict the class label given \( x \)?

Take \( x \)

\[ \text{compute logits} \rightarrow \text{compute softmax} \rightarrow P(y=i \mid x) = \text{softmax}(w x + b)_i \]

and take \( x \) as having class

\[ \hat{y} = \arg\max_{i \in [K]} P(y=i \mid x) \]

\[ = \arg\max_{i \in [K]} (w x + b)_i \]
We determine \( \omega \) & \( b \) given training data by minimizing the NLL (i.e. MLE):

\[
L(\omega, b) := \frac{1}{n} \sum_{i=1}^{n} -\log P[Y = y_i \mid X = x_i ; \omega \cdot b]
\]

and take

\[
\hat{\omega}, \hat{b} = \arg\min_{\omega \cdot b} L(\omega, b)
\]

To write the NLL more conveniently, encode the class observations as one-hot vectors. Instead of \( y_i \in [k] \), consider \( y_i \in \{e_1, \ldots, e_k\} \subset \mathbb{R}^k \). E.g. if training example \( i \) is of class 2, we take

\[
y_i = e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \in \mathbb{R}^k
\]
Now we see that if the $i$th training example has class $j$, then $y_i = e_j$ and

$$\log \mathbb{P}(y = y_i | x = x_i) = \log \mathbb{P}(y = e_j | x = x_i)$$

$$= \log \text{softmax}(\mathbf{w} x_i + b)_j$$

$$= \log e_j^T \text{softmax}(\mathbf{w} x_i + b)$$

$$= \log y_i^T \text{softmax}(\mathbf{w} x_i + b)$$

$$= y_i^T \log \left( \frac{\exp(\mathbf{w} x_i + b)}{1^T \exp(\mathbf{w} x_i + b)} \right)$$
\[
\log P[y_i = y_i | X = x_i] = y_i^T (\omega x_i + b) \\
- \log \left( \sum_{l=1}^{k} \exp(\omega x_i + b) \right) \\
= y_i^T (\omega x_i + b) - \text{logsumexp}(\omega x_i + b)
\]
Therefore we get the MCE estimates for the model parameters by solving

\[ \hat{\omega}, \hat{b} = \arg \min_{\omega, b} - \frac{1}{n} \sum_{i=1}^{n} \left[ y_i (\omega x_i + b) - \log \left( \frac{\sum_{l=1}^{K} \exp(\omega x_i + b)_l}{\sum_{l=1}^{K} \exp(\omega x_i + b)_l} \right) \right]\]

question: what is the loss function so that

\[ \frac{1}{n} \sum_{i=1}^{n} l(y_i, \omega x_i + b) = L(\omega, b) ? \]