（日本語は下にあります）

We apply Logistic Regression model on Hand-written digits recognition task.

In logistic regression model, an occur probability of an event is represented by a logistic function. For example, in two-class problem, logistic sigmoid function is commonly used. In multi-class problem, softmax function is known offering a well performance.

In this text, we introduce how a posteriori probability can be represented in logistic function. Next, we imply linear transformation of each category to input patterns and estimate their posteriori probabilities.

Consider a classification task with the number of class C. Using maximum a posteriori probability rule, the output category of an input pattern $\boldsymbol{x}$ is the category y where $p(y = i|\boldsymbol{x})$ is maximum.

A posteriori probability $p(y = i|\boldsymbol{x})$ is,

$$p(y = i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y = i)p(y = i)}{\sum_{j=1}^{C} p(\boldsymbol{x}|y = j)p(y = j)} \tag{1}$$

Here, let

$$a_i = \ln p(\boldsymbol{x}|y = i)p(y = i) \tag{2}$$

Therefore,

$$\begin{aligned} p(y = i|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y = i)p(y = i)}{\sum_{j=1}^{C} p(\boldsymbol{x}|y = j)p(y = j)} \\ &= \frac{\exp(a_i)}{\sum_{j=1}^{C} \exp\,(a_j)} \\ &= \mathrm{softmax}(a_i) \end{aligned} \tag{3}$$

$\mathrm{softmax}(a_i) = \frac{\exp(a_i)}{\sum_{j=1}^{C} \exp\,(a_j)}$ is called softmax function. When $C = 2$, it is known as logistic sigmoid function. From (3), a posteriori probability can be represented by a logistic function.

In multi-class task, we use softmax function. $p(y = i|\boldsymbol{x}) = \mathrm{softmax}(a_i)$. To estimate this probability, we estimate $a_i$ by linear transformation of input pattern $\boldsymbol{x}$ in to $i$-th category. This means, $a_i \approx \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}$ ,where parameter $\boldsymbol{w}_i$ is estimated by maximum likelihood method.

Hence, a posteriori probability is $p(y = i|\boldsymbol{x}) \approx \mathrm{softmax}\left(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\right) = \frac{\exp\left(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\right)}{\sum_{j=1}^{C} \exp\left(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}\right)} \stackrel{\mathrm{def}}{=} \pi_i(\boldsymbol{x})$

Let C be the number of categories,$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ be sample data set. Likelihood function is

$$
\begin{aligned}
&p(y_1, y_2, \dots, y_N | \boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_C) \\
&= \prod_{n=1}^{N} \prod_{k=1}^{C} p(y = k | \boldsymbol{x}_n)^{[\![y_n = k]\!]} = \prod_{n=1}^{N} \prod_{k=1}^{C} \pi_{nk}^{[\![y_n = k]\!]}
\end{aligned} \tag{4}
$$

where

$$
[\![p]\!] = \begin{cases} 1 \text{ if p is true} \\ 0 \text{ if p is false} \end{cases}
$$

$$
\pi_{nk} = p(y = k | \boldsymbol{x}_n) = \frac{\exp(a_{nk})}{\sum_{j=1}^{C} \exp(a_{nj})} \quad , \qquad a_{nj} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_n
$$

Log-likehood can be written as

$$
\begin{aligned}
&L(\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_C) = \ln p(y_1, y_2, \dots, y_N | \boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_C) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{C} [\![y_n = k]\!] \ln \pi_{nk} = \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} \ln \pi_{nk}
\end{aligned} \tag{5}
$$

where $t_{nk} = [\![y_n = k]\!]$

If $\widehat{\boldsymbol{w}}_j$ is an estimator of $\boldsymbol{w}_j$ then $\frac{\partial}{\partial \boldsymbol{w}_j} L(\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_C)\Big|_{\boldsymbol{w}_j = \widehat{\boldsymbol{w}}_j} = 0$

$$
\begin{aligned}
&\frac{\partial L}{\partial \boldsymbol{w}_j} = \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} \frac{\partial L}{\partial \boldsymbol{w}_j} \ln \pi_{nk} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} \frac{\partial \pi_{nk}}{\partial a_{nj}} \frac{\partial a_{nj}}{\partial \boldsymbol{w}_j} = \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} (\delta_{kj} - \pi_{nj}) \boldsymbol{x}_n \\
&= \sum_{n=1}^{N} t_{nj} \boldsymbol{x}_n - \sum_{n=1}^{N} \pi_{nj} \boldsymbol{x}_n \sum_{k=1}^{C} t_{nk} = \sum_{n=1}^{N} (t_{nj} - \pi_{nj}) \boldsymbol{x}_n = 0
\end{aligned} \tag{6}
$$

We cannot analytically solve (6). Here, we use gradient ascent learning method to maximize the log-likelihood function.

Learning Algorithm：

1. Initialize $\boldsymbol{w}_j$
2. Update $\boldsymbol{w}_j$ by 3. while the change of $\boldsymbol{w}_j$ is less than a small value $\varepsilon$
3. $\boldsymbol{w}_j := \boldsymbol{w}_j + \eta \frac{\partial L}{\partial \boldsymbol{w}_j} \left(= \boldsymbol{w}_j + \eta \sum_{n=1}^{N} (t_{nj} - \pi_{nj}) \boldsymbol{x}_n\right)$ , $\eta$: learning rate

手書き数字認識の問題にロジスティック回帰モデルを適応してみる。

ロジスティック回帰モデルでは、事象の生起確率をロジスティック関数で表現する。２クラス問題の場合は、ロジスティックシグモイド関数でモデル化する。多クラス問題では、その拡張であるソフトマック関数でモデル化する。そのため、まず、各クラスの事後確率がロジスティック関数（ソフトマックス関数）で表現できることを示す。
その後、入力パターンに対して、各クラスへの線形変換を導入し、事後確率を計算する。

$C$ クラス問題を考える際、最大事後確率則より、入力パターン$\boldsymbol{x}$の予測クラスは、クラス$i$ の事後確率 $p(y=i|\boldsymbol{x})$が最大となるクラスを選べば良い。
事後確率 $p(y=i|\boldsymbol{x})$は、

$$p(y=i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y=i)p(y=i)}{\sum_{j=1}^{C} p(\boldsymbol{x}|y=j)p(y=j)} \tag{1}$$

となるが、

$$a_i = \ln p(\boldsymbol{x}|y=i)p(y=i) \tag{2}$$

とおけば、

$$\begin{aligned} p(y=i|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y=i)p(y=i)}{\sum_{j=1}^{C} p(\boldsymbol{x}|y=j)p(y=j)} \\ &= \frac{\exp(a_i)}{\sum_{j=1}^{C} \exp\,(a_j)} \\ &= \mathrm{softmax}(a_i) \end{aligned} \tag{3}$$

$\mathrm{softmax}(a_i) = \frac{\exp(a_i)}{\sum_{j=1}^{C} \exp\,(a_j)}$ はソフトマックス関数と呼ばれる。$C=2$の場合、その関数はロジスティックシグモイド関数となる。(3)より、クラス分類問題の事後確率がロジスティック関数で表現できることが示される。

多クラス問題において、ソフトマック関数で事後確率をモデル化する。すなわち、事後確率 $p(y=i|\boldsymbol{x}) = \mathrm{softmax}(a_i)$ とモデル化し、この事後確率を推定するために、$a_i$ を、入力パターン$\boldsymbol{x}$のクラス$i$への線形変換で近似する。すなわち、$a_i \approx \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}$ とし、パラメータ$\boldsymbol{w}_i$は訓練データを利用し、最尤推定で求まる。

この近似より、事後確率$p(y=i|\boldsymbol{x}) \approx \mathrm{softmax}\left(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\right) = \frac{\exp\left(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\right)}{\sum_{j=1}^{C}\exp\left(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}\right)} \stackrel{\text{def}}{=} \pi_i(\boldsymbol{x})$となる。

今クラス数を$C$、訓練データを$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ とする。尤度関数は、

$$
\begin{aligned}
&p(y_1, y_2, \ldots, y_N | \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_C) \\
&= \prod_{n=1}^{N} \prod_{k=1}^{C} p(y = k | \boldsymbol{x}_n)^{[\![y_n = k]\!]} = \prod_{n=1}^{N} \prod_{k=1}^{C} \pi_{nk}^{[\![y_n = k]\!]}
\end{aligned} \tag{4}
$$

ただし、

$$
[\![p]\!] = \begin{cases} 1 \text{ if p is true} \\ 0 \text{ if p is false} \end{cases}
$$

$$
\pi_{nk} = p(y = k | \boldsymbol{x}_n) = \frac{\exp(a_{nk})}{\sum_{j=1}^{C} \exp(a_{nj})} \quad , \qquad a_{nj} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_n
$$

である。

対数尤度関数は、

$$
\begin{aligned}
&L(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_C) = \ln p(y_1, y_2, \ldots, y_N | \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_C) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{C} [\![y_n = k]\!] \ln \pi_{nk} = \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} \ln \pi_{nk}
\end{aligned} \tag{5}
$$

ここで、 $t_{nk} = [\![y_n = k]\!]$ とおいた。

各パラメータ$\boldsymbol{w}_j$の最尤推定$\hat{\boldsymbol{w}}_j$で、$\frac{\partial}{\partial \boldsymbol{w}_j} L(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_C)\Big|_{\boldsymbol{w}_j = \hat{\boldsymbol{w}}_j} = 0$が成立する。

$$
\begin{aligned}
&\frac{\partial L}{\partial \boldsymbol{w}_j} = \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} \frac{\partial L}{\partial \boldsymbol{w}_j} \ln \pi_{nk} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} \frac{\partial \pi_{nk}}{\partial a_{nj}} \frac{\partial a_{nj}}{\partial \boldsymbol{w}_j} = \sum_{n=1}^{N} \sum_{k=1}^{C} t_{nk} (\delta_{kj} - \pi_{nj}) \boldsymbol{x}_n \\
&= \sum_{n=1}^{N} t_{nj} \boldsymbol{x}_n - \sum_{n=1}^{N} \pi_{nj} \boldsymbol{x}_n \sum_{k=1}^{C} t_{nk} = \sum_{n=1}^{N} (t_{nj} - \pi_{nj}) \boldsymbol{x}_n = 0
\end{aligned} \tag{6}
$$

(6)の方程式は解析的な解法で解けないので、対数尤度関数を最大化するために、山登り法（最急勾配降下法）を適用する。

学習アルゴリズム：

4. $\boldsymbol{w}_j$の初期化として適当な値を与える。
5. $\boldsymbol{w}_j$が変化しない(変化分が非常に小さい)まで 3.を繰り返す。
6. $\boldsymbol{w}_j := \boldsymbol{w}_j + \eta \frac{\partial L}{\partial \boldsymbol{w}_j} \left(= \boldsymbol{w}_j + \eta \sum_{n=1}^{N} (t_{nj} - \pi_{nj}) \boldsymbol{x}_n\right)$ ， $\eta$: 学習率