Linear Models

Classification

CS534 - Machine Learning

Yubin Park, PhD

For a binary classification task

i.e. $y \in \{0, 1\}$

Is Squared Loss

a "good" loss function to use?

How about this?

$$y \in \{\text{Pos}, \text{Neg}\}$$

or

$$y \in \{\text{True}, \text{False}\}$$

Such classes may be generated from a coin toss

with a probability p.

For example,

$$y = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } (1 - p) \end{cases}$$

Or,

$$E[y] = p$$

Good news is that

p is a real number,

so we may be able to model p with $\mathbf{x}^T\beta$.

However, bad news is that $0 \le p \le 1$.

Remember: $\mathbf{x}^T\beta$ can be "any" real number.

Here is a trick!

$$0 \le \frac{1}{1 + \exp(-\mathbf{x}^T \beta)} \le 1$$

This is called a logistic function.

So, I can estimate p with

a logistic-transformed linear model.

And, this is called a logistic regression.

The log-likelihood of a coin toss is:

$$y \log(p) + (1 - y) \log(1 - p)$$

Extending this form with our logistic regression for n samples:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n} y_i \log(\frac{1}{1 + \exp(-\mathbf{x}_i^T \beta)}) + (1 - y_i) \log(\frac{\exp(-\mathbf{x}_i^T \beta)}{1 + \exp(-\mathbf{x}_i^T \beta)})$$

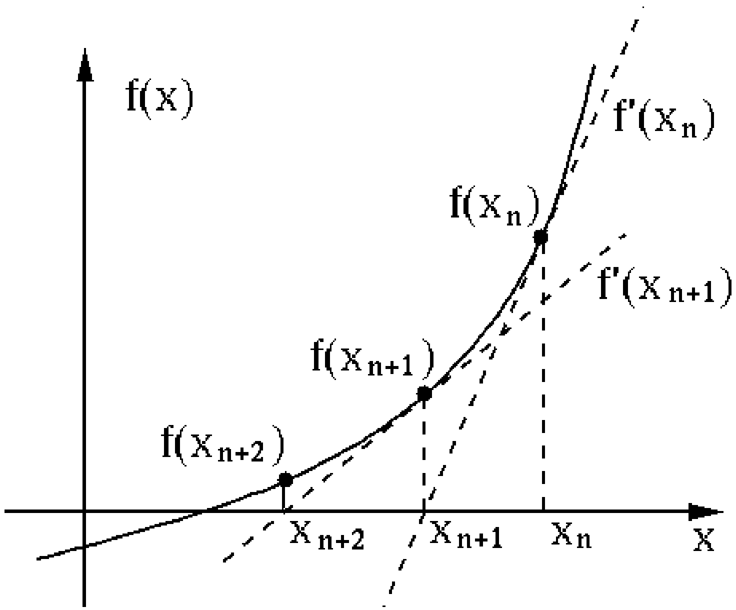
With some arithmetic,

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n} y_i \mathbf{x}_i^T \beta - \log(1 + \exp(\mathbf{x}_i^T \beta))$$

$\hat{\beta}$ that minimizes the above loss function is the solution of:

$$\sum_{i=1}^{n} \mathbf{x}_i (y_i - \frac{1}{1 + \exp(-\mathbf{x}_i^T \beta)}) = 0$$

Newton-Raphson algorithm



http://fourier.eng.hmc.edu/e176/lectures/NM/node20.html

Let $\mathbf{p} = \frac{1}{1+\exp(-\mathbf{X}^T \beta^{\text{old}})}$ and $\mathbf{W} = diag(\mathbf{p}(1 - \mathbf{p}))$

then

$$\beta^{\text{new}} = \beta^{\text{old}} + (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p})$$

This algorithm is known as

Iterative Reweighted Least Squares (IRLS).

IRLS (1)

```
data = load_breast_cancer()
...

def loss(y, X, beta):
    ll = y * np.dot(X, beta) - np.log(1 + np.exp(np.dot(X, beta)))
    return -np.sum(ll)

beta = np.zeros(m+1)
loss_lst = []
loss_lst.append(loss(y, X, beta))
for i in range(30):
    p = expit(np.dot(X, beta))
    W = np.diag(p * (1-p))
    XWXinv = np.linalg.inv(np.dot(np.dot(X.T, W), X))
    beta = beta + np.dot(XWXinv, np.dot(X.T, y-p))
    loss_lst.append(loss(y, X, beta))
print(loss_lst)
```

```
[394.40074573860886, 134.65976155779023, 77.34464088185803, 50.213695232241705,
36.03778398459961, 27.9193029909705, 21.317288480218703, 17.693240538595294,
15.486294065947352, 13.69064323793789, 11.138891617659137, 8.972106451009168,
6.815535841033875, inf, inf, inf, inf, inf, inf, inf, inf, ...]
```

IRLS (2)

The issues with $\lim_{x\to 0} \log x$ and $\lim_{x\to 0} \frac{1}{x}$.

A heuristic fix:

```
eps = 1e-5
beta = np.zeros(m+1)
loss_lst = []
loss_lst.append(loss(y, X, beta))
for i in range(30):
    p = np.clip(expit(np.dot(X, beta)), eps, 1-eps) # NOTE: np.clip()
    W = np.diag(p * (1-p))
    XWXinv = np.linalg.inv(np.dot(np.dot(X.T, W), X))
    beta = beta + np.dot(XWXinv, np.dot(X.T, y-p))
    loss_lst.append(loss(y, X, beta))
print(loss_lst)
```

```
[394.40074573860886, 134.65976155779023, 77.34464088185803, 50.213695232241705,
36.037792317633716, 27.921357429458112, 21.363127719004243, 17.7941874322518,
15.712620105268597, 14.313956463333824, 13.059598232213126, 12.127395626682928,
11.44752822203149, 10.928370872078371, 10.516914209497251, 10.18009512545763,
9.89655605214852, 9.652059524611392, 9.436848965755562, ...]
```

Log-odds

For a binary classification problem,

$$\log \frac{p_i}{1 - p_i} = \mathbf{x}_i^T \beta$$

The left-hand term is called the log-odds or logit function.

$$\mathrm{logit}(p) = \log \frac{p}{1 - p}$$

$$\mathrm{logit}^{-1}(q) = \frac{1}{1 + \exp(-q)}$$

Logistic regression can be viewed as a linear model for the log-odds target.

Coefficients

A central limit theorem and the least sqaure update form can be used to derive:

$$\hat{\beta} \sim N(\beta, (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1})$$

In logistic regression, the regression coefficient represents the change in the log-odds.

$$\exp(\beta_j) = \frac{\Pr(y = 1 | x_j = 1) / \Pr(y = 0 | x_j = 1)}{\Pr(y = 1 | x_j = 0) / \Pr(y = 0 | x_j = 0)}$$

This is called as Odds-ratio (OR). In other words,

$$\exp(\beta_j) = \frac{\text{Odds of } y = 1 \text{ with } x_j = 1}{\text{Odds of } y = 1 \text{ with } x_j = 0}$$

Probabilistic Perspective

$$y = \begin{cases} 1, & \text{if } \mathbf{x}\beta + \epsilon > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\Pr(\epsilon > s) = \frac{1}{1 + \exp(-s)}$$

The noise is from the standard logistic distribution. To see how this works,

$$\Pr(\mathbf{x}\beta + \epsilon > 0) = 1 - \Pr(\epsilon < -\mathbf{x}\beta)$$

$$\Pr(y = 1) = \frac{1}{1 + \exp(-\mathbf{x}\beta)}$$

Other Approaches for Binary Classification

$$y = \begin{cases} 1, & \text{if } \mathbf{x}\beta + \epsilon > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\epsilon \sim N(0, 1)$$

This model is known as Probit Regression.

If the cumulative distribution of the noise follows the standard Gumbel distribution

$$\Pr(\epsilon > s) = \exp(-\exp(-s))$$

then the model is known as the complementary log-log regression.

Note that a logistic regression is just **one way** to model a binary classification problem.

Regularized Logistic Regression

Elastic-Net:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n}[y_i \mathbf{x}_i^T \beta - \log(1 + \exp(\mathbf{x}_i^T \beta))] + \lambda((1-\alpha)\beta^T \beta + \alpha|\beta|)$$

Ridge:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n}[y_i \mathbf{x}_i^T \beta - \log(1 + \exp(\mathbf{x}_i^T \beta))] + \lambda\beta^T \beta$$

Lasso:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n}[y_i \mathbf{x}_i^T \beta - \log(1 + \exp(\mathbf{x}_i^T \beta))] + \lambda|\beta|$$

Taylor Approximation

Applying the quadratic Taylor approximation to the logistic loss part,

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) \approx \sum p_i(1-p_i)(\mathbf{x}_i^T\beta - \frac{y_i - p_i}{p_i(1-p_i)} - \mathbf{x}_i^T\beta^{\text{old}})^2 + \lambda|\beta|$$

where $p_i = \frac{1}{1+\exp(-\mathbf{x}_i^T\beta^{\text{old}})}$.

Let $z_i = \frac{y_i - p_i}{p_i(1-p_i)} + \mathbf{x}_i^T\beta^{\text{old}}$ and $w_i = p_i(1-p_i)$, then

$$\sum w_i(z_i - \mathbf{x}_i^T\beta)^2 + \lambda|\beta|$$

This form is the same as a weighted Lasso loss function, which you can solve with a proximal gradient method.

You can extend this result to the Elastic-Net penalty - note that you can modify the loss function of an Elastic-Net regression to be equivalent to a lasso regression.

Generalized Linear Models

Multi-class and count targets

Multi-class Target

Consider modeling $y \in \{\text{Red, Green, Yellow}\}$.

How about

$$\Pr(y = \text{Red}) = \frac{\exp(\mathbf{x}^T \beta^R)}{\exp(\mathbf{x}^T \beta^R) + \exp(\mathbf{x}^T \beta^G) + \exp(\mathbf{x}^T \beta^Y)}$$

The above probability distribution is known as a softmax function.

The log-likelihood of this model is:

$$\sum_{i=1}^{n} \sum_{k=\{R,G,Y\}} I(y_i = k)\mathbf{x}_i^T \beta^k - \log(1 + \exp(\mathbf{x}_i^T \beta^k))$$

Count Target

Consider modeling $y \in \{0, 1, 2, 3, \ldots\}$.

How about

$$\Pr(y = k) = \frac{1}{k!}\exp(k\mathbf{x}^T\beta)\exp(-\exp(\mathbf{x}^T\beta))$$

If we let $\lambda = \exp(\mathbf{x}\beta)$, then

$$\Pr(y = k) = \frac{1}{k!}\lambda^k \exp(-\lambda)$$

This distribution is known as Poisson distribution. The log-likelihood of this model is:

$$\sum_{i=1}^{n} y_i\mathbf{x}_i^T\beta - \exp(\mathbf{x}_i^T\beta)$$

Generalized Linear Model

$$E[y] = \mu = g^{-1}(\mathbf{x}^T\beta)$$

where $g(\cdot)$ is a link function.

Distribution	**Use Case**	**Link**	**Mean Function**
Normal	Linear Response	$\mathbf{X}\beta = \mu$	$\mu = \mathbf{X}\beta$
Poisson	Count Response	$\mathbf{X}\beta = \log(\mu)$	$\mu = \exp(\mathbf{X}\beta)$
Categorical	K Response	$\mathbf{X}\beta = \log(\frac{\mu}{1-\mu})$	$\mu = \frac{1}{1+\exp(-\mathbf{X}\beta)}$

The maximum likelihood estimator for β can be found using IRLS or Coordinate Descent.

Please read "GLMNET Vignette" for more details.

Questions?