DM825

Introduction to Machine Learning

Lecture 3
Logistic Regression and Model Assessment in Regression

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Binary Variables
2. Logistic Regression
3. Model Assessment and Selection

We saw three ways to derive the parameter of Linear Models

- least square loss
- maximum likelihood approach
- Bayesian approach

Outline

Binary Variables

- We toss a coin a number of times and wish to learn the probability of the coin.

- $x \in \{0, 1\}$

- μ probability of getting 1: $p(x = 1 \mid \mu) = \mu$

- $\mathrm{Bern}(x \mid \mu) = \mu^x (1 - \mu)^{1-x}$ Bernoulli distribution
  $E[x] = \mu$, $\mathrm{Var}[x] = \mu(1 - \mu)$

- $\mathcal{D} = \{x^1, \ldots, x^m\}$ observed values

- likelihood: prob. of $\mathcal{D}$ under the assumption that data are i.i.d from $p(x \mid \mu)$:

$$p(\mathcal{D} \mid \mu) = \prod_{i=1}^{m} p(x_i \mid \mu)$$

Frequentist approach

$$\max \log p(\mathcal{D} \mid \mu) = \sum_{i=1}^{m} \log p(x^i \mid \mu) = \sum_{i=1}^{m} x^i \log \mu + (1 - x^i) \log(1 - \mu)$$

derivative wrt μ to null:

$$\mu_{ML} = \frac{1}{m} \sum_{i=1}^{m} x^i$$

Hence, with 3 heads in 3 tosses we get $\mu_{ML} = 1$, which sounds like an unreasonable overfitting

Bayesian approach
We know the likelihood as expressed in the previous slide, we need to choose a prior distribution $p(\mu)$.

Conjugacy property: the posterior has the same functional form as the prior.

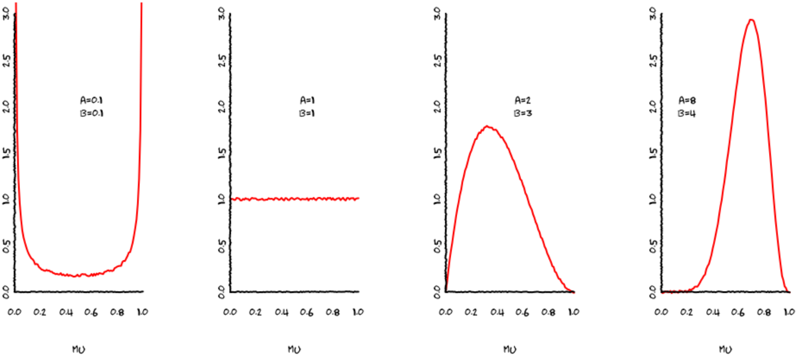
A distribution that has this property is the beta distribution:

$$\mathrm{Beta}(\mu \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$$

$\Gamma(z) = \int_0^\infty u^{z-1}e^{-u}du$ and $\Gamma(z+1) = z\Gamma(z)$. $\Gamma(1) = 1$ and $\Gamma(z) = z!$

$$E[\mu] = \frac{a}{a+b} \qquad \mathrm{Var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)}$$

The density of beta for different values of a and b, in this context called the hyperparameters.



Step 1: from Bayes theorem, with a batch learning approach in which k number of head in the observations:

$$p(\mu \mid k, m-k, a, b) \propto \frac{\Gamma(a+b+m)}{\Gamma(k+a)\Gamma(m-k+b)} \mu^{(k+a)-1}(1-\mu)^{(m-k+b)-1}$$

sequential learning approach: update at each observation, if additional data arrive, the posterior becomes prior

Step 2: express the predictive distribution and predict the value:

$$p(X=1 \mid \mathcal{D}) = \int_0^1 p(X=1 \mid \mu, \mathcal{D}), p(\mu \mid \mathcal{D}) d\mu$$

$p(x)=\int p(x,y)dy$
sum rule

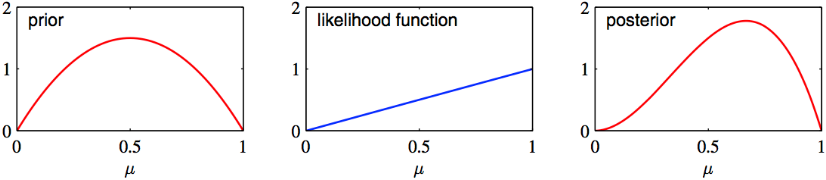
$$= \int_0^1 \mu p(\mu \mid \mathcal{D}) d\mu = E[\mu \mid \mathcal{D}]$$

$p(x,y)=p(y|x)p(x)$
product rule

$$= \frac{k+a}{a+b+m}$$

because expected value of beta distribution

fraction of real observations and fictitious prior distribution.
for $k \to \infty$ reduces to max likelihood.



Outline

Classification and Logistic Regression

Binary classification problem: $Y = \{0, 1\}$ or $Y = \{-1, 1\}$ (labels)

- We could use the linear regression algorithms that we saw and round.
- Or: we change our hypothesis:

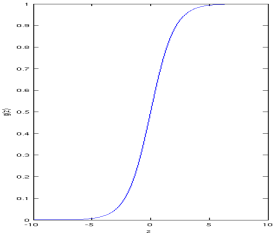
$$h_{\vec{\theta}}(\vec{x}) = g(\vec{\theta}^T \vec{x}) \qquad g : \mathbb{R} \to [0, 1], h : \mathbb{R}^p \to [0, 1].$$

In ML $g(\cdot)$ is called activation function
In statistics $g^{-1}(\cdot)$ is called link function

A common choice for g is the logistic function or sigmoid function:

$$g(z) = \frac{1}{1 + e^{-z}}, \qquad \text{hence}$$

$$h_{\vec{\theta}}(\vec{x}) = \frac{1}{1 + e^{-\vec{\theta}^T \vec{x}}}$$



- Note that g is nonlinear in both the parameters and the inputs

- However, the decision surface corresponds to $h(\vec{x}) = \text{constant}$ and hence to a linear function of $\vec{x}$ ($\vec{\theta}^T\vec{x} = g^{-1}(\text{constant}) = \text{constant}$)

- for later use the derivative of the sigmoid function is:

$$
\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} = \frac{d}{dz}\frac{1}{(1+e^{-z})^2}e^{-z} \\
&= \frac{1}{1+e^{-z}}\left(1 - \frac{1}{1+e^{-z}}\right) = g(z)(1-g(z))
\end{aligned}
$$

- how do we fit $\vec{\theta}$?

- Remark: the methods we see remain valid if we use basis functions in place of $\vec{x}$, that is, $\vec{\phi}(\vec{x})$

Maximum likelihood approach:
Let's assume that:

$$\Pr(y = 1 \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x})$$

$$\Pr(y = 0 \mid \vec{x}; \vec{\theta}) = 1 - h_{\vec{\theta}}(\vec{x})$$

Then, the likelihood for one single example is a Bernoulli distribution

$$\Pr(y \mid \vec{x}, \vec{\theta}) = h_{\vec{\theta}}(\vec{x})^y (1 - h_{\vec{\theta}}(\vec{x}))^{1-y}$$

and for m i.i.p. training examples:

$$L(\vec{\theta}) = p(\vec{y} \mid \mathbf{X}, \vec{\theta}) = \prod_{i=1}^{m} p(y^i \mid \vec{x}^i, \vec{\theta})$$

$$= \prod_{i=1}^{m} h_{\vec{\theta}}(\vec{x}^i)^{y^i} (1 - h_{\vec{\theta}}(\vec{x}^i))^{1-y^i}$$

$$\log L(\vec{\theta}) = \sum_{i=1}^{m} y^i \log h(x^i) + (1 - y^i) \log(1 - h(x^i))$$

To maximize we use the gradient descent: $\theta_j := \theta_j + \alpha \nabla_{\theta_j} \log L(\vec{\theta})$

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} &= \left(y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1-g(\theta^T x)} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x) \\
&= \left(y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1-g(\theta^T x)} \right) g(\theta^T x)(1-g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x \\
&= (y(1-g(\theta^T x)) - (1-y) g(\theta^T x)) x_j = (y - h_\theta(x)) x_j
\end{aligned}$$

$$\theta_j := \theta_j + \alpha(y - h_{\vec{\theta}}(\vec{x})) x_j$$

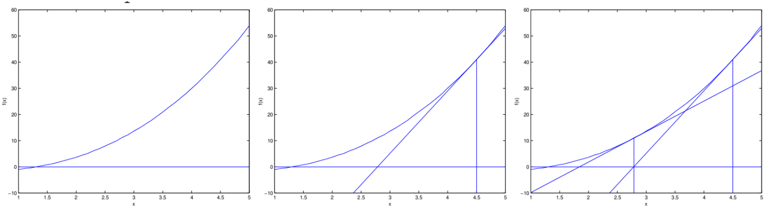
hence the update rule remains the same even though h is now nonlinear.

Newton-Raphson Method

Newton method to find zeros of a function in one dimension:

$$\theta := \theta - \frac{f(\theta)}{f'(\theta)}$$

moves to point where tangent meets zero.



Minimizing a function corresponds to set its first derivative to zero hence:

$$\theta := \theta - \frac{f'(\theta)}{f''(\theta)} \quad \text{finds minima}$$

In n dimensions:

$$\theta := \theta - H^{-1}\nabla_\theta f(\theta) \qquad H_{ij} = \frac{\partial^2 f(\theta)}{\partial\theta} \qquad \text{Hessian}$$

Outline

Loss Functions for Regression

Loss function

$$L(Y, h(\vec{X})) = (Y - h(\vec{X}))^2 \qquad L(\vec{Y}, h(\mathbf{X})) = \sum_i (Y^i - h(\vec{X}^i))^2 \tag{1}$$

$$L(Y, h(\vec{X})) = |Y - h(\vec{X})| \qquad L(\vec{Y}, h(\mathbf{X})) = \sum_i |Y_i - h(\vec{X}^i)| \tag{2}$$

We saw that (1) has a probabilistic interpretation which makes it appealing.

Training error

$$\overline{err} = \frac{1}{m}\left[\sum_{i=1}^{m} L(y^i, h(\vec{x}^i))\right]$$

Test error or generalization error (expected prediction error):
$Err = EPE = E[L(y, h(\vec{x}))], \qquad (y, \vec{x})$ drawn from test set

Expected loss:

$$E[L] = \int\int L(y, y(\vec{x}))p(\vec{x}, y)d\vec{x}dy = \int\int [y(\vec{x}) - y]^2 p(\vec{x}, y)d\vec{x}dy$$

which function $y(\vec{x})$ minimizes $E[L]$?

$$\frac{\partial E[L]}{\partial y(\vec{x})} = 2\int [y(\vec{x}) - y]p(\vec{x}, y)dy = 0 \qquad \text{solving in } y(\cdot):$$

$$y(\vec{x}) = \frac{\int yp(\vec{x}, y)dy}{p(\vec{x})} = E_y[y \mid \vec{x}]$$

that is, the optimal solution is the expectation conditional on $\vec{x}$

- We used this fact already with the probabilitic interpretation.
- It is also the outcome with the least square method.
- The next slide shows another way to obtain this result.

Bias-Variance Decomposition

by adding and removing $E_y[y \mid \vec{x}]$ in $[y(\vec{x}) - y]^2$:

$$E[L] = \int \{y(\vec{x}) - E[y|\vec{x}]\}^2 \, p(\vec{x}) d\vec{x} + \int \{E[y|\vec{x}] - y\}^2 \, p(\vec{x}) d\vec{x}$$

- first term vanishes when $y(\vec{x}) = E_y[y \mid \vec{x}]$
- second term is the variance of the distribution of y averaged over $\vec{x}$, it is intrinsic variability of target data and can be regarded as noise. (irreducible)

In practice we have a limited number of observations $\mathcal{D}$ therefore the exact $y(\vec{x}) = E_y[y \mid \vec{x}]$ cannot be found. We use instead a parametric function $h(\vec{\theta}, \vec{x})$.

To estimate the perofmance of a learning algorithm we average over an ensamble of data sets $\mathcal{D}$. We add and remove $E_{\mathcal{D}}[y(\vec{x}, \mathcal{D})]$

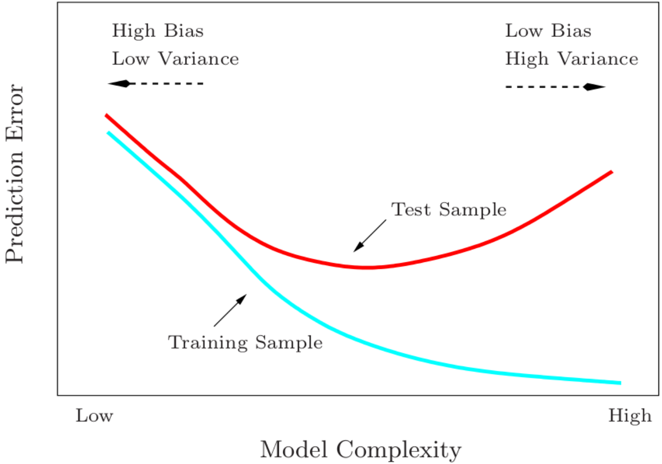
$$
\begin{aligned}
E_{\mathcal{D}}[\{y(\vec{x}, \mathcal{D}) - h(\vec{x})\}^2] =& \{E_{\mathcal{D}}[\{y(\vec{x}, \mathcal{D})] - h(\vec{x})\}^2 \\
&+ E_{\mathcal{D}}[\{y(\vec{x}, \mathcal{D}) - E_{\mathcal{D}}[y(\vec{x}, \mathcal{D})]\}^2].
\end{aligned}
$$

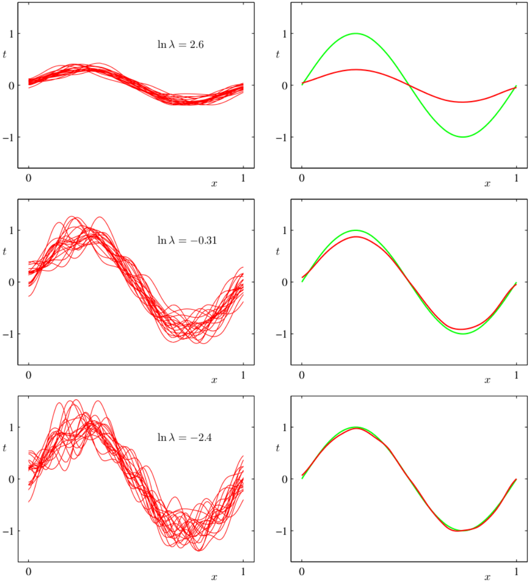
$$
\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}
$$

where

$$
\begin{aligned}
(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{noise} &= \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t
\end{aligned}
$$

Trade off


High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Training Sample
Low
High
Model Complexity


$\ln \lambda = 2.6$
$\ln \lambda = -0.31$
$\ln \lambda = -2.4$
t
x
1
0
−1