Generalized Linear Regression and Classification

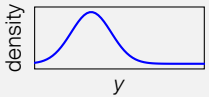
Johanni Brea

Introduction to Machine Learning

Table of Contents

The Normal, Bernoulli and Categorical Distribution

Normal



$$p(y|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-f(x))^2}{2\sigma^2}}$$

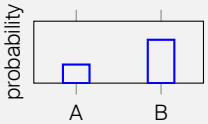
$f(x)$: a number

mean: $f(x)$

variance: σ^2

mode: $f(x)$

Bernoulli



$$p(\mathrm{A}|x) = p_\mathrm{A} = \sigma(f(x))$$

$f(x)$: a number

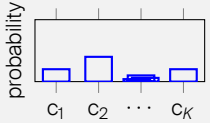
sigmoid/logistic function

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$p(\mathrm{B}|x) = 1 - p_\mathrm{A} = \sigma(-f(x))$$

mode: A if $p_\mathrm{A} > p_\mathrm{B}$

Categorical



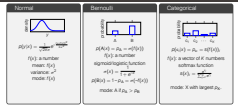
$$p(\mathrm{c}_i|x) = p_{\mathrm{c}_i} = \mathrm{s}(f(x))_i$$

$f(x)$: a vector of K numbers

softmax function

$$\mathrm{s}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

mode: X with largest p_X.



Notes

- You have probably already seen the normal (Gaussian), the Bernoulli and the Categorical distribution. What is special here, is that the distribution depends on the input through some function $f(x)$, e.g. the mean of the normal can be different for different inputs or the probability of class C_1 can depend on the input.
- The function $f(x)$ can be anything! In this lecture we assume it is linear, i.e. $f(x) = \theta_0 + \theta_1 x$. Later in this course, $f(x)$ could be a neural network or some other non-linear function.
- If the response variable Y is real-valued, we can take the normal or some other distribution, like the Laplacian. If the response is binary, it is natural to take the Bernoulli and if the response can be in one of $K > 2$ classes, it is natural to take the Categorical distribution to model the conditional data generating process of the response Y given the input X.

Blackboard: Maximum Likelihood Estimation (Categorical)

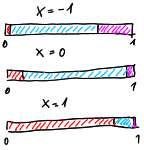
Data Generating Process

$$P(Y = C_1 | x) = \frac{e^{5x+1}}{e^{5x+1} + e^{x+3} + e^{-x}}$$

$$P(Y = C_2 | x) = \frac{e^{x+3}}{e^{5x+1} + e^{x+3} + e^{-x}}$$

$$P(Y = C_3 | x) = \frac{e^{-x}}{e^{5x+1} + e^{x+3} + e^{-x}}$$

$$f(x;\theta) = \begin{pmatrix} \theta_{10} + \theta_{11} x \\ \theta_{20} + \theta_{21} x \\ \theta_{30} + \theta_{31} x \end{pmatrix}$$



Family of Distributions

$$P(Y = c_i | x, \theta) = s(f(x;\theta))_i$$

Training Data

$$((x_1 = 0, y_1 = C_2), (x_2 = -1, y_2 = C_3), (x_3 = 1, y_3 = C_1), (x_4 = 2, y_4 = C_1))$$

Log-Likelihood Function

$$\log \ell(\theta) = \sum_{i=1}^{n} \log P(y_i | x_i, \theta)$$

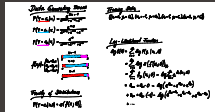
$$= \sum_{i=1}^{n} \log s(f(x_i;\theta))_{y_i}$$

$$= \sum_{i=1}^{n} f_{y_i}(x_i;\theta) - \log \sum_{k=1}^{3} e^{f_k(x_i;\theta)}$$

$$= \theta_{20} + \theta_{21} \cdot 0 - \log\left(e^{\theta_{10}} + e^{\theta_{20}} + e^{\theta_{30}}\right)$$

$$+ \theta_{30} + \theta_{31} \cdot (-1) - \log\left(e^{\theta_{10} - \theta_{11}} + e^{\theta_{20} - \theta_{21}} + e^{\theta_{30} - \theta_{31}}\right)$$

$$+ \dots$$

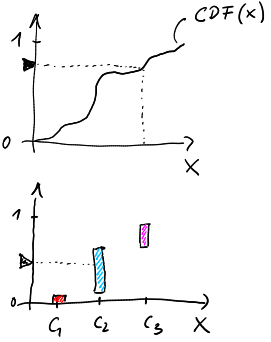


Notes

To generate data from the conditional data generating process, we use inverse transform sampling. Inverse transform sampling (also known as inversion sampling, the inverse probability integral transform, the inverse transformation method, or the Smirnov transform) is a basic method for pseudo-random number sampling, i.e., for generating sample numbers at random from any probability distribution given its cumulative distribution function.(source: `https://en.wikipedia.org/wiki/Inverse_transform_sampling`)

In the case of the categorical distribution, this leads to the following procedure:

1. for a given X, compute the conditional probabilities $P(Y = c_i|x)$ for all classes c_i, and form the cumulative distribution (visually: stack the probabilities as in the colored bars on the slide).
2. sample U uniformly from $[0, 1]$; if U falls in the cumulative distribution into the region of class c_i, set $Y = c_i$.



Blackboard: Maximum Likelihood Estimation (Bernoulli)

Data Generating Process

$P(y = A \mid x) = \text{Bernoulli}(2x - 1)$

$Y = A$ if $\sigma(2x - 1) > \varepsilon$, $\varepsilon \sim \text{Uniform}([0,1])$

Training Data

$((x_1 = 0, y_1 = B), (x_2 = 2, y_2 = A), (x_3 = 3, y_3 = B))$

Family of Distributions

$P(y = A \mid x, \theta) = \sigma(\theta_0 + \theta_1 x)$

Log-Likelihood Function

$$\log \ell(\theta) = \log \ell(\theta_0, \theta_1) = \sum_{i=1}^{n} \log P(y_i \mid x_i, \theta)$$

$$= \log \sigma(-\theta_0) + \log \sigma(\theta_0 + 2\theta_1) + \log \sigma(-\theta_0 - 3\theta_1)$$

Optimizer: Default

Solution: $\hat{\theta}_0 \approx -1.3$ $\hat{\theta}_1 \approx 0.3$ $\log \ell(\hat{\theta}) \approx -1.9$

Test Log-Likelihood at x_0: $E[\log P(Y \mid x_0)]$

$$\underbrace{\sigma(2x_0 - 1)}_{P(Y=A \mid x_0)} \cdot \log \underbrace{\sigma(0.3x_0 - 1.3)}_{P(Y=A \mid x_0, \hat{\theta})} + \underbrace{\sigma(-2x_0 + 1)}_{P(Y=B \mid x_0)} \log \underbrace{\sigma(-0.3x_0 + 1.3)}_{P(Y=B \mid x_0, \hat{\theta})}$$

Notes

- **Data Generating Process:** If the data is generated by a Bernoulli process with probability of class A equal to $\rho \in [0,1]$ and probability of class B equal to $1-\rho$, there is a simple way to sample data: sample a random number ϵ uniformly distributed in $[0,1]$; if $\rho \geq \epsilon$ take class A otherwise take class B. This works, because the probability that ϵ is smaller than ρ is exactly ρ (and $1-\rho$ for being larger than ρ). Here the probability of class A depends on the input, so $\rho = \sigma(2x-1)$.
- **Test Log-Likelihood at x_0:** In short: we are measuring how (log-)likely it is to generate label Y given a fixed, fitted model, weighted by how likely it is that the true generator samples Y.
  - We want to compute the expected log-probability of giving the correct response at a given x_0 with the fitted parameters $\hat{\theta}$, i.e. $E_{Y|x_0}[\log P(Y|x_0, \hat{\theta})]$.
  - We know $P(Y = \mathrm{A}|x_0, \hat{\theta}) = \sigma(0.3x_0 - 1.3)$ and $P(Y = \mathrm{A}|x_0) = \sigma(2x_0 - 1)$
  - We can only compute this expectation here, because we know the true conditional data generating process $P(Y|X)$. In practice we never know the true conditional data generating process (and if we would know, we would not need machine learning to approximate the generator :)). In practice we would rather estimate the test log-likelihood of the joint data generating process with a test set.

Cross-Entropy Loss

- Entropy of a distribution over a discrete variable: $H(P) = -\sum_x P(x) \log P(x)$
- Cross-Entropy: $H(P, Q) = -\sum_x P(x) \log Q(x) \geq H(P)$
- Maximizing the Log-Likelihood for some training set $\mathcal{D} = \big((x_1, y_1), \ldots, (x_n, y_n)\big)$ is equivalent to minimizing the cross-entropy for $P(x) = \sum_{i=1}^{n} \delta(x - x_i)$.

Nomenclature

For some models (families of probability distributions) with linear function $f(x)$ we see occasionally specific names for the likelihood maximizing machine.

- Gaussian (normal distribution): Linear Regression
- Bernoulli: Logistic Regression or Linear (Binary) Classification
- Categorical: Multinomial Logistic Regression or Multiclass Linear Regression (or Classification)
- Poisson: Poisson Regression

Later we will see that there are natural generalizations for all these models with non-linear $f(x)$, where $f(x)$ is for example given by a neural network.

Quiz

Correct or wrong?

1. The only difference between linear regression and linear classification is in the choice of the conditional distribution $P(Y|f_\theta(x) = \theta_0 + \theta_1 x)$.
2. The softmax function has the property $\sum_{i=1}^{K} s(x)_i = 1$.
3. For any model where we know the likelihood we can formulate an equivalent loss minimization perspective by defining the loss function as the negative log-likelihood function, i.e. $L(y, f_\theta(x)) = -\log P(y|f_\theta(x))$.

Table of Contents

Spam Classification

spam

Subject: follow up
here ' s a question i ' ve been wanting to ask you , are you feeling down but too embarrassed to go to the doc to get your m / ed ' s ?
here ' s the answer , forget about your local p harm . acy and the long waits , visits and embarassments . . do it all in the privacy of your own home , right now . http : / / chopin . manilamana . com / p / test / duet it ' s simply the best and most private way to obtain the stuff you need without all the red tape .

Feature Representation

There are many ways to extract useful features from text. Here we use a very simple "bag of words" approach: word counts for a lexicon of size p.

E.g.

X_1 (your)	X_2 (need)	X_3 (pay)	$\cdots$	X_p (red)
3	1	0	$\cdots$	1

All n emails get such a representation.

Multiple Logistic Regression

$$\Pr(Y = \text{spam}|X) = \sigma(\theta_0 + \theta_1 X_1 + \cdots + \theta_p X_p)$$

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \sigma(0) = 0.5 \quad \sigma(-\infty) = 0 \quad \sigma(\infty) = 1$$

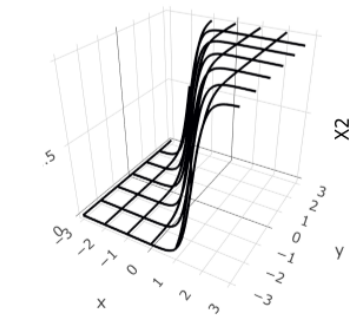
Find $\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_p$ that maximize the likelihood function.

Predictions (at **decision threshold** 0.5):
A new email is classified as spam, if its feature representation x leads to
$\sigma(\hat{\theta}_0 + \hat{\theta}_1 x_1 + \cdots + \hat{\theta}_d x_d) \geq 0.5$.

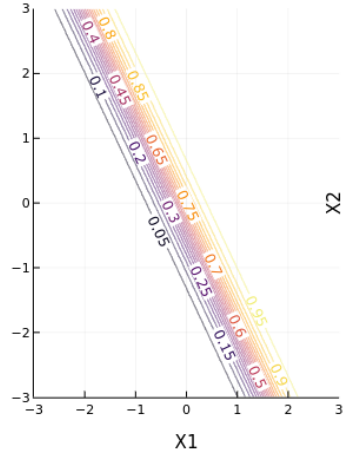
The corresponding **decision boundary** is linear:
$\hat{\theta}_0 + \hat{\theta}_1 x_1 + \cdots + \hat{\theta}_d x_d = 0$

Multiple Logistic Regression Example: p = 2

$\Pr(Y = \mathrm{A}|X)$ as 3D plot



$\Pr(Y = \mathrm{A}|X)$ as contour plot



samples and predictions

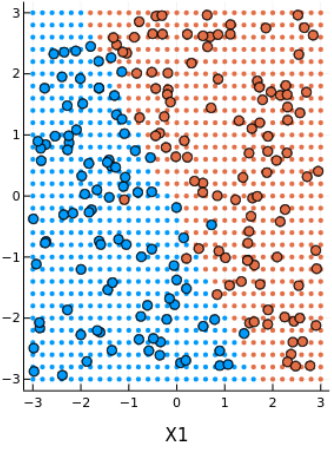
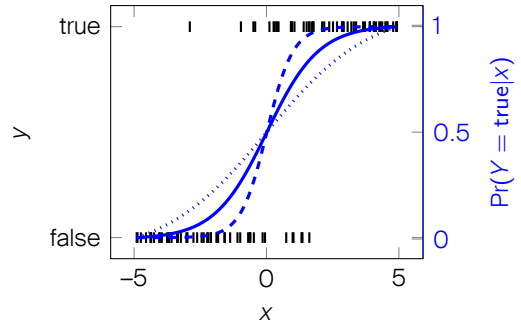


Table of Contents

Confusion Matrix



At decision threshold 0.5

		true class label		
		false	true	Total
predicted class label	false	42	4	46
	true	7	47	54
	Total	49	51	100

At decision threshold $\sigma(x) = 0.1$

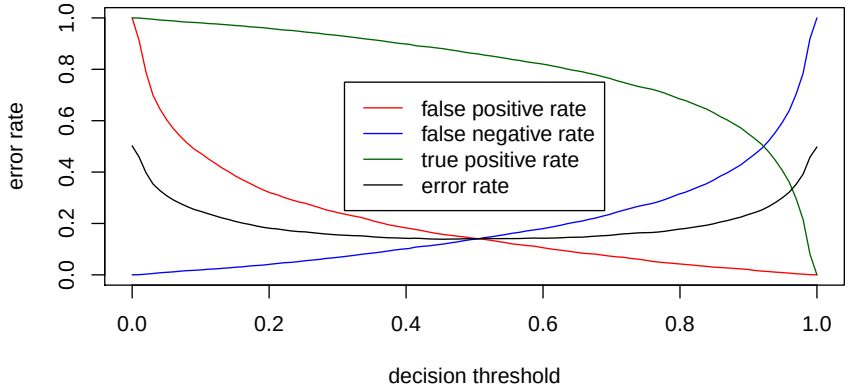
		true class label		
		false	true	Total
predicted class label	false	25	1	26
	true	24	50	74
	Total	49	51	100

Confusion Matrix & Error Rates

		true class label		
		Neg.	Pos.	Total
predicted class label	Neg.	True Neg. (TN)	False Neg. (FN)	N^*
	Pos.	False Pos. (FP)	True Pos. (TP)	P^*
	Total	N	P	

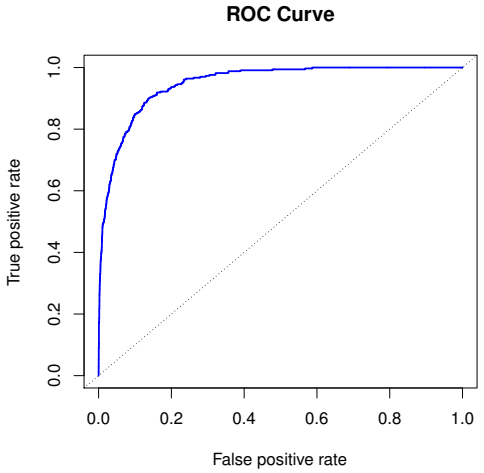
Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1-Specificity
True Pos. rate	TP/P	1-Type II error, Power, Sensitivity, Recall
False Neg. rate	FN/P	
Pos. Pred. value	TP/P^*	Precision, 1-false discovery, Proportion
Error Rate	$(FP+FN)/(P+N)$	Misclassification rate
Accuracy	1 - Error Rate	

Decision Thresholds and Error Rates



Finding the right threshold value depends on domain knowledge:
which error do we most care about?
E.g. disease detection: do we want a small false negative rate?

ROC curve and AUC



- measure True Pos. rate and False Pos. rate for different thresholds on test data to obtain the receiver operating characteristics **ROC** curve.
- Random classification would be on diagonal.
- Area under the ROC curve **AUC** assesses the classifier.
- Random classifier has AUC = 0.5, perfect classifier has AUC = 1.

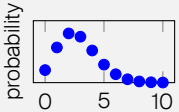
Quiz

1. Multiplying all parameters of logistic regression by a factor larger than 1 leaves the decision boundary (at decision threshold 0.5) unchanged.
2. If it is possible to perfectly classify the data, there exists a classifier with AUC = 1.
3. If we classify according to the worst classifier (class A if $p_A < 0.5$ and class B otherwise), the AUC is expected to be smaller than 0.5.
4. Typically we expect the AUC on the training set to be higher than on the test set.
5. No matter what classifier we use, the ROC curve always starts at (0, 0) and ends at (1, 1).

Table of Contents

Poisson Regression

Poisson



$$p(k|x) = \frac{e^{-f(x)} f(x)^k}{k!}$$

$f(x)$: a number

mean: $f(x)$

variance: $f(x)$

mode: $\lfloor f(x) \rfloor$ (floor)

When the response is a non-negative count variable, e.g. number of bicycles rented, it can be problematic to use the normal distribution to model the noise, because the support of the normal distribution is not restricted to positive numbers and the variance is independent of the mean.
The Poisson distribution can be better suited in this case (see bike sharing example in the notebook).

Take-home message

Always ask yourself: which distribution is best to model the noise.

Table of Contents

Where Does Noise Come From?

For most data generating processes we **cannot measure all factors** that determine the outcome.

$\Rightarrow$ **same values of the measured factors can cause different outcomes**.

- **MNIST** Different persons may label the same handwritten digit differently.
- **Spam** What is spam for somebody, may not be spam for someone else.
- **Weather** Even when all considered weather stations measure exactly the same values at time t_1 and t_2, the full state of the weather at t_1 differs most likely from the one at t_2.

In machine learning we treat the effect of unmeasured factors as noise with certain probability distributions.

Suggested Reading

- 4.1 An Overview of Classification
- 4.2 Why Not Linear Regression?
- 4.3 Logistic Regression
- 4.3.4 Multiple Logistic Regression
- 4.4.2 Linear Discriminant Analysis
  (mostly the part on confusion matrix, ROC, AUC).
- 4.6. Generalized Linear Models