Please scan this code to register your presence on Ilias for contact-tracing purposes. Your Ilias identity will be used. – Only do this if you are physically present!

Data Literacy
Lecture 07: Logistic regression and its friends

Jakob Macke

Eberhard Karls Universität Tübingen
Faculty of Science
Department of Computer Science
Machine Learning in Science

December 6, 2021

Please register here if you want to participate in the course evaluation.

Plan for today

Binary classification: Answering ‘yes/no’ questions

Logistic regression

Bayesian logistic regression

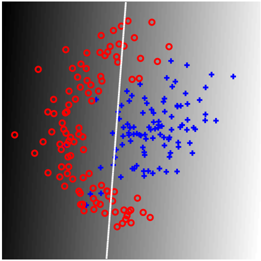
Interpreting the weights in (logistic) regression

Why stop here? Generalized linear models

Summary

Binary classification: Answering ‘yes/no’ questions

Binary Classification: Answering ‘yes/no’ questions



Examples:

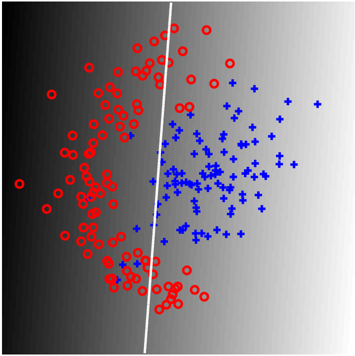
- Is there a face in this image?
- Based on this brain-scan, does this patient have a given disease or not?
- Will this customer buy this product or not?
- Will the home-team win this match?

Often we are interested in probabilities:

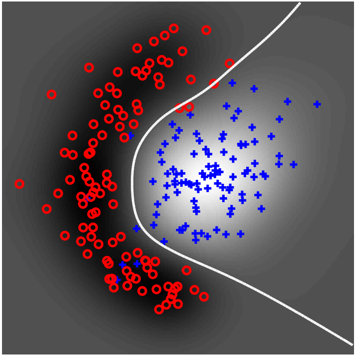
- How certain are you that this patient has the disease?
- How likely is it that the home-team win the match?

We focus on linear decision rules,

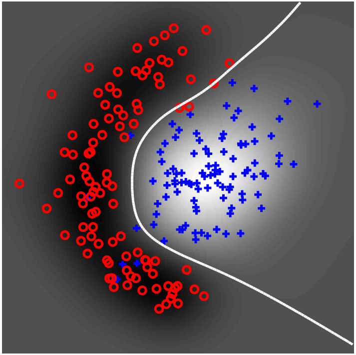
also known as ‘linear discriminant functions’.



We focus on linear decision rules,
also known as ‘linear discriminant functions’.



Of course, linear models can be used with nonlinear basis functions to solve nonlinear classification problems!



Linear discriminants separate the space by a hyperplane, and the parameters define its normal vector.

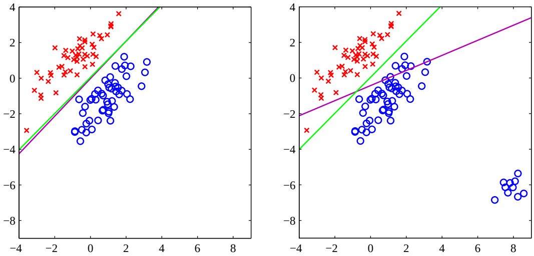
- Decision function: $z(\mathbf{x}) = \omega^\top \mathbf{x}$
- Classification:

$$\text{if } z(\mathbf{x}) > 0 \text{ say } \mathbf{x} \text{ belongs to class 1 (“yes”)} \quad (1)$$
$$\text{if } z(\mathbf{x}) < 0 \text{ say } \mathbf{x} \text{ belongs to class -1 (“’no”)} \quad (2)$$
$$(3)$$

- The decision-surface has equation $z(\mathbf{x}) = 0$, and is a hyperplane of dimensionality $D - 1$.
- ω is the normal vector to the plane, and points into the positive class.
- ω_o determines the location of the decision-surface
- $|z(\mathbf{x})|$ is proportional to the perpendicular distance to the decision-surface (with factor 1 if $||\omega|| = 1$).

Why not just use linear regression?

- We have to fit the function $z(\mathbf{x}) = \omega^\top \mathbf{x}$ to data.
- Simply do a linear regression from $\mathbf{x}$ to t by minimizing the sum-of-squared errors $\sum_n (z(\mathbf{x}_n) - t_n)^2$?
- $\omega_{lsq} = \left(\sum_n \mathbf{x}_n \mathbf{x}_n^\top\right)^{-1} \sum_n \mathbf{x}_n t_n$
- Q: Why is this a bad idea?



Bishop PRML Figure 4.4

Reminder: for (Gaussian) linear regression, we assumed Gaussian outputs.

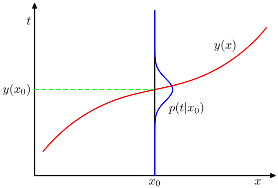
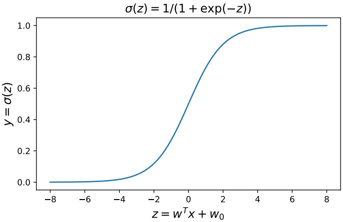


Figure Bishop PRLM 1.28

- For linear regression, we conditioned on $\mathbf{x}$, and assumed a Gaussian distribution over t: $t|x \sim \mathcal{N}(y(\mathbf{x}), \sigma^2)$
- We maximized the *conditional* log-likelihood $L(\omega) = \sum_n \log p(t_n|\mathbf{x}_n, \omega)$, i.e we assumed that the $\mathbf{x}$ were given.
- Aside: Note that we do not need to make any assumptions about $p(\mathbf{x})$! $\mathbf{x}$ can be high-dimensional, so it is difficult to make appropriate distributional assumptions for it.

We need a (conditional) distribution for binary outcomes!

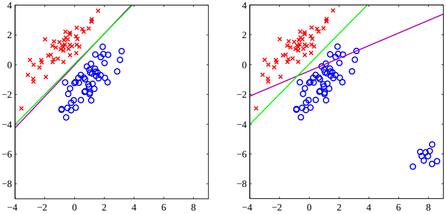
- Remember: Bernoulli distribution (‘coin flip’):
  $P(t = 1|p) = p$, $P(t = -1|p) = 1 - p$,...
- ... and we make this probability dependent on $z(x) = \omega^\top x$.
- We set $P(t = 1|x) = \sigma(z(x))$ where $\sigma(z) = 1/(1 + \exp(-z))$ is the logistic sigmoid function– this makes sure that the predicted probability is in $[0, 1]$.



- [Note/Homework]: This class-conditional density is exactly what we would get if we assume that data *within* each class is Gaussian.

Logistic regression

The model we just defined is logistic regression.



Bishop PRML Figure 4.4 a and b

- For maximum likelihood estimation, optimize log-likelihood numerically (no closed form solution like linear regression.)
- Typically: minimize the negative log-likelihood $\mathscr{L}$

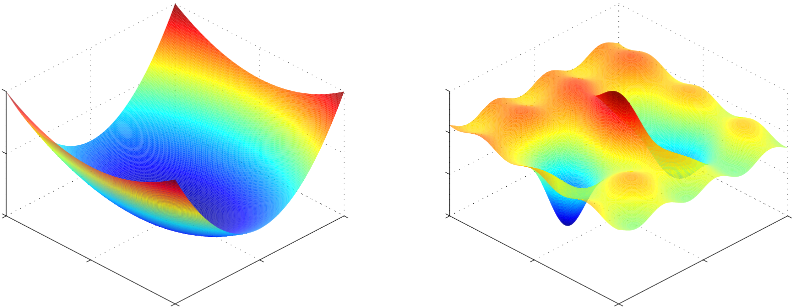
$$\mathscr{L} = -\sum_n s_n \log(y_n) + (s_n - 1)\log(1 - y_n) \tag{4}$$

where $y_n = \sigma(z_n)$, and we have introduced new parameters s_n

$$s_n : s_n = 1 \text{ if } t_n = 1, \text{ and } s_n = 0 \text{ otherwise.} \tag{5}$$

- For $s_n = 1$, only the $\log(y_n)$-term 'survives', if $s_n = 0$ only the $\log(1 - y_n)$ term does.

The cost-function for logistic regression is convex.



- Fact: The negative log-likelihood is convex.
- There are no local minima to get stuck in, and there are good optimization methods and theoretical results for convex problems.

Gradient descent is a simple method for numerically minimizing a function.

Treble Cone, Wanaka NZ, commons.wikimedia.org

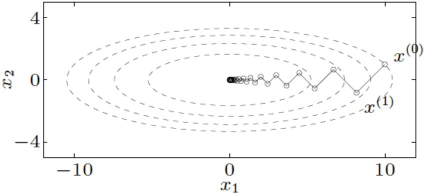


Figure from Stephen Boyd, Convex Optimization

But for logistic regression, more efficient approaches exist – toolboxes are your friend!

Bayesian logistic regression

Bayesian logistic regression I: Maximum a posteriori estimation

- Maximum Likelihood estimation (MLE): Maximize $\log P(D|\omega)$
- Maximum a posteriori estimation (MAP): Maximize $\log P(\omega|D) = \log P(D|\omega) + \log P(\omega) + \text{const.}$, where $P(\omega)$ is a prior distribution
- Good news: If the log-prior is convex (e.g. Gaussian), then this is still a convex ('easy') optimization problem.
- In fact, the 'logistic regression' function in sci-kit learn does MAP (not MLE) by default!

Bayesian logistic regression II: Full Bayesian inference

- Bayesian inference: Estimate full posterior distribution $P(\omega|D)$
- Why is Bayesian inference useful?
- Error bars, predictive uncertainty, model-selection, active learning
- Bad news: No closed-form solutions for posterior
- Good news: For logistic regression, the posterior will generally be *uni-modal*, and often look Gaussian-ish, so that Gaussian approximations can work well.
- How to find an approximation to the posterior?
  - Laplace approximation (simple and works well here!)
  - Variational inference
  - MCMC

Interpreting the weights in (logistic) regression

Interpreting weights in linear regression

- Recall our linear model $y = \omega_o + \omega_1 x_1 + \omega_2 x_2 ... + \omega_M x_M$
- In the model, moving x_i by δ_i (which keeping all other x_i fixed!) changes y by $\omega_i \delta_i$.
- $\omega_i = 0$: x_i does not have any predictive value (beyond what can be predicted from the other variables).
- ω_i big: Model outputs will be very sensitive to changes in x_i, suggesting that x_i is 'important'.

Example 1: Predicting weight (loosely inspired by Gelman, Hill, Vehtari 2020)

- Suppose we want to predict the *weight* of an (adult) person:

$$
\begin{aligned}
\text{[weight in pounds]} = {} & \omega_o + \omega_1 \text{[height in inches]} + && (6)\\
& + \omega_2 \text{[age in years]} && (7)\\
& + \omega_3 \text{[is person male, yes=1, no=0]} && (8)\\
& + \omega_4 \text{[advanced education, yes=1, no=0]} && (9)
\end{aligned}
$$

- ω_2 tells us "how much weight do people gain per year with age"
- ω_3 tells us "Everything else being equal, how much heavier is an (average) male person?"
- Observation 1: If we change the units to 'cm', then ω_1 will change– regression weights are always relative to the scaling of the inputs!
- Observation 2: Weight is inversely correlated with educational status, so ω_4 would likely be negative. But this is not to be interpreted causally– your weight will (likely) not drop on the day that you receive your Master's degree!

Interpreting weights in logistic regression

- In logistic regression, things are a bit more complicated: We have

$$P(t = 1|x) = 1/(1 + \exp(\omega_o + \omega_1 x_1 + \ldots \omega_M x_M) \tag{10}$$

- We can write

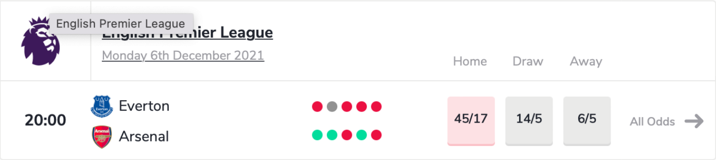
$$\frac{P(t = 1|x)}{1 - P(t = 1|x)} = [\ldots] = \exp(\omega_o + \omega_1 x_1 + \ldots \omega_M x_M), \tag{11}$$

$$\log \frac{P(t = 1|x)}{1 - P(t = 1|x)} = \omega_o + \omega_1 x_1 + \ldots \omega_M x_M. \tag{12}$$

- $\frac{P(t=1|x)}{1-P(t=1|x)}$ is called the odds of the event $t = 1$.
- So, changes in x linearly change the log-odds!

Odds are a bit odd

- A has a probability of 50%. The odds of A are $\frac{50\%}{50\%} = 1$.
- A has probability of $1/10 = 10\%$. The odds of A are $\frac{1/10}{9/10} = \frac{1}{9}$.
- It is tempting to interpret this as it occurring in '1 in 9 cases'. However, it means 'it will happen 1 time and it will not happen 9 times', i.e. in '1 in 10 cases'.
- UK Bookmakers will quote 'fractional odds' for bets:



,

- to be interpreted as 'If you bet 5 pounds on a draw, you can earn 14 pounds (and get your 5 stake back'. This will be a fair bet if the probability is $\frac{5}{14+5} \approx 26\%$.
- Odds-ratios are used extensively in medicine.

Odds are less odd for *rate* events

- Suppose we want to predict the risk of a rare disease $t = 1$. Then $P(t = -1|x) \approx 1$, so $\frac{P(t=1|x)}{P(t=-1d|x)} \approx P(t = 1|x)$. E.g.

$$
\begin{aligned}
P(t = 1|x) \approx \exp(\omega_o + & \quad (13)\\
& + \omega_1[\text{smoker: 1 or 0}]] \quad (14)\\
& + \omega_2[\text{obese: 1 or 0}] \quad (15)\\
& + \omega_3[\text{high blood pressure: 1 or 0}]) \quad (16)
\end{aligned}
$$

- One could interpret this as: If you do not have any of the three risk-factors, your risk is $\exp(\omega_o)$.
- If you are a smoker, your risk increases by a factor of $\exp(\omega_1)$, i.e. is $\exp(\omega_o) \times \exp(\omega_1)$, etc.

Why stop here? Generalized linear models

What if we want to predict *count*-observations?

- We have met linear regression
  (to predict Gaussian outcomes, mean is linear function of x)
  and logistic regression
  (to predict binary outcomes, mean is *sigmoid* of linear function of x)
- Suppose we want to predict count-valued observations, e.g.
  - Number of new covid-cases in a district
  - Number of action potentials a neuron will fire
  - Number of emails that will arrive in the next 5 minutes
- A natural choice is the Poisson distribution with mean μ,

$$P(t|\mu) = \frac{\mu^t \exp(-\mu)}{t!} \tag{17}$$

- We want the mean μ to be dependent on $z = \omega^\top x$. Can we just set $\mu = \omega^\top \mathbf{x}$?

What if we want to predict *count*-observations?

- We want the mean μ to be dependent on $z = \omega^\top \mathbf{x}$. Can we just set $\mu = \omega^\top \mathbf{x}$?
- Better: Set $\mu = \exp(\omega^\top \mathbf{x})$ to make sure mean is always positive.
- Poisson regression:

$$P(t|\mathbf{x}) = \frac{\exp(\omega^\top \mathbf{x})^t \exp(-\exp(\omega^\top \mathbf{x}))}{t!} \tag{18}$$

- Log-likelihood

$$L = t\omega^\top \mathbf{x} - \exp(\omega^\top \mathbf{x}) - \log t! \tag{19}$$

This can be generalized: Generalized linear models

- Recipe: 1) Write down a model for observations t from the exponential family, e.g.
- Gaussian, Exponential, Gamma, Poisson, Bernoulli, Binomial, Categorical, Multinomial, ...
- 2) Pick a suitable (nonlinear) function g to like the mean-parameter of the observation model with the input z: $E(t|z) = g(z(\mathbf{x}))$
- 3) Use this as a regression model!
- Terminology: g^{-1} is called the link-function, e.g
  - Gaussian: $g(z) = z$, link function $g^{-1}(y) = y$.
  - Bernoulli: $g(z) = \sigma(z)$, link function $g^{-1}(y) = \log(y/(1-y))$.
  - Poisson: $g(z) = \exp(z)$, link function $g^{-1}(y) = \log(y)$.
- For each exponential family model, there is an canonical link function– in the examples above, we used the canonical link functions.

Summary

Summary

- Logistic regression can be used to predict binary ('yes/no') outcomes.
- The cost-functions of logistic regression (both MLE and MAP) can be efficiently optimized via convex optimization problems.
- Interpretation of weights of linear regression: Linear influence on output, but be careful when interpreting them causally!
- Interpretation of weights of logistic regression: Linear influence on *odds* of output-event.
- Generalized linear models: General observation models, in which the mean is a nonlinear function of a linear function of the inputs.