LOGISTIC REGRESSION

Sergey Nikolenko

Harbour Space University, Barcelona, Spain
March 17, 2017

- We have already considered the logistic sigmoid:

$$p(\mathcal{C}_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+e^{-a}} = \sigma(a),$$

$$\text{where } a = \ln \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)}, \qquad \sigma(a) = \frac{1}{1+e^{-a}}.$$

- We have derived LDA and QDA, and trained them with maximal likelihood.

- Let's go back to classification.
- Two classes, the posterior is the logistic sigmoid of a linear function:

$$p(\mathcal{C}_1 \mid \phi) = y(\phi) = \sigma(\mathbf{w}^\top \phi), \quad p(\mathcal{C}_2 \mid \phi) = 1 - p(\mathcal{C}_1 \mid \phi).$$

- *Logistic regression* is when we optimize $\mathbf{w}$ directly.

- For a dataset $\{\phi_n, t_n\}$, $t_n \in \{0, 1\}$, $\phi_n = \phi(\mathbf{x}_n)$:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n}, \quad y_n = p(\mathcal{C}_1 \mid \phi_n).$$

- We look for maximal likelihood parameters by minimizing $-\ln p(\mathbf{t} \mid \mathbf{w})$:

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\right].$$

TWO CLASSES

- Since $\sigma' = \sigma(1-\sigma)$, we take the gradient:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n.$$

- If we now perform gradient descent, we get the separating surface.
- Note that if the data are actually separable, we could get heavy overfitting: $\|\mathbf{w}\| \to \infty$, and the sigmoid turns into a Heaviside function.
- We have to regularize.

- Logistic regression does not yield a closed form solution because of the sigmoid.
- But function $E(\mathbf{w})$ is convex, and we can use Newton–Raphson's method: use local quadratic approximation to the loss function on each step:

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \mathbf{H}^{-1} \nabla E(\mathbf{w}),$$

  where $\mathbf{H}$ (Hessian) is the matrix of second derivatives for $E(\mathbf{w})$.

- Aside: let us apply Newton–Raphson's method to regular linear regression with quadratic error:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \left(\mathbf{w}^\top \phi_n - t_n\right) \phi_n = \Phi^\top \Phi \mathbf{w} - \Phi^\top \mathbf{t},$$
$$\nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \phi_n \phi_n^\top = \Phi^\top \Phi,$$

and the optimization step will be

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \left(\Phi^\top \Phi\right)^{-1} \left[\Phi^\top \Phi \mathbf{w}^{\text{old}} - \Phi^\top \mathbf{t}\right] =$$
$$= \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{t},$$

i.e., we get a solution in one step.

- For logistic regression:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\,\phi_n = \Phi^\top (\mathbf{y} - \mathbf{t})\,,$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^\top = \Phi^\top R\Phi$$

for a diagonal matrix R c $R_{nn} = y_n(1-y_n)$.

- Optimization step formula:

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \left(\Phi^\top R \Phi\right)^{-1} \Phi^\top (\mathbf{y} - \mathbf{t}) = \left(\Phi^\top R \Phi\right)^{-1} \Phi^\top R \mathbf{z},$$

  where $\mathbf{z} = \Phi \mathbf{w}^{\text{old}} - R^{-1} (\mathbf{y} - \mathbf{t})$.
- This is like a weighted least squares optimization problem with matrix of weights R.
- Hence the title: iterative reweighted least squares (IRLS).

- In case of several classes

$$p(\mathcal{C}_k \mid \phi) = y_k(\phi) = \frac{e^{a_k}}{\sum_j e^{a_j}} \text{ for } a_k = \mathbf{w}_k^\top \phi.$$

- Consider the ML estimate again; first,

$$\frac{\partial y_k}{\partial a_j} = y_k \left([k = j] - y_j \right).$$

SEVERAL CLASSES

- Let us now write the likelihood: for a 1-of-K coding scheme we have target vector $\mathbf{t}_n$ and likelihood

$$p(\mathbf{T} \mid \mathbf{w}_1, \dots, \mathbf{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathcal{C}_k \mid \phi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

for $y_{nk} = y_k(\phi_n)$; taking the log, we get

$$E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\ln p(\mathbf{T} \mid \mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk},$$ И

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^{N} \left(y_{nj} - t_{nj}\right) \phi_n.$$

- Again, we can optimize with Newton–Raphson's method; the Hessian is

$$\nabla_{\mathbf{w}_k} \nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^{N} y_{nk} \left([k = j] - y_{nj} \right) \phi_n \phi_n^\top.$$

- What if we have a different sigmoid?
- The same setting: two classes, $p(t = 1 \mid a) = f(a)$, $a = \mathbf{w}^\top \phi$, f is the activation function.
- Consider an activation function with threshold θ: for each ϕ_n we compute $a_n = \mathbf{w}^\top \phi_n$, and

$$\begin{cases} t_n = 1, & \text{if } a_n \geq \theta, \\ t_n = 0, & \text{if } a_n < \theta. \end{cases}$$

- If θ is taken by distribution $p(\theta)$, this corresponds to

$$f(a) = \int_{-\infty}^{a} p(\theta) \mathrm{d}\theta.$$

- Suppose, e.g., that $p(\theta)$ is a Gaussian with zero mean and unit variance. Then

$$f(a) = \Phi(a) = \int_{-\infty}^{a} \mathcal{N}\left(\theta \mid 0, 1\right) \mathrm{d}\theta.$$

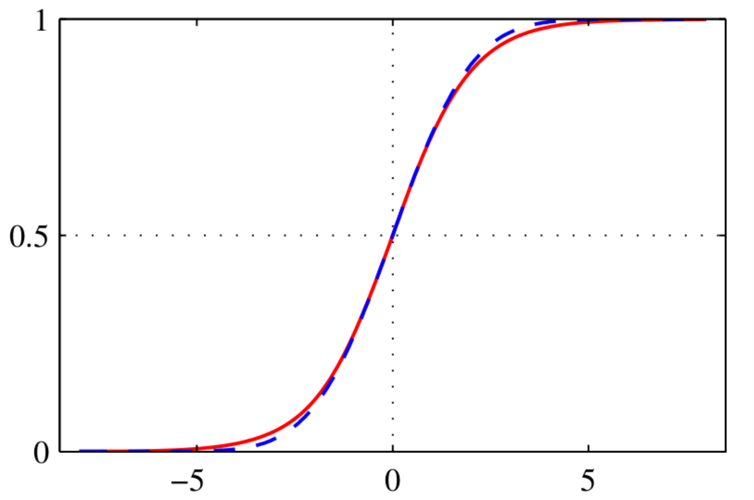
- This is called the *probit function*; it's non-elementary, related to

$$\operatorname{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-\frac{\theta^2}{2}} \mathrm{d}\theta :$$

$$\Phi(a) = \frac{1}{2}\left[1 + \frac{1}{\sqrt{2}}\operatorname{erf}(a)\right].$$

- Probit regrassion is the model with probit activation function.

σ и Φ



LAPLACE APPROXIMATION AND BAYESIAN LOGISTIC REGRESSION

- An aside: how do we approximate a complex distribution with a simpler one?
- E.g., how do we approximate a distribution near its maximum with a Gaussian? (a very natural idea)
- Let's first consider the distribution of a single continuous variable $p(z) = \frac{1}{Z} f(z)$.

- Step 1: find the maximum z_0.
- Step 2: decompose into Taylor series

$$\ln f(z) \approx \ln f(z_0) - \frac{1}{2}A(z - z_0)^2, \text{ where } A = -\frac{d^2}{dz^2}\ln f(z)\mid_{z=z_0}.$$

- Step 3: approximate

$$f(z) \approx f(z_0)e^{-\frac{A}{2}(z-z_0)^2},$$

and it will be a Gaussian after normalization.

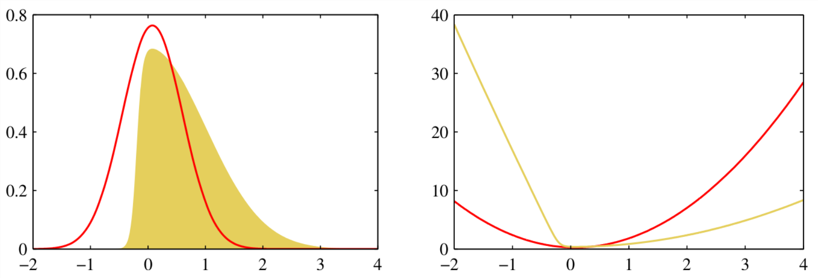
- This can be generalized to the multidimensional case $p(\mathbf{z}) = \frac{1}{Z} f(\mathbf{z})$:

$$f(\mathbf{z}) \approx f(\mathbf{z}_0) e^{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^\top \mathbf{A}(\mathbf{z}-\mathbf{z}_0)},$$

$$\text{where } \mathbf{A} = -\nabla\nabla \ln f(\mathbf{z}) \mid_{z=z_0} .$$

Exercise. What is the normalizing constant here?

LAPLACE APPROXIMATION



- Having understood Laplace approximation, let us apply it first to model selection.
- To compare models from $\{\mathcal{M}_i\}_{i=1}^{L}$, by the test set D we estimate the posterior

$$p(\mathcal{M}_i \mid D) \propto p(\mathcal{M}_i)p(D \mid \mathcal{M}_i).$$

- If a model is defined parametrically, we get $p(D \mid \mathcal{M}_i) = \int p(D \mid \theta, \mathcal{M}_i)p(\theta \mid \mathcal{M}_i)d\theta$.
- This is the probability to generate D if we choose model parameters according to its prior; the denominator from Bayes' theorem:

$$p(\theta \mid \mathcal{M}_i, D) = \frac{p(D \mid \theta, \mathcal{M}_i)p(\theta \mid \mathcal{M}_i)}{p(D \mid \mathcal{M}_i)}.$$

MODEL COMPARISON WITH LAPLACE APPROXIMATION

- Earlier we approximated it with a nearly piecewise constant function.
- Let us now approximate with a Gaussian; integrating, we get

$$Z = \int f(\mathbf{z}) d\mathbf{z} \approx \int f(\mathbf{z}_0) e^{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^\top \mathbf{A}(\mathbf{z}-\mathbf{z}_0)} d\mathbf{z} = f(\mathbf{z}_0) \frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}}.$$

- And we have $Z = p(D)$, $f(\theta) = p(D \mid \theta) p(\theta)$.

- We get

$$\ln p(D) \approx \ln p(D \mid \theta_{\mathrm{MAP}}) + \ln P(\theta_{\mathrm{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|.$$

- $\ln P(\theta_{\mathrm{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|$ is called the *Occam's factor*.
- $\mathbf{A} = -\nabla\nabla \ln p(D \mid \theta_{\mathrm{MAP}}) p(\theta_{\mathrm{MAP}}) = -\nabla\nabla \ln p(\theta_{\mathrm{MAP}} \mid D)$.

- We get

$$\ln p(D) \approx \ln p(D \mid \theta_{\text{MAP}}) + \ln P(\theta_{\text{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|.$$

- If the Gaussian prior $p(\theta)$ is wide enough, and $\mathbf{A}$ has full rank, we can roughly approximate (prove it!) as

$$\ln p(D) \approx \ln p(D \mid \theta_{\text{MAP}}) - \frac{1}{2} M \ln N,$$

  where M is the number of parameters, N is the number of points in D, and we have omitted additive constants.
- This is called the *Bayesian information criterion* (BIC), or *Schwarz criterion*.

- And now the full Bayesian treatment.
- Logistic regression is not as simple as linear regression: we can’t get an exact answer out of a product of logistic sigmoids.
- We’ll make a Laplace approximation.

- Gaussian prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mu_0, \Sigma_0).$$

- The posterior is then

$$\begin{aligned} p(\mathbf{w} \mid \mathbf{t}) &\propto p(\mathbf{w}) p(\mathbf{t} \mid \mathbf{w}), \text{ и} \\ \ln p(\mathbf{w} \mid \mathbf{t}) &= -\frac{1}{2}\left(\mathbf{w} - \mu_0\right)^\top \Sigma_0^{-1}\left(\mathbf{w} - \mu_0\right) \\ &\quad + \sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\right] + \text{const}, \\ \text{where } y_n &= \sigma(\mathbf{w}^\top \phi_n). \end{aligned}$$

BAYESIASN LOGISTIC REGRESSION

- To approximate, we first find the maximum $\mathbf{w}_{\text{MAP}}$, and then the covariance matrix is the matrix of second derivatives

$$\Sigma_N = -\nabla\nabla \ln p(\mathbf{w} \mid \mathbf{t}) = \Sigma_0^{-1} + \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^\top.$$

- Our approximation is now

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{w}_{\text{MAP}}, \Sigma_N).$$

- And we can now get the Bayesian prediction:

$$p(\mathcal{C}_1 \mid \phi, \mathbf{t}) = \int p(\mathcal{C}_1 \mid \phi, \mathbf{w}) p(\mathbf{w} \mid \mathbf{t}) d\mathbf{w} \approx \int \sigma(\mathbf{w}^\top \phi) q(\mathbf{w}) \mathrm{d}\mathbf{w}.$$

- Note that $\sigma(\mathbf{w}^\top \phi)$ depends on $\mathbf{w}$ only via its projection on ϕ.
- We denote $a = \mathbf{w}^\top \phi$:

$$\sigma(\mathbf{w}^\top \phi) = \int \delta(a - \mathbf{w}^\top \phi) \sigma(a) \mathrm{d}a.$$

- $\sigma(\mathbf{w}^\top \phi) = \int \delta(a - \mathbf{w}^\top \phi)\sigma(a)\mathrm{d}a$, and therefore

$$\int \sigma(\mathbf{w}^\top \phi) q(\mathbf{w}) d\mathbf{w} = \int \sigma(a) p(a) \mathrm{d}a,$$

$$\text{where } p(a) = \int \delta(a - \mathbf{w}^\top \phi) q(\mathbf{w}) \mathrm{d}\mathbf{w}.$$

- $p(a)$ is the marginalization of Gaussian $q(\mathbf{w})$, where we integrate over everything which is orthogonal to ϕ.

- $p(a)$ is the marginalization of Gaussian $q(\mathbf{w})$, where we integrate over everything which is orthogonal to ϕ.
- Hence, $p(a)$ is a Gaussian too, and we can find its parameters

$$\begin{aligned}\mu_a =& \mathrm{E}[a] = \int a p(a) \mathrm{d}a = \int q(\mathbf{w}) \mathbf{w}^\top \phi \mathrm{d}\mathbf{w} = \mathbf{w}_{\mathrm{MAP}}^\top \phi, \\ \sigma_a^2 =& \int \left(a^2 - \mathrm{E}[a]\right)^2 p(a) \mathrm{d}a = \\ =& \int q(\mathbf{w}) \left[(\mathbf{w}^\top \phi)^2 - (\mu_N^\top \phi)^2\right]^2 \mathrm{d}\mathbf{w} = \phi^\top \Sigma_N \phi.\end{aligned}$$

- Thus, we get that

$$p(\mathcal{C}_1 \mid \mathbf{t}) = \int \sigma(a) p(a) \mathrm{d}a = \int \sigma(a) \mathcal{N}(a \mid \mu_a, \sigma_a^2) \mathrm{d}a.$$

- $p(\mathcal{C}_1 \mid \mathbf{t}) = \int \sigma(a)\mathcal{N}(a \mid \mu_a, \sigma_a^2)\mathrm{d}a.$
- This integral is not easy to take, because sigmoid is hard, but we can approximate it by approximating $\sigma(a)$ with the probit: $\sigma(a) \approx \Phi(\lambda a)$ for $\lambda = \sqrt{\pi/8}$.

Exercise. Prove that $\lambda = \sqrt{\pi/8}$ y σ and Φ have the same slope at zero.

- And if we pass to the probit function, its convolution with a Gaussian will be another probit:

$$\int \Phi(\lambda a)\mathcal{N}(a \mid \mu, \sigma^2)\mathrm{d}a = \Phi\left(\frac{\mu}{\sqrt{\frac{1}{\lambda^2} + \sigma^2}}\right).$$

Exercise. Prove it.

- As a result, we get the approximation

$$\int \sigma(a)\mathcal{N}(a \mid \mu, \sigma^2)\mathrm{d}a \approx \sigma\left(\kappa(\sigma^2)\mu\right),$$

$$\text{where } \kappa(\sigma^2) = \frac{1}{\sqrt{1+\frac{\pi}{8}\sigma^2}}.$$

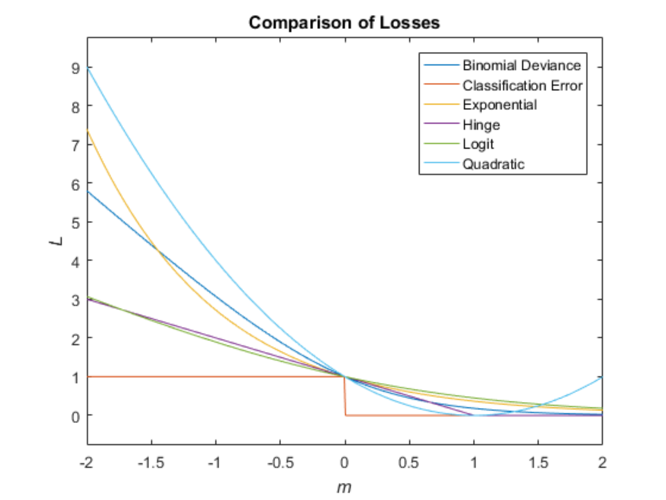
- And now, putting it all together, we get the predictive distribution:

$$
\begin{aligned}
p(\mathcal{C}_1 \mid \phi, \mathbf{t}) =& \sigma\left(\kappa(\sigma_a^2)\mu_a\right), \text{ where} \\
\mu_a =& \mathbf{w}_{\mathrm{MAP}}^\top \phi, \\
\sigma_a^2 =& \phi^\top \Sigma_N \phi, \\
\kappa(\sigma^2) =& \frac{1}{\sqrt{1 + \frac{\pi}{8}\sigma^2}}.
\end{aligned}
$$

- By the way, the separating hyperplane $p(\mathcal{C}_1 \mid \phi, \mathbf{t}) = \frac{1}{2}$ is defined by equation $\mu_a = 0$, and it's the same as just using $\mathbf{w}_{\mathrm{MAP}}$.
- The difference is important only for more complex criteria.

LOSS FUNCTIONS IN CLASSIFICATION

- And a different look at classification: different methods differ by which loss function they optimize.
- Classification has a problem with the “correct” error function, i.e., misclassification rate:
  - it’s not differentiable everywhere,
  - and its derivative is useless.
- Let us look at different loss functions; we have seen several of them, but there are lots more.


Comparison of Losses
Binomial Deviance
Classification Error
Exponential
Hinge
Logit
Quadratic
L
m
9
8
7
6
5
4
3
2
1
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

THANK YOU!

Thank you for your attention!