Logistic Regression

Gautam Kamath

Intuition

- "Predictions with confidence"
  - Binary classification, but also gives a "confidence" for correctness
  - Today: use 0 and 1 as labels, instead of ± 1
- (Draw hours studied vs passed class example)
- (Draw perceptron example, sharp threshold, boundary examples)
- Bernoulli model: parameterize probability of label y by feature vector x, parameter vector w
  - $\Pr[y = 1 \mid x, w] = p(x, w) \in [0,1]$
  - $\Pr[y = 0 \mid x, w] = 1 - p(x, w)$

How to parameterize?

- $\Pr[y = 1 \mid x, w] = p(x, w) \in [0,1]$. How do we define $p(x, w)$?
- Take 1: $p(x, w) = \langle x, w \rangle$
  - Being far on the positive side of the hyperplane makes it large, vice versa
  - Why doesn't it work? LHS is in $[0,1]$, while RHS is over $\mathbf{R}$
- Take 2: $\log\left(\frac{p(x,w)}{1-p(x,w)}\right) = \langle x, w \rangle$ ("logit transform")
  - Sanity check: $p(x, w)$ ranging over $[0,1]$ causes LHS to range over $\mathbf{R}$ (like RHS)

Rearranging the parameterization

$$\log\left(\frac{p(x,w)}{1-p(x,w)}\right) = \langle x, w\rangle$$

$$\frac{p(x,w)}{1-p(x,w)} = \exp(\langle x, w\rangle)$$ (LHS: "odds ratio")

$$p(x,w) = \exp(\langle x, w\rangle)\left(1 - p(x,w)\right)$$

$$p(x,w) = \exp(\langle x, w\rangle) - \exp(\langle x, w\rangle)\, p(x,w)$$

$$p(x,w)(1 + \exp(\langle x, w\rangle)) = \exp(\langle x, w\rangle)$$

$$p(x,w) = \frac{\exp(\langle x, w\rangle)}{1 + \exp(\langle x, w\rangle)}$$

$$p(x,w) = \frac{1}{1 + \exp(-\langle x, w\rangle)} \triangleq \mathrm{sigmoid}(\langle x, w\rangle)$$

(draw sigmoid)

Visualizing $p(x, w)$

- $p(x, w) = \frac{1}{1+\exp(-\langle x,w \rangle)} \triangleq \mathrm{sigmoid}(\langle x, w \rangle)$
- (Draw linear separator, point on line $\langle x, w \rangle = 0$, point on either side)
- (Draw picture from the side of sigmoid)
- Why sigmoid? Admittedly a bit arbitrary
- Any monotone function from $\mathbf{R} \to [0,1]$ works
- E.g., take $\mathrm{sign}(\langle x, w \rangle)$ and you recover perceptron

Predicting using $p(x, w)$

- $p(x, w) = \frac{1}{1+\exp(-\langle x,w \rangle)}$
- If $p(x, w) > \frac{1}{2}$, predict $\hat{y} = 1$. Otherwise, predict $\hat{y} = 0$.
- Note this corresponds to $\hat{y} = \mathrm{sign}(\langle x, w \rangle)$, same as perceptron!
  - Difference 1: While we use same predictions, we optimize different functions
    - E.g., this can handle non linearly separable, couldn't with perceptron
  - Difference 2: Magnitude of $p(x, w)$ indicates *confidence* in prediction
    - Hence why we call it *regression* – we're learning confidences, which imply predictions

Deriving the MLE parameter vector

$$\hat{w} = \arg\max_w \prod_{i=1}^{n} \Pr[(x_i, y_i)|w]$$

$$= \arg\max_w \prod_{i=1}^{n} p(x_i, w)^{y_i}\big(1 - p(x_i, w)\big)^{1-y_i}$$

(Let $p_i = p_i(x_i, w)$)

$$= \arg\max_w \log \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}$$

$$= \arg\max_w \sum_{i=1}^{n} \log\big(p_i^{y_i}(1 - p_i)^{1-y_i}\big)$$

$$= \arg\max_w \sum_{i=1}^{n} y_i \log p_i + (1 - y_i)\log(1 - p_i)$$

("cross entropy loss")

Recall: $p(x, w) = \frac{1}{1+\exp(-\langle x,w\rangle)}$.

Suppose some $y_i = 1$. Then

$$y_i \log p_i + (1 - y_i)\log(1 - p_i) = \log p_i$$
$$= \log(1 + \exp(-\langle x, w\rangle))^{-1}$$
$$= -\log(1 + \exp(-\langle x, w\rangle))$$

Similarly, if $y = 0$, then

$$y_i \log p_i + (1 - y_i)\log(1 - p_i) = \log(1 - p_i)$$
$$= \log\left(\frac{\exp(-\langle x, w\rangle)}{1 + \exp(-\langle x, w\rangle)}\right)$$
$$= -\log(1 + \exp(\langle x, w\rangle))$$

Deriving the MLE parameter vector

$$\hat{w} = \arg\max_{w} \sum_{i=1}^{n} y_i \log p_i + (1 - y_i) \log(1 - p_i)$$

If $y_i = 1$, argument is $-\log(1 + \exp(-\langle x, w \rangle))$

If $y_i = 0$, argument is $-\log(1 + \exp(\langle x, w \rangle))$

$$\hat{w} = \arg\max_{w} \sum_{i=1}^{n} -\log\big(\exp(-y_i \langle x_i, w \rangle) + \exp\big((1 - y_i)\langle x, w \rangle\big)\big)$$

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} \log\big(\exp(-y_i \langle x_i, w \rangle) + \exp\big((1 - y_i)\langle x, w \rangle\big)\big)$$

An equivalent formulation

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} \log\Big(\exp(-y_i \langle x_i, w\rangle) + \exp\big((1 - y_i)\langle x, w\rangle\big)\Big)$$

Let $\tilde{y}_i = +1$ if $y_i = 1$, and $\tilde{y} = -1$ if $y_i = 0$.

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-\tilde{y}_i \langle x_i, w\rangle))$$

A step back: Optimization

- Letting

$$\text{loss } \ell_w(x_i, y_i) = y_i \log p_i + (1 - y_i) \log(1 - p_i),$$

Goal is to compute

$$\widehat{w} = \arg\min_w \frac{1}{n} \sum_{i=1}^{n} \ell_w(x_i, y_i)$$

- Claim 1: $\ell_w(x_i, y_i)$ is convex
  - Thus only need to find point where gradient $\frac{1}{n} \sum_{i=1}^{n} \nabla \ell_w(x_i, y_i)$ is 0
- Claim 2: $\nabla_w \ell_w(x_i, y_i) = (p_i(x_i, w) - y_i) x_i$
  - No closed form solution to set it equal to 0… (cf. linear regression)

Optimization methods

- (Draw iterative method picture for 1D, multiple D)
- Initialize w_0
- For $t = 1, 2, \ldots$
  - Choose direction d_t and step size η_t
  - $w_t = w_{t-1} - \eta_t d_t$
- How to pick step size η_t?
  - Constant, decaying (e.g., $1/\sqrt{t}$), "adaptively"
- How to pick direction d_t?

How to pick direction d_t?

- Gradient Descent
  - $d_t = \frac{1}{n}\sum_{i=1}^{n} \nabla_w \ell_{w_{t-1}}(x_i, y_i)$ (note, $= 0$ at optimum)
  - Running time?
- Stochastic gradient descent
  - Draw random set $B \subseteq [n]$, then let $d_t = \frac{1}{|B|}\sum_{i \in B} \nabla_w \ell_{w_{t-1}}(x_i, y_i)$
- Newton's Method
  - $d_t = \left(\frac{1}{n}\sum_{i=1}^{n} \nabla_w^2 \ell_{w_{t-1}}(x_i, y_i)\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla_w \ell_{w_{t-1}}(x_i, y_i)\right)$
  - Often needs fewer steps to converge, but more time/memory per step

Multiclass Logistic Regression

$$\Pr[y = k \mid x, w] = \frac{\exp(\langle w_k, x \rangle)}{\sum_{\ell=1}^{c} \exp(\langle w_\ell, x \rangle)}$$