UNIVERSITY OF CALIFORNIA, BERKELEY

Lecture 11: newton's method, logistic regression, and LDA

Oscar Ortega

July 16, 2021

1 ISL, SECTIONS 4.4.3, 7.1, 9.3.3; ESL, SECTION 4.4.1

2 ISL 4.4.3: LINEAR DISCRIMINANT ANALYSIS FOR MULTIVARIATE PREDICTORS

Assumptions:
We will now assume that $X = (x_1, x_2, ..., x_p)$ is drawn from a **multi-variate Gaussian** distribution, with a class-specific mean vector and a common co-variance matrix.
Definition:

$$\textbf{Covariance matrix}: \mathbf{E}((X - \mathbf{E}(X))((X - \mathbf{E}(X))^T)$$

$$Cov_{i,j} = cov(x_i, x_j)$$

This matrix is symmetric and is also positive semi-definite.

Multivariate Gaussian Density:

$$\frac{1}{(2\pi)^{p/2}}|\Sigma|^{1/2}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Because for every class the co-variance matrix will be the same we can reduce prediction to the class that maximizes this quantity, in other words our rule is the following:

$$\text{class}(x)^* = \text{argmax}_x \gamma_k(x) = (x-\mu)^T\Sigma^{-1}(x-\mu) - \frac{1}{2}\ln(|\Sigma|) + \ln(\pi_k): \text{ for } k \in \{1, 2, ..., k\}$$

We can consider this as maximizing the class conditional probability of x times the probability that x is a member of class c, $p(X|\text{class c})p(class)$

3 7.1: Polynomial Regression

In our original formulation of least squares we are are trying to estimate the following:

$$y_i = b_0 + b_1 x_i + \epsilon_i$$

naturally we can extend this to the following formulation:

$$y_i = \sum_{i=1}^{d} b_i x^i$$

a d dimensional polynomial.
Note that the function is linear in terms of the weights we are trying to estimate. so we can find the optimal function by performing least squares.

Note: The more terms in our our predictor, the more likely we are to overestimate the complexity of functions that are much more likely to be simple.
Definition:
variance of fit: if we let C the co-variance matrix of the weights, and

$$l_0^T = \sum_{i=0}^{i=4} x^i$$

then

$$\text{Var}\hat{(}f)(x_0) = l_0^T C l_0$$

4 ISL 9.3.3: An application Heart Disease Data

5 4.4.1: Fitting Logistic Regression Models

Logistic regression models are usually fit by Maximum Likelihood of G given X:

$$\mathscr{L}(\theta) = \sum_{i=1}^{N} log(P(G = k|X = x_i;\theta)$$

We ca simplify the notation in the two class case by considering a 0-1 response variable $y_i = 1$ when $g_i = 1$ and $y_i = 0$ when $g_i = 2$:

$$= \sum_{i=1}^{n} y_i log P(x_i;\theta) + (1 - y_i log(1 - p(x_i;\theta))$$

We can maximize by setting taking derivatives and setting equal to 0.

$$\frac{\partial \mathscr{L}(B)}{\partial B} = \sum_{i=1}^{N} x_i(y_i - p(x_i;B)) = 0$$

These are known as **score equations**
We can solve for the equations in the line above, by performing the Newton-Raphson algorithm which requires the Hessian matrix.

$$\frac{\partial^2 \mathscr{L}(B)}{\partial B \partial B^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i;B)(1 - p(x_i : B))$$

We can then perform a **newton update** as follows:

$$\mathscr{B}^{\text{new}} = \mathscr{B}^{\text{0ld}} - H^{-1}\frac{\partial \mathscr{L}(B)}{\partial B}$$

Where we define the H as the Hessian matrix, defined in the previous equation. Compactly we can describe the score and the hessian of the log-likelihood as $X^T(y-p)$ and $-X^T W X$, where $X \in \mathbf{R}^{n,p+1}$ of x_i values, p is a vector of probabilities $p_i = p(x_i; B^{\text{old}})$ and $W \in \mathbf{R}^{n,n}$ of diagonal weights.

$$\begin{aligned} B_{\text{new}} \rightarrow &= B_{\text{old}} + (X^T W X)^{-1} X^T (y-p) \\ \rightarrow &(X^T W X)^{-1} X^T W (X B_{\text{old}} + W^{-1}(y-p) \\ &= (X^T W X)^{-1} X^T W z \end{aligned}$$

where $z = X B_{\text{old}} + W^{-1}(y-p)$
We denote z as the **adjusted response** and because at each iteration, P, W and z change, this algorithm is known as **iteratively reweighed least squares** or IRLS, since each iteration solves the weighted least squares problem below.

$$B_{\text{new}} = \text{argmin}_B (z - XB)^T W (z - XB)$$

6 Lecture

6.1 Least-Squares polynomial Regression

Replace each x_i with feature vector $\Phi(x_i)$ with all terms of degrees $0, ..., p$

$$\Phi(x_i) = \begin{bmatrix} x_{i1}^2 & x_{i1}x_{i2} & x_{i1} & x_{i2} & 1 \end{bmatrix}^T$$

We can also use non-polynomial features (e.g edge detectors). Otherwise just like least squares.
Log. reg + quadratic features = same logistic posteriors as QDA. Very easy to overfit!
interpolating versus extrapolating data
interpolating data is when you generate a polynomial that best fits a set of data, and extrapolation is when one generates a polynomial that fits points that might not necessarily occur.

6.2 Weighted Least-Squares Regression

We assign weights to the set of points so that some are **more valuable** than others.

- Assign each sample pt a weight w_i: collect them in $\omega \in \mathbf{R}^{n,n}$
- Greater $w_i \rightarrow$ work harder to minimize $\|\hat{y}_i - y_i\|_{22}$
- recall $\hat{y} = Xw$
- Find w that minimizes $(Xw-y)^T\Omega(Xw-y) = \sum_{i=1}^{n} w_i(X_i^T w - y_i)^1$
- we can solve for w in the normal equations.

Note: $\Omega^{1/2}\hat{y}$ is the pt nearest $\Omega^{1/2}y$ on sub-spaces spanned by the columns of $\Omega^{1/2}x$

6.3 Newton's Method

Iterative optimization method for smooth $J(w)$ Often much faster than gradient descent.
Idea: You're a pt. v. Approximate J(w) near v by a quadratic function. From there we jump to its unique critical pt. and we repeat until convergence.

Taylors series about v: $\nabla J(w) = \nabla J(v) + \nabla^2 J(v)(w-v) + O(\|w-v\|^2)$
Where the squared term is the **Hessian** of J at v.
Find critical pt. w by setting $\nabla J(w) = 0$, because it finds the unique minima of the quadratic form.

$$w = v - (\nabla^2 J(w))^{-1}\nabla J(v)$$

Newtons method:

1. pick starting point w
2. repeat under convergence
3. $e \leftarrow$ solution to linear system $(H(w)e = -\nabla J(w)$
4. $w \leftarrow w + e$-+

Warning: The function doesn't know difference between minima, maxima, saddle points. Also note, that if the function is a quadratic, we will be able to converge in one iteration. Also note that newton's method will tend not to work well in cases that the function is not smooth.

6.4 Logistic Regression (cont'd)

Recall:

$$s'(\gamma) = s(\gamma)(1 - s(\gamma), s_i = s(X_i^T w)$$

$$\nabla J = -\sum_{i=1}^{n} (y_i - s_i) X_i = -X^T (y - s)$$

$$H(w) = \sum_{i=1}^{n} s_i (1 - s_i) X_i X_i^T = X^T \Omega X$$

$$\Omega = \text{diag}(s_i(1 - s_i)) : i = 1, ..., m$$

Ω is positive definite $\forall w \rightarrow X^T \Omega X$ is positive definite $\forall w \rightarrow J(w)$ is convex.
Newton's method:

1. $w \leftarrow 0$
2. repeat until convergence:
3. $e \leftarrow$ solution to normal equations $(X^T \Omega X) e = X^T (y - s)$
4. $w \leftarrow w + e$

Ω prioritizes pts. with s_i near 0.5; tunes out pts. near 0/1.
Idea: If n very large, save time by using a random sub-sample of the pts. per iteration. Increase sample size as you go.

6.5 LDA v. Logistic Regression

Advantages of LDA:

- for well-separated classes, LDA stable; log.reg. surprisingly unstable (because higher weight is placed onto points that are near the decision boundary.
- > 2 classes easy and elegant; log. reg needs modifying (softmax - regression).
- LDA slightly more accurate when classes nearly normal, especially if n is small.

Advantages of log. reg:

- More emphasis on decision boundary.
- Hence less sensitive*- to outliers
- Easy Elegant treatment of 'partial' class membership; LDA pts all-or-nothing
- more robust on some non-Gaussian distributions (eg. dists w/large skews)

6.6 ROC CURVE

Lets us measure the change in classification accuracy of false positives versus false negatives.