Lecture 4c. Training logistic regression with the IRLS algorithm

COMP90051 Statistical Machine Learning

Semester 2, 2020
Lecturer: Ben Rubinstein

THE UNIVERSITY OF MELBOURNE

This lecture

- Iterative optimisation for extremum estimators
  - First-order method: Gradient descent
  - Second-order: Newton-Raphson method
  - Later: Lagrangian duality
- Logistic regression: workhorse linear classifier
  - Possibly familiar derivation: frequentist
  - Decision-theoretic derivation
  - **Training with Newton-Raphson** looks like repeated, weighted linear regression

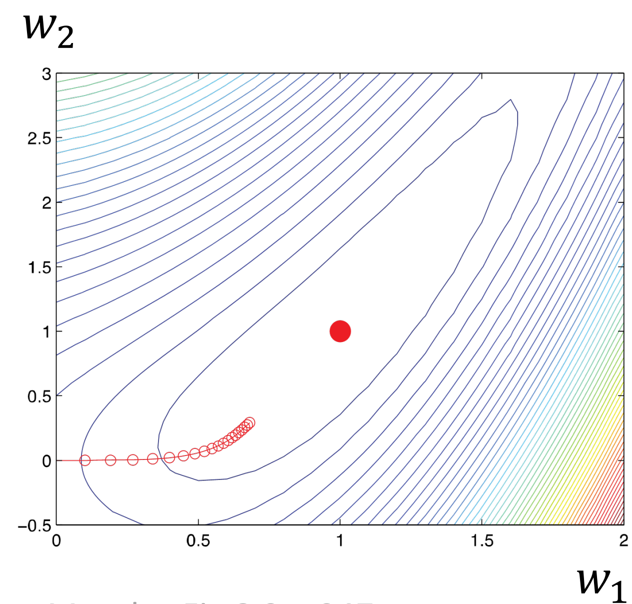
Training Logistic Regression: the IRLS Algorithm

Analytical? Newton-Raphson!

Iterative optimisation

- Training logistic regression: $\mathbf{w}$ maximising log-likelihood $L(\mathbf{w})$ or cross-entropy loss
- **Bad news**: No closed form solution
- **Good news**: Problem is strictly convex, if no irrelevant features → convergence!

Look ahead: regularisation for irrelevant features



Murphy, Fig 8.3, p247

How does gradient descent work?

- $\mu(z) = \frac{1}{1+\exp(-z)}$ then $\frac{d\mu}{dz} = \mu(z)(1-\mu(z))$
- Then $\nabla L(\mathbf{w}) = \sum_{i=1}^{n}\big(y_n - \mu(\mathbf{x}_n)\big)\mathbf{x}_n = \mathbf{X}'(\mathbf{y}-\boldsymbol{\mu})$, stacking instances in $\mathbf{X}$, labels in $\mathbf{y}$, $\mu(\mathbf{x}_n)$ in $\boldsymbol{\mu}$

Note I'm abusing notation:
$\mu(\mathbf{x}_n) = \mu(z)$ where $z = \mathbf{w}'\mathbf{x}_n$
Meaning by input type

$$L(w) = \sum \left(y_i \log(\mu(x_i)) + (1-y_i)\log(1-\mu(x_i)) \right)$$

$$\log(\mu(x_i)) = \log \frac{1}{1+e^{-wx_i}} = -\log(1+e^{-wx_i})$$

$$\log(1-\mu(x_i)) = \log\left(1-\frac{1}{1+e^{-wx_i}}\right) = \log(e^{-wx_i}) - \log(1+e^{-wx_i}) = -wx_i -$$

now: $$L(w) = \sum\left(-y_i\log(1+e^{-wx_i}) + (1-y_i)\cdot(-wx_i - \log(1+e^{-wx_i}))\right)$$

$$= -\sum\left(y_i\log(1+e^{-wx_i}) + (1-y_i)(wx_i+\log(1+e^{-wx_i}))\right)$$

$$= -\sum\left(wx_i + \underline{\log(1+e^{-wx_i}) - wx_iy_i}\right)$$

$$= -\sum\left(\log(e^{wx_i}) + \quad\right) = \sum\left(wx_iy_i - \log(1+e^{wx_i})\right)$$

$$\therefore \frac{\partial L(w)}{\partial w} = \sum\left(x_iy_i - \frac{x_ie^{wx_i}}{1+e^{wx_i}}\right) = \sum(x_iy_i - x_i\mu(z)) = \sum x_i(y_i-\mu(z))$$

$$= \sum x_i\left(y_i - \frac{1}{e^{-wx_i}+1}\right)$$

$$\frac{x_ie^{wx_i}}{1+e^{wx_i}} = \frac{x_i}{e^{-wx_i}+1}$$

Iteratively-Reweighted Least Squares

- Instead of GD, let's apply Newton-Raphson → IRLS algorithm
- Recall: $\nabla L(\mathbf{w}) = \mathbf{X}'(\mathbf{y} - \boldsymbol{\mu})$. Differentiate again for Hessian:

$$\nabla_2 L(\mathbf{w}) = -\sum_i \frac{d\mu}{dz_i}\mathbf{x}_n\mathbf{x}_n{}' = -\sum_i \mu(\mathbf{x}_i)\big(1-\mu(\mathbf{x}_i)\big)\mathbf{x}_n\mathbf{x}_n'$$
$$= -\mathbf{X}'\mathbf{M}\mathbf{X}, \text{ where } M_{ii} = \mu_i(1-\mu_i) \text{ otherwise } 0$$

- Newton-Raphson then says (now with $\mathbf{M}_t, \boldsymbol{\mu}_t$ dependence on $\mathbf{w}_t$)

$$\mathbf{w}_{t+1} = \mathbf{w}_t - (\nabla_2 L)^{-1}\nabla L = \mathbf{w}_t + (\mathbf{X}'\mathbf{M}_t\mathbf{X})^{-1}\mathbf{X}'(\mathbf{y}-\boldsymbol{\mu}_t)$$
$$= (\mathbf{X}'\mathbf{M}_t\mathbf{X})^{-1}[\mathbf{X}'\mathbf{M}_t\mathbf{X}\mathbf{w}_t + \mathbf{X}'(\mathbf{y}-\boldsymbol{\mu}_t)]$$
$$= (\mathbf{X}'\mathbf{M}_t\mathbf{X})^{-1}\mathbf{X}'\mathbf{M}_t\mathbf{b}_t, \text{ where } \mathbf{b}_t = \mathbf{X}\mathbf{w}_t + \mathbf{M}_t{}^{-1}(\mathbf{y}-\boldsymbol{\mu}_t)$$

Compare to normal equations

- Each IRLS iteration solves a least squares problem weighted by $\mathbf{M}_t$, which are reweighted iteratively!

IRLS intuition: Putting labels on linear scale

IRLS: $\mathbf{w}_{t+1} = (\mathbf{X}'\mathbf{M}_t\mathbf{X})^{-1}\mathbf{X}'\mathbf{M}_t\mathbf{b}_t$
where $\mathbf{b}_t = \mathbf{X}\mathbf{w}_t + \mathbf{M}_t{}^{-1}(\mathbf{y} - \boldsymbol{\mu}_t)$
and $M_{ii} = \mu_t(\mathbf{x}_i)[1 - \mu_t(\mathbf{x}_i)]$ otherwise 0
and $\mu_t(\mathbf{x}) = [1 + \exp(-\mathbf{w}_t{}'\mathbf{x})]^{-1}$

- The $\mathbf{y}$ are not on linear scale. Invert logistic function?
- The $\mathbf{b}_t$ are a "linearised" approximation to these: the $\mathbf{b}_t$ equation matches a linear approx. to $\mu_t{}^{-1}(\mathbf{y})$.
- Linear regression on new labels!

- Setting derivative to zero and solving for $\boldsymbol{w}$ yields
  $$\hat{\boldsymbol{w}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$
  - This system of equations called the normal equations
  - System is well defined only if the inverse exists

IRLS intuition: Equalising label variance

IRLS: $\mathbf{w}_{t+1} = (\mathbf{X}'\mathbf{M}_t\mathbf{X})^{-1}\mathbf{X}'\mathbf{M}_t\mathbf{b}_t$
where $\mathbf{b}_t = \mathbf{X}\mathbf{w}_t + \mathbf{M}_t{}^{-1}(\mathbf{y} - \boldsymbol{\mu}_t)$
and $M_{ii} = \mu_t(\mathbf{x}_i)[1 - \mu_t(\mathbf{x}_i)]$ otherwise 0
and $\mu_t(\mathbf{x}) = [1 + \exp(-\mathbf{w}_t{}'\mathbf{x})]^{-1}$

- In linear regression, each y_i has equal variance σ^2
- Our y_i are Bernoulli, variance: $\mu_t(\mathbf{x}_i)[1 - \mu_t(\mathbf{x}_i)]$
- Our reweighting standardises, dividing by variances*!!*

Fun exercise: Show that Newton-Raphson for linear regression gives you the normal equations!

Summary

- Training logistic regression
  - No analytical solution
  - Gradient descent possible, but convergence rate not ideal
  - Newton-Raphson: iteratively reweighted least squares

Next time: Regularised linear regression for avoiding overfitting and ill-posed optimisation