Logistic Regression

Lyle Ungar

Learning objectives
Logistic model & loss
Decision boundaries as hyperplanes
Multi-class regression
What do you do with a binary $y$?

- Can you use linear regression?
  - $y = w^T x$

- How about a different link function?
  - $y = f(w^T x)$

- Or a different probability distribution
  - $P(y=1|x) = f(w^T x)$
Logistic function

$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$
Logistic Regression

\[ P(Y = 1|x, w) = \frac{1}{1 + \exp\{-\sum_j w_j x_j\}} = \frac{1}{1 + \exp\{-w^T x\}} = \frac{1}{1 + \exp\{-yw^T x\}} \]

\[ P(Y = -1|x, w) = 1 - P(Y = 1|x, w) = \frac{\exp\{-w^T x\}}{1 + \exp\{-w^T x\}} = \frac{1}{1 + \exp\{-yw^T x\}} \]

\[ \log\left(\frac{P(Y=1|x, w)}{P(Y=-1|x, w)}\right) = w^T x \quad \text{Log odds} \]
Log likelihood of data

\[ \log(P(D_Y|D_X, w)) = \log \left( \prod_i \frac{1}{1 + \exp\{-y_i w^\top x_i\}} \right) \]

\[ = - \sum_i \log(1 + \exp\{-y_i w^\top x_i\}) \]

\( y = 1 \) or \(-1\)
Decision Boundary

\[ P(Y = 1|x, w) = P(Y = -1|x, w) \]

\[
\frac{1}{1 + \exp\{-w^T x\}} = \frac{\exp\{-w^T x\}}{1 + \exp\{-w^T x\}}
\]

\[ w^T x = 0 \]
Representing Hyperplanes

- How do we represent a line?
  
  \[
  y = x \\
  0 = x - y \\
  0 = [1, -1] \begin{bmatrix} x \\ y \end{bmatrix}
  \]

- In general, a hyperplane is defined by
  
  \[
  0 = w^T x
  \]

  The red vector \( w \) defines the green hyperplane that is orthogonal to it.

  Why bother with this weird representation?
Projections

\((\vec{w} \cdot \vec{x})\vec{w}\) is the projection of \(\vec{x}\) onto \(\vec{w}\)
Now classification is easy!

\[ h(x) = \text{sgn}(w^T x) \]
Computing MLE

- Use gradient ascent

\[ w^{t+1} = w^t + \eta_t \nabla_w \ell(w) \]

Loss function = log-likelihood

\[
\nabla_w \ell(w) = \frac{\delta \log(P(D_y|D_x,w))}{\delta w} = \sum_i y_i x_i \frac{\exp\{-y_i w^T x_i\}}{1+\exp\{-y_i w^T x_i\}} = \sum_i y_i x_i (1 - P(y_i|x_i, w))
\]
Computing MAP

- Prior
  
  \[ w_j \sim \mathcal{N}(0, \gamma^2) \text{ so } P(w) = \prod_j \frac{1}{\gamma \sqrt{2\pi}} \exp \left\{ -\frac{w_j^2}{2\gamma^2} \right\} \]

- So solve

  \[
  \arg \max_w \log P(w | D, \gamma) = \arg \max_w (\ell(w) + \log P(w | \gamma))
  \]

  \[
  \arg \max_w \left( \ell(w) - \frac{1}{2\gamma^2} w^\top w \right)
  \]

- Again use gradient descent
Questions?
Multi-Class Classification

Disease diagnosis: healthy / cold / flu / pneumonia
Object classification: desk / chair / monitor / bookcase
Multi-Class Logistic Regression

- For 2 classes:
  \[ h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)} = \frac{\exp(\theta^T x)}{1 + \exp(\theta^T x)} \]

- For K classes:
  \[ p(y = k \mid x; \theta_1, \ldots, \theta_k) = \frac{\exp(\theta_k^T x)}{\sum_{k=1}^{K} \exp(\theta_k^T x)} \]

  - Called the **softmax** function
    - maps a vector to a probability distribution
Multi-Class Logistic Regression

- Train a logistic regression classifier for each class $k$ to predict the probability that $y = k$ with

$$h_k(x) = \frac{\exp(\theta_k^T x)}{\sum_{k=1}^k \exp(\theta_k^T x)}$$
Implementing Multi-Class Logistic Regression

- $P(y=k|x)$ estimated by: 
  $$h_k(x) = \frac{\exp(\theta_k^T x)}{\sum_{k=1}^K \exp(\theta_k^T x)}$$

- Gradient descent simultaneously updates all parameters for all models
  - Same derivative as before, just with the above $h_k(x)$

- Predict class label as the most probable label
You should know

- **Logistic model & loss**
  - Linear in log-odds

- **Decision boundaries**
  - hyperplane

- **Softmax**
  - Maps vector to probability distribution
Questions?