Multinomial Logistic Regression/Maximum Entropy

Fernando Pereira

CIS 620 2005

Multinomial Logistic Regression

- Model form

$$P_{\mathbf{w}}(y \mid \mathbf{x}) = \frac{\exp \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, y)}{Z_{\mathbf{w}}(\mathbf{x})}$$

$$Z_{\mathbf{w}}(\mathbf{x}) = \sum_{y} \exp \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, y)$$

- Useful properties
  - Multi-class
  - May use different features for different classes
  - Training is convex optimization

Maxent Duality

- Maximize *conditional log likelihood*

$$L_{\mathbf{w}} = \sum_i \log P_{\mathbf{w}}(y_i \mid \mathbf{x}_i)$$
$$\tilde{\mathbf{w}} = \arg\max_{\mathbf{w}} L_{\mathbf{w}}$$

- Maximizing conditional entropy

$$\tilde{P} = \arg\max_P \left[-\sum_i \sum_y P(y \mid \mathbf{x}_i) \log P(y \mid \mathbf{x}_i) \right]$$

subject to constraints

$$\sum_i f(\mathbf{x}_i, y_i) = \sum_y P(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) \quad \forall f$$

yields *data average* = *model expectation*

$$\tilde{P}(y \mid \mathbf{x}) = P_{\tilde{\mathbf{w}}}(y \mid \mathbf{x}) = \frac{\exp \tilde{\mathbf{w}} \cdot \mathbf{f}(\mathbf{x}, y)}{Z_{\tilde{\mathbf{w}}}(\mathbf{x})}$$

Maximizing Log-Likelihood

- Log-likelihood gradient

$$\frac{\partial L_{\mathbf{w}}}{\partial w_f} = \sum_i f(\mathbf{x}_i, y_i) - \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y)$$

data average

model expectation

- At maximum feature averages and expectations coincide
- Training algorithm: use your favorite convex optimizer
  - Second-order methods: big Hessian
  - Limited-memory approximations of 2nd order (eg. L-BFGS)

Overfitting

- Small count features lead to unreliably large feature weights
- *Question*: what happens if a binary feature f is associated to just one class c and is *on* for all c training instances?

$$\sum_i f(\mathbf{x}_i, y_i) = \sum_i f(\mathbf{x}_i, c) = \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) = \sum_i P_{\mathbf{w}}(c \mid \mathbf{x}_i) f(\mathbf{x}_i, c)$$

$$\Downarrow$$

$$P_\Lambda(c \mid \mathbf{x}_i) \to 1 \Rightarrow w_f \to +\infty$$

- What if f is off for all c training instances?

Gaussian Prior

- Penalize large weights

$$P'_{\mathbf{w}}(y \mid x) = P_{\mathbf{w}}(y \mid x) P(\mathbf{w})$$ *weight prior*

$$L'_{\mathbf{w}} = L_{\mathbf{w}} - \sum_f \frac{w_f^2}{2\sigma_f^2} + \text{const}(\mathbf{w})$$

- Penalized log-likelihood gradient

$$\frac{\partial L'_\Lambda}{\partial w_f} = \sum_i f(\mathbf{x}_i, y_i) - \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) - \frac{w_f}{\sigma_f^2}$$

Relationship to (Binary) Logistic Discrimination

$$
\begin{aligned}
p(+1 \mid \mathbf{x}) &= \frac{\exp \sum_f w_f f(\mathbf{x},+1)}{\exp \sum_f w_f f(\mathbf{x},+1) + \exp \sum_f w_f f(\mathbf{x},-1)} \\
&= \frac{1}{1 + \exp - \sum_f w_f \left(f(\mathbf{x},+1) - f(\mathbf{x},-1) \right)} \\
&= \frac{1}{1 + \exp - \sum_f w_f g_f(\mathbf{x})}
\end{aligned}
$$

Relationship to Linear Discrimination

- Decision rule

$$\operatorname{sign}\left(\log \frac{p(+1 \mid \mathbf{x})}{p(-1 \mid \mathbf{x})}\right) = \operatorname{sign} \mathbf{w} \cdot \mathbf{g}(\mathbf{x})$$

- Bias term: parameter for "always on" feature

$$(b, w_1, \cdots, w_d) \cdot (1, x_1, \cdots, x_d)$$

- *Question*: relationship to other trainers for linear discriminant functions

Solution Techniques (I)

- Generalized iterative scaling (GIS)
  - Parameter updates

$$w_f \leftarrow w_f + \frac{1}{C} \log \frac{\sum_i f(\mathbf{x}_i, y_i)}{\sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y)}$$

  - Requires that features add up to constant independent of instance or label (add *slack feature*)

$$\sum_f f(\mathbf{x}_i, y) = C \quad \forall i, y$$

Solution Techniques (2)

- Improved iterative scaling (IIS)
  - Parameter updates

$$w_f \leftarrow w_f + \delta_f$$

$$\sum_i f(\mathbf{x}_i, y_i) = \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) e^{\delta_f \#(\mathbf{x}_i, y)}$$

$$\#(\mathbf{x}, y) = \sum_f f_f(\mathbf{x}, y)$$

  - For binary features reduces to solving a polynomial with positive coefficients
  - Reduces to GIS if feature sum constant

Deriving IIS (1)

- Conditional log-likelihood

$$l(\mathbf{w}) = \sum_i \log P_{\mathbf{w}}(y_i \mid \mathbf{x}_i)$$

- Log-likelihood update

$$\begin{aligned}
l(\mathbf{w}+\Delta) - l(\mathbf{w}) &= \sum_i \left[\Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) - \log \frac{Z_{\mathbf{w}+\Delta}(\mathbf{x}_i)}{Z_{\mathbf{w}}(\mathbf{x}_i)} \right] \\
&= \sum_i \Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) - \sum_i \log \sum_y \frac{e^{(\mathbf{w}+\Delta)\cdot \mathbf{f}(\mathbf{x}_i, y)}}{Z_{\mathbf{w}}(\mathbf{x}_i)} \\
&= \sum_i \Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) - \sum_i \log \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) e^{\Delta \cdot \mathbf{f}(\mathbf{x}_i, y)} \\
(\log x \le x - 1) \quad &\ge \underbrace{\sum_i \Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) + N - \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) e^{\Delta \cdot \mathbf{f}(\mathbf{x}_i, y)}}_{A(\Delta)}
\end{aligned}$$

IIS (2)

- Maximizing $A(\Delta)$ puts a lower bound on the likelihood improvement
- But differentiating $A(\Delta)$ couples all the parameters
- Rewrite

$$A(\Delta) = \sum_i \Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) + N - \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) \exp\left(\#(\mathbf{x}_i, y) \sum_f \frac{\delta_f f(\mathbf{x}_i, y)}{\#(\mathbf{x}_i, y)} \right)$$

$$= \sum_i \Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) + N - \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) \exp\left(\sum_f \frac{f(\mathbf{x}_i, y)}{\#(\mathbf{x}_i, y)} \delta_f \#(\mathbf{x}_i, y) \right)$$

IIS (3)

- By Jensen's inequality $\exp \sum_z p(z)q(z) \leq \sum_z p(z) \exp q(z)$

$$A(\Delta) \geq \sum_i \Delta \cdot \mathbf{f}(\mathbf{x}_i, y_i) + N -$$

$$\sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) \sum_f \frac{f(\mathbf{x}_i, y)}{\#(\mathbf{x}_i, y)} e^{\delta_f \#(\mathbf{x}_i, y)} = B(\Delta)$$

$$\frac{\partial B(\Delta)}{\partial \delta_f} = \sum_i f(\mathbf{x}_i, y_i) - \sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) e^{\delta_f \#(\mathbf{x}_i, y)}$$

- Solve for δ_f (1-dimensional Newton's method)
- Maximize lower bound on update

Solution Techniques (3)

- GIS very slow if slack variable takes large values
- IIS faster, but still problematic
- Second-order convex optimization methods seem to do better

Gaussian Prior

- Log-likelihood gradient

$$\frac{\partial l(\Lambda)}{\partial w_k} = \sum_i f(\mathbf{x}_i, y_i) - \sum_i \sum_y P_\Lambda(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) - \frac{w_f}{\sigma_f^2}$$

- Modified IIS update

$$w_f \leftarrow w_f + \delta_f$$

$$\sum_i f(\mathbf{x}_i, y_i) =$$

$$\sum_i \sum_y P_{\mathbf{w}}(y \mid \mathbf{x}_i) f(\mathbf{x}_i, y) e^{\delta_f \#(\mathbf{x}_i, y)} + \frac{w_f + \delta_f}{\sigma_f^2}$$

$$\#(\mathbf{x}, y) = \sum_f f(\mathbf{x}, y)$$