Learning Linear Classifiers

- Quality metric:

For example, input: #awesome, #awful; output: +1/-1 (sentiment)

- If we take the negative examples and the positive examples, there might not be $\widehat{w}$ that achieves exactly 0 for the negatives, 1 for the positives, for all of them. So the quality metric of the likelihood function tries to figure out kind of an average. It measures the quality throughout all the data points.

- Best model: Highest likelihood.

- The quality metric we try to optimize is Data Likelihood.

--- Data Likelihood example1:

x[1] = #awesome = 2, x[2] = #awful = 1, y1 = sentiment = +1

If model is good, should predict: $\widehat{y1}$= +1

Pick w to maximize: $P(y_1 = +1|x_1, w)$

--- Data Likelihood example2:

x[1] = #awesome = 0, x[2] = #awful = 2, y1 = sentiment = -1

If model is good, should predict: $\widehat{y2}$= -1

Pick w to maximize: $P(y_2 = -1|x_2, w)$

--- Lots of data points

We want to do that (maximize $P(\hat{y}_i = +1|x_i, w, y_i = +1)$, maximize $P(\hat{y}_i = -1|x_i, w, y_i = -1)$).

--- Maximizing likelihood (probability of data)

$$L(w) = P(y_1|x_1, w) * P(y_2|x_2, w) * \dots * P(y_N|x_N, w) = \prod_{i=1}^{N} P(y_i|x_i, w)$$

- Gradient Ascent algorithm

- Hill climbing algorithm.

- Moving to multiple dimensions: Gradients

$$\nabla \mathrm{L}(\mathrm{w}) = \begin{pmatrix} \frac{\partial L}{\partial w_0} \\ \frac{\partial L}{\partial w_1} \\ \cdots \\ \frac{\partial L}{\partial w_D} \end{pmatrix}$$

- Derivative of log-likelihood

$$\frac{\partial L(w)}{\partial w_j} = \sum_{i=1}^{N} h_j(x_i)(I[y_i = +1] - P(y = +1|x_i, w))$$

I is the "indicator function": $I[y_i = +1] = \begin{cases} 1, & if\ y = +1 \\ 0, & if\ y = -1 \end{cases}$

$$w_i^{(t+1)} = w_i^{(t)} + \eta \frac{\partial L(w_i^{(t)})}{\partial w_i}$$

- Pseudo code:

Initial $w^{(1)} = 0$ (or randomly/smartly), t = 1

while$\left\|\nabla L(w^{(t)})\right\| > \varepsilon$:

for $j = 0, \ldots, D$:

$partial[j] = \sum_{i=1}^{N} h_j(x_i)(I[y_i = +1] - P(y = +1|x_i, w^{(t)}))$

$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta * partial[j]$

$t \leftarrow t + 1$

- Choosing step size:

- Pick the step size: build a learning curve (x: # of iterations; y: log likelihood over all data points.)

- If too large: oscillate, may never converge; if too small: too slow.

- Picking step size requires a lot of trial and error.

- Try several values, exponentially spaced.

- Try values in between to find "best" eta.

- Overfitting:

- Classification error:

$$error = \frac{\#\ Mistakes}{Total\ number\ of\ data\ points}$$

- Accuracy:

$$accuracy = \frac{\#\ Correct}{Total\ number\ of\ data\ points}$$

- Overfitting if there exists w^*: 1) $training\_error(w^*) > training\_error(\widehat{w})$; 2) $true\_error(w^*) < true\_error(\widehat{w})$

- Often, overfitting associated with very large estimated coefficients

- Desired total cost format

- Want to balance: 1) How well the function fits data; 2) Magnitude of coefficients.

- Total quality = (want to balance) measure of fit – measure of magnitude of coefficients.

- Maximum Likelihood Estimation (MLE)

Measure of fit = Data likelihood

- Choose coefficients *w* that maximize likelihood:

$$\prod_{i=1}^{N} P(y_i|X_i, w)$$

- Typically, we use the log of likelihood function:

$$l(w) = ln \prod_{i=1}^{N} P(y_i|X_i, w)$$

- Measure of magnitude of logistic regression coefficients.

- (L2 norm) Sum of squares

$$||w||_2^2 = w_0^2 + w_1^2 + w_2^2 + \cdots + w_D^2$$

- Sum of absolute value (L1 norm)

$$\|w\|_1 = \|w_0\| + \|w_1\| + \|w_2\| + \cdots + \|w_D\|$$

- Consider specific total cost

$$Total\ quality = measure\ of\ fit - measure\ of\ maginitude\ of\ coefficients$$

- Consider resulting objective.

- Select $\widehat{w}$ to minimize

$$l(w) - \lambda\|w\|_2^2$$

Where λ is the tuning parameter to balance fit and magnitude.

- if $\lambda = 0$ --- Standard (unpenalized) MLE solution.

- if $\lambda = \infty$ --- Only cares about penalizing the large coefficients.

- if λ in between --- Pick λ using: 1) Validation set (for large datasets); 2) Cross-validation (for smaller datasets).

- Bias-variance tradeoff:

- Large λ: high bias, low variance.

- Small λ: low bias, high variance.

- In essence, λ controls model complexity.

- Gradient of L2 regularized log-likelihood.

- Total quality = measure of fit – measure of magnitude of coefficients

- $Total\ derivative = \frac{\partial l(w)}{\partial w_j} - \lambda \frac{\partial \|w\|_2^2}{\partial w_j}$

- Derivative of (log-) likelihood

- $\frac{\partial l(w)}{\partial w_j} = \sum_{i=1}^{N} h_j(X_i)(I[y_i = +1] - P(y = +1|X_{i,}w))$

- $\frac{\partial \|w\|_2^2}{\partial w_j} = \frac{\partial}{\partial w_j}\left(w_0^2 + w_1^2 + w_2^2 + \cdots + w_j^2 + \cdots + w_D^2\right) = 2w_j$

- $Total\ derivative = \frac{\partial l(w)}{\partial w_j} - 2\lambda w_j$

- Summary of gradient ascent for logistic regression with L2 Regularization.

  Initial $w^{(1)} = 0$ (or randomly, or smartly), t=1

  While not converged:

  For j = 0, …, D:

$$partical[j] = \sum_{i=1}^{N} h_j(X_j)(I[y_i = +1] - P(y = +1|X_i, w^{(t)}))$$

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta(partial[j] - 2\lambda w_j^{(t)})$$

$$t \leftarrow t + 1$$

- Sparsity (many $\widehat{w}_j = 0$) gives efficiency and interpretability.

- Efficiency:

1) if $size(w) = 100B$, each prediction is expensive.

2) if $\widehat{w}$ sparse (many zeros), computation only depends on # of non-zeros.

$$\hat{y}_i = sign(\sum_{\widehat{w}_j \neq 0} \widehat{w}_j h_j(X_i))$$

- Sparse logistic regression

$$Total\ quality = measur\ of\ fit - measure\ of\ magnitude\ of\ coefficients$$

- L1 regularized logistic regression

- Just like L2 regularization, solution is governed by a continuous parameter λ,

$$l(w) - \lambda\|w\|_1$$

If $\lambda = 0$: No regularization --- Standard MLE solution.

If $\lambda = \infty$: All weights are on regularization.

If λ in between: Sparse solutions: Some $\widehat{w}_j \neq 0$, many other $\widehat{w}_j = 0$.