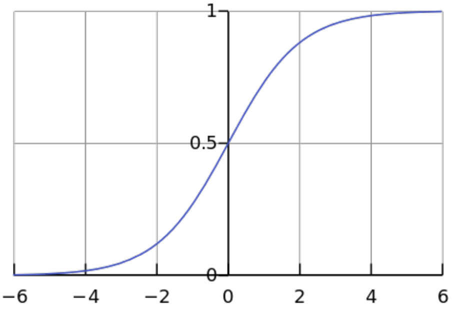
Lecture 17: Logistic Regression contd.

Instructor: Prof. Ganesh Ramakrishnan

Sigmoidal (perceptron) Classifier

1. **(Binary) Logistic Regression**, abbreviated as **LR** is a single node perceptron-like classifier, but with....
   - $sign\left((\mathbf{w}^*)^T\phi(\mathbf{x})\right)$ replaced by $g\left((\mathbf{w}^*)^T\phi(\mathbf{x})\right)$ where $g(s)$ is sigmoid function: $g(s) = \frac{1}{1+e^{-s}}$
2. $f_{\mathbf{w}}(\mathbf{x}) = g\left((\mathbf{w}^*)^T\phi(\mathbf{x})\right) = \frac{1}{1+e^{-(\mathbf{w}^*)^T\phi(\mathbf{x})}} \in [0,1]$ can be interpreted as $Pr(y=1|\mathbf{x})$
   - Then $Pr(y=0|x) = 1 - f_{\mathbf{w}}(\mathbf{x})$



Logistic Regression: The Sigmoidal (perceptron) Classifier

1. Estimator $\widehat{\mathbf{w}}$ is a function of the dataset $\mathcal{D} = \left\{(\phi(\mathbf{x}^{(1)}, y^{(1)}), (\phi(\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\phi(\mathbf{x}^{(m)}, y^{(m)}))\right\}$
   - Estimator $\widehat{\mathbf{w}}$ is meant to approximate the parameter $\mathbf{w}$.
2. Maximum Likelihood Estimator: Estimator $\widehat{\mathbf{w}}$ that maximizes the likelihood $L(\mathcal{D}; \mathbf{w})$ of the data $\mathcal{D}$.
   - Assumes that all the instances $(\phi(\mathbf{x}^{(1)}, y^{(1)}), (\phi(\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\phi(\mathbf{x}^{(m)}, y^{(m)}))$ in $\mathcal{D}$ are all independent and identically distributed (iid)
   - Thus, Likelihood is the probability of $\mathcal{D}$ under iid assumption: $\hat{\mathbf{w}} = \max_{\mathbf{w}} L(\mathcal{D}, \mathbf{w}) =$

Logistic Regression: The Sigmoidal (perceptron) Classifier

1. Estimator $\widehat{\mathbf{w}}$ is a function of the dataset
$\mathcal{D} = \left\{(\phi(\mathbf{x}^{(1)}, y^{(1)}), (\phi(\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\phi(\mathbf{x}^{(m)}, y^{(m)}))\right\}$
    - Estimator $\widehat{\mathbf{w}}$ is meant to approximate the parameter $\mathbf{w}$.
2. Maximum Likelihood Estimator: Estimator $\widehat{\mathbf{w}}$ that maximizes the likelihood $L(\mathcal{D}; \mathbf{w})$ of the data $\mathcal{D}$.
    - Assumes that all the instances $(\phi(\mathbf{x}^{(1)}, y^{(1)}), (\phi(\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\phi(\mathbf{x}^{(m)}, y^{(m)}))$ in $\mathcal{D}$ are all independent and identically distributed (iid)
    - Thus, Likelihood is the probability of $\mathcal{D}$ under iid assumption: $\hat{\mathbf{w}} = \max_{\mathbf{w}} L(\mathcal{D}, \mathbf{w}) =$

$$\text{argmax}_{\mathbf{w}} \prod_{i=1}^{m} p(y^{(i)}|\phi(\mathbf{x}^{(i)})) = \text{argmax}_{\mathbf{w}} \prod_{i=1}^{m} \Bigg(\underbrace{\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}}_{\text{If } y^{(i)}=1 \text{ then } Pr(y=1|x^{(i)})}\Bigg)^{y^{(i)}} \Bigg(\underbrace{\frac{e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}}_{\text{If } y^{(i)}=0 \text{ then } Pr(y=0|x^{(i)})}\Bigg)^{1-y^{(i)}}$$

Training LR

Recall Entropy

$-\sum p_i \log(p_i)$

1. Thus, Maximum Likelihood Estimator for $\mathbf{w}$ is

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\ L(\mathcal{D}, w) = \underset{\mathbf{w}}{\text{argmax}} \prod_{i=1}^{m} p(y^{(i)} | \phi(\mathbf{x}^{(i)}))$$

$$= \underset{\mathbf{w}}{\text{argmax}} \prod_{i=1}^{m} \left(\frac{1}{1 + e^{-\mathbf{w}^T \phi(\mathbf{x}^{(i)})}} \right)^{y^{(i)}} \left(\frac{e^{-\mathbf{w}^T \phi(\mathbf{x}^{(i)})}}{1 + e^{-\mathbf{w}^T \phi(\mathbf{x}^{(i)})}} \right)^{1-y^{(i)}}$$

$$= \underset{\mathbf{w}}{\text{argmax}} \prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}}$$

$$-\frac{1}{m}\sum_{i=1}^{m} \left\{ y^{(i)} \log f_w(x^{(i)}) + (1-y^{(i)}) \log(1-f_w(x^{(i)})) \right\}$$

2. Maximizing the likelihood $\Pr(\mathcal{D}; \mathbf{w})$ w.r.t $\mathbf{w}$, is the same as minimizing the negative log-likelihood $E(\mathbf{w}) = -\frac{1}{m} \log \Pr(\mathcal{D}; \mathbf{w})$ w.r.t $\mathbf{w}$.
   - Derive the expression for $E(\mathbf{w})$.
   - $E(\mathbf{w})$ is called the cross-entropy loss function

Minimizing negative Log-likelihood for LR

1. The Cross-entropy Loss function:[1]

then this is low & this $-\log(\cdot)$ is high

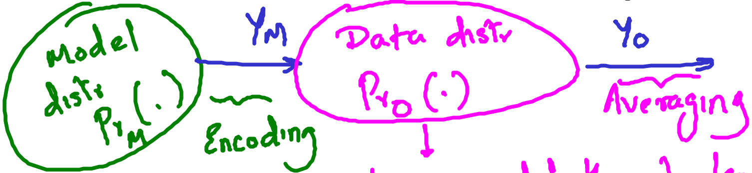
$$E(w) = -\frac{1}{m}\sum_{i=1}^{m} \overbrace{Pr_0(Y=1|x^{(i)})}\log\left(\overbrace{Pr_M(Y=1|x^{(i)})}\right)$$

$$+ \underbrace{Pr_0(Y=0|x^{(i)})}\log\left(\underbrace{Pr_M(Y=0|x^{(i)})}\right)$$

If this prob is also high

If this prob is close to 1, $-\log(\cdot)$ low

If $Pr_0 \approx Pr_M$ then $E(w)$ will be low



Viewing model through lens of data

[1] https://en.wikipedia.org/wiki/Cross_entropy

$$-\frac{1}{m}\sum_i y^i \log(f_w(x^i)) + (1-y^i)\log(1-f_w(x^i))$$

$$f_w(x^i) = \frac{1}{1+e^{-w^T\phi(x)}} \quad \& \quad 1-f_w(x) = \frac{e^{-w^T\phi(x)}}{1+e^{-w^T\phi(x)}}$$

$$\Downarrow$$

$$-\frac{1}{m}\sum_i y^i\left(\cancel{\log(1)} - \log\left(1+e^{-w^T\phi(x^i)}\right)\right)$$

$$+ (1-y^i)\left(\underbrace{\log\left(e^{-w^T\phi(x^i)}\right)}_{-w^T\phi(x^i)} - \log\left(1+e^{-w^T\phi(x^i)}\right)\right)$$

$$= -\frac{1}{m}\sum_i y_i w^T\phi(x^i) - \log\left(\left(1+e^{-w^T\phi(x^i)}\right)e^{w^T\phi(x^i)}\right)$$

Minimizing negative Log-likelihood for LR

1. The Cross-entropy Loss function: "Observed/Empirical distribution" $Pr_O(y=1|x^{(i)})$

$f_w(x) = \frac{1}{1+e^{-w^T\phi(x)}}$

$$E(\mathbf{w}) = -\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (1)$$

Model distribution $Pr_M(y=1|x^{(i)})$

or with some simplification,

$$E(\mathbf{w}) = -\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\mathbf{w}^T\phi(\mathbf{x}^{(i)}) - \log\left(1+exp\left(\mathbf{w}^T\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (2)$$

Like unsigned distance error of perceptron

2. Cross-entropy[1] is the average number of bits needed to identify an event (example $\mathbf{x}$) drawn from the (data) set $\mathcal{D}$, if a coding scheme is used that is optimized for a modeled probability distribution $\Pr\left(y|\mathbf{w},\phi(.)\right)$, rather than the 'true' distribution $\Pr(y|\mathcal{D})$.

$$E(\mathbf{w}) = \mathbf{E}_{\Pr(y|\mathcal{D})}\left[-\log\Pr\left(y|\mathbf{w},\phi(.)\right)\right] \quad (3)$$

empirical distr

[1] https://en.wikipedia.org/wiki/Cross_entropy

Gradient descent for LR

1. No closed form solution to the cross-entropy loss

$$\widehat{\mathbf{w}}^{MLE} = \underset{\mathbf{w}}{\operatorname{argmin}} - \left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (4)$$

2. Apply gradient descent with $\mathbf{w}^{(k+1)} = \mathbf{w}^k - \eta\nabla E\left(\mathbf{w}^k\right)$

$$\nabla E(\omega^k) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\,\nabla_{\omega^k}\left(\log f_\omega(x^{(i)})\right) + (1-y^{(i)})\,\nabla_{\omega^k}\left(\log(1-f_\omega(x^{(i)}))\right)$$

$$f_\omega(x) = \frac{1}{1+e^{-\omega^T\phi(x)}}$$

$$\log(f_\omega(x)) = -\log(1+e^{-\omega^T\phi(x)})$$

$$\nabla_{\omega^k}\left(\log f_\omega(x^{(i)})\right) = \left(\frac{\phi(x)e^{-\omega^T\phi(x)}}{1+e^{-\omega^T\phi(x)}}\right)_{\omega^k} = (1-f_{\omega^k}(x^i))\,\phi(x^i)$$

Gradient descent for LR

1. No closed form solution to the cross-entropy loss

$$\widehat{\mathbf{w}}^{MLE} = \underset{\mathbf{w}}{\operatorname{argmin}} - \left[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \log f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) + \left(1 - y^{(i)} \right) \log \left(1 - f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) \right) \right) \right] \tag{4}$$

2. Apply gradient descent with $\mathbf{w}^{(k+1)} = \mathbf{w}^k - \eta \nabla E \left(\mathbf{w}^k \right)$
3. The descent update

$$-\eta \nabla E(\mathbf{w}) = -\eta \left[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \nabla \log f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) + \left(1 - y^{(i)} \right) \nabla \log \left(1 - f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) \right) \right) \right] \tag{5}$$

Gradient descent for LR

1. No closed form solution to the cross-entropy loss

$$\widehat{\mathbf{w}}^{MLE} = \underset{\mathbf{w}}{\text{argmin}} - \left[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \log f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) + \left(1 - y^{(i)} \right) \log \left(1 - f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) \right) \right) \right] \tag{4}$$

2. Apply gradient descent with $\mathbf{w}^{(k+1)} = \mathbf{w}^{k} - \eta \nabla E \left(\mathbf{w}^{k} \right)$
3. The descent update

$$-\eta \nabla E(\mathbf{w}) = -\eta \left[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \nabla \log f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) + \left(1 - y^{(i)} \right) \nabla \log \left(1 - f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) \right) \right) \right] \tag{5}$$

4. $\nabla f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) = \phi(\mathbf{x}^{(i)}) \left(\frac{e^{-(\mathbf{w})^T \phi(\mathbf{x}^{(i)})}}{1 + e^{-(\mathbf{w})^T \phi(\mathbf{x}^{(i)})}} \right)$
   $\Rightarrow$

Gradient descent for LR

1. No closed form solution to the cross-entropy loss

$$\widehat{\mathbf{w}}^{MLE} = \underset{\mathbf{w}}{\text{argmin}} - \left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (4)$$

2. Apply gradient descent with $\mathbf{w}^{(k+1)} = \mathbf{w}^{k} - \eta\nabla E\left(\mathbf{w}^{k}\right)$
3. The descent update

$$-\eta\nabla E(\mathbf{w}) = -\eta\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\nabla\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\nabla\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (5)$$

4. $\nabla f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) = \phi(\mathbf{x}^{(i)})\left(\frac{e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)$
$\Rightarrow$
5. $\nabla\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) = \phi(\mathbf{x}^{(i)})e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}\left(\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)^2$ and
$\nabla\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right) = -\phi(\mathbf{x}^{(i)})\left(\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)^2$

Descent update for LR

$$-\eta\nabla E(\mathbf{w}) = -\eta\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\nabla\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\nabla\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (6)$$

1. $\nabla\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) = \phi(\mathbf{x}^{(i)})e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}\left(\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)^2$ and
   $\nabla\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right) = -\phi(\mathbf{x}^{(i)})\left(\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)^2$
2. $\Rightarrow$ The final descent update is

Descent update for LR

$$-\eta\nabla E(\mathbf{w}) = -\eta\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\nabla \log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\nabla \log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] \tag{6}$$

1. $\nabla \log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) = \phi(\mathbf{x}^{(i)})e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}\left(\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)^2$ and
   $\nabla \log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right) = -\phi(\mathbf{x}^{(i)})\left(\frac{1}{1+e^{-(\mathbf{w})^T\phi(\mathbf{x}^{(i)})}}\right)^2$
2. $\Rightarrow$ The final descent update is

misclassification cost of $x^{(i)}$ higher if $y^{(i)}=1$ & $f_w(x^{(i)}) \to 0$ and vice verse

$$-\eta\nabla E(\mathbf{w}) = \eta\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\phi(\mathbf{x}^{(i)})\right] \tag{7}$$

like perceptron but weighted

Gradient descent for LR

1. The final descent update

$$-\eta\nabla E(\mathbf{w}) = \eta\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\phi(\mathbf{x}^{(i)})\right] \quad (8)$$

2. The iterative update rule: (Weka package implements gradient & also LBFGS)

$$\mathbf{w}^{(k+1)} = \mathbf{w}^k + \eta\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right)\right)\phi(\mathbf{x}^{(i)})\right] \quad (9)$$

3. Stochastic version of the same: (Mahout implements batch stochastic) on each data batch B_j

$$\mathbf{w}^{(k+1)} = \mathbf{w}^k + \eta\left(y^{(i)} - f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right)\right)\phi(\mathbf{x}^{(i)}) \quad (10)$$

$$w^{(k+1)} = w^k + \eta \frac{1}{B_j}\sum_{i \in B_j}\left(y^{(i)} - f_{w^k}(x^{(i)})\right)\phi(x^{(i)})$$

4. How would you contrast the updates with sigmoid (LR) against those with the step function (perceptron)?

Sigmoid (LR) vs. step function (perceptron)

1. Stochastic update for step fn (perceptron) with $y^{(i)} \in \{-1, 1\}$: Pick any example $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, for which *sign* $\left(\left(\mathbf{w}^{(k)}\right)^T \phi\left(\mathbf{x}^{(i)}\right)\right) \neq y^{(i)}$.

$$\mathbf{w}^{(k+1)} = \mathbf{w}^k + \eta y^{(i)} \phi(\mathbf{x}^{(i)}) \tag{11}$$

penalty based on sign mismatch

2. Stochastic update for sigmoid fn (LR) with $y^{(i)} \in \{0, 1\}$: Pick any example $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, for which $|f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right) - y^{(i)}| > 0.5$.

$$\mathbf{w}^{(k+1)} = \mathbf{w}^k + \eta \left(y^{(i)} - f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right)\right) \phi(\mathbf{x}^{(i)}) \tag{12}$$

$$\omega^{(k+1)} = \omega^k + \eta \sum \left(y^{(i)} - \left(\phi^T(x^i)\omega^k + b\right)\right)\phi(x^i)$$

penalty is based on extent of digression

3. Recall: (12) is also the stochastic update for linear regression! (12) is a characteristic update for **generalized linear models**[2] of which perceptron, linear regression and logistic are special cases.

$\hookrightarrow g(\omega^T\phi(x) + b)$

[2] https://en.wikipedia.org/wiki/Generalized_linear_model

Regularized LR and its Probabilistic Interpretation

1. The Regularized (Logistic) Cross-Entropy Loss function:

$$E(\mathbf{w}) = -\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2m}\|\mathbf{w}\|_2^2 \tag{13}$$

2. Motivations: Avoiding overfitting by discouraging large values of w_j for every j.
3. Probabilistic Explanation?

$\downarrow$ To help generalize well to new data points

$Pr(y=1/\phi(x)) = \frac{1}{1+e^{-w^T\phi(x)}}$

Need: $Pr(w)$ . . . s.t $Pr(w|D) \propto Pr(D|w)Pr(w) = L(w|D)Pr(w)$

has similar form as $Pr(w)$

Regularized LR and its Probabilistic Interpretation

1. The Regularized (Logistic) Cross-Entropy Loss function:

$$E(\mathbf{w}) = -\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2m}\|\mathbf{w}\|_2^2 \tag{13}$$

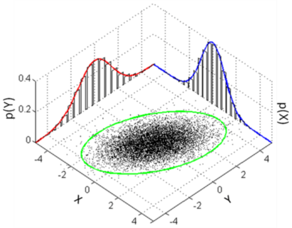
2. Motivations: Avoiding overfitting by discouraging large values of w_j for every j.
3. Probabilistic Explanation? A Bayesian Posterior probabilistic explanation to regularized LR (next)
4. We will reinvoke Bayesian (Parameter) Estimation

$$P_r(w) = \mathcal{N}\left(0, \frac{1}{\lambda} I\right) \Leftrightarrow \underbrace{w_i \in}_{\text{roughly.}} \left[\frac{-3}{\sqrt{\lambda}}, \frac{+3}{\sqrt{\lambda}}\right]$$

Bayesian Inference For Logistic Regression

MAP Estimation and regularized LR

1. Recall the multivariate Gaussian (Normal) Distribution:
   $\mathcal{N}(\mathbf{w};\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{m}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{w}-\mu)^T\Sigma^{-1}(\mathbf{w}-\mu)}$ when $\Sigma \in \Re^{m\times m}$ is positive-definite and $\mu \in \Re^m$



2. Suppose we want each $|w_i|$ to be bounded roughly by $\pm\frac{3}{\lambda}$
3. Then by the $3-\sigma$ rule we let $\mathbf{w} \sim \mathcal{N}(\mathbf{w}; 0, \frac{1}{\lambda}I)$ where I is an $m \times m$ identity matrix
4. $\Rightarrow \Pr(\mathbf{w}) = \frac{1}{\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2}$

· Write expression for $Pr(D|w)$ & then derive $Pr(w|D) \propto Pr(D|w)\, Pr(w)$

$\log(Pr(w|D)) = \log(Pr(D|w)) + \log(Pr(w)) = \text{Cross entropy} + \frac{\lambda}{2}\|w\|_2^2$

MAP estimation and regularized LR

1. $\Pr(\mathbf{w}) = \frac{1}{\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2}$
2. Recall the MLE for LR: $\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} L(\mathcal{D}; \mathbf{w})$

$$= \operatorname*{argmax}_{\mathbf{w}} \prod_{i=1}^{m} \left(f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) \right)^{y^{(i)}} \left(1 - f_{\mathbf{w}} \left(\mathbf{x}^{(i)} \right) \right)^{1-y^{(i)}}$$

3. Now the MAP for LR: $\tilde{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} \Pr(\mathbf{w}) L(\mathcal{D}; \mathbf{w}) =$

$$\operatorname*{argmax}_{w} \log \Pr(w) + \log L(D, w)$$

MAP estimation and regularized LR

1. $\Pr(\mathbf{w}) = \frac{1}{(\frac{2\pi}{\lambda})^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2}$
2. Recall the MLE for LR: $\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} L(\mathcal{D}; \mathbf{w})$
$$= \operatorname*{argmax}_{\mathbf{w}} \prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}}$$
3. Now the MAP for LR: $\tilde{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} \Pr(\mathbf{w}) L(\mathcal{D}; \mathbf{w}) =$
$$\operatorname*{argmax}_{\mathbf{w}} \frac{1}{\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2} \underbrace{\prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{(1-y^{(i)})}}$$

$$= \operatorname*{argmax}_{w} \log\left(e^{-\lambda/2 \|w\|^2}\right) - m \text{ Cross Entropy} \cancel{- m/2 \log(2\pi/\lambda)}$$

Can be ignored

$$= \operatorname*{argmax}_{w} -m \text{ Cross Entropy} - \frac{\lambda}{2}\|w\|^2$$

MAP estimation and regularized LR

1. **FROM** MAP for LR: $\tilde{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}} \Pr(\mathbf{w}) L(\mathcal{D}, \mathbf{w})$

$$= \underset{\mathbf{w}}{\text{argmax}} \frac{1}{\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2} \prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}}$$

.....Taking $-\frac{1}{m}\log(.)$ transformation,

2. **TO** Min of the Regularized Logistic (Cross-Entropy) Loss function:

$$\tilde{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} - \left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} \log f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \qquad (14)$$

where we have ignored $-\frac{1}{m}\log\left(\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}\right)$ since this term is independent of $\mathbf{w}$.

.....Thus, MAP $\tilde{\mathbf{w}}$ can be found by minimizing the *Regularized Cross Entropy Error*

Tut 7: Posterior $\Pr(w|D)$ is Gaussian!

Gradient descent for Regularized LR

Gradient descent for Regularized LR

1. The final descent update

$$-\eta \nabla E(\mathbf{w}) = \eta \left[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right) \right) \phi(\mathbf{x}^{(i)}) - \underline{\lambda \mathbf{w}} \right] \tag{15}$$

2. The iterative update rule:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{k} + \eta \left[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right) \right) \phi(\mathbf{x}^{(i)}) - \underline{\lambda \mathbf{w}^k} \right] \tag{16}$$

shrinking

3. Stochastic version of the same:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{k} + \eta \left(y^{(i)} - f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right) \right) \phi(\mathbf{x}^{(i)}) - \eta \lambda \mathbf{w}^k \tag{17}$$

Extension to multi-class logistic

1. Each class $c = 1, 2, \ldots, K-1$ can have a different weight vector $[\mathbf{w}_{c,1}, \mathbf{w}_{c,2}, \ldots, \mathbf{w}_{c,k}, \ldots, \mathbf{w}_{c,K-1}]$ and

$$p(Y = c|\phi(\mathbf{x})) = \frac{e^{-(\mathbf{w}_c)^T\phi(\mathbf{x})}}{1 + \sum_{k=1}^{K-1} e^{-(\mathbf{w}_k)^T\phi(\mathbf{x})}}$$

for $c = 1, \ldots, K-1$ so that

$$p(Y = K|\phi(\mathbf{x})) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{-(\mathbf{w}_k)^T\phi(\mathbf{x})}}$$

Alternative (equivalent) extension to multi-class logistic

1. Each class $c = 1, 2, \ldots, K$ can have a different weight vector $[\mathbf{w}_{c,1}, \mathbf{w}_{c,2} \ldots \mathbf{w}_{c,p}]$ and

$$p(Y = c|\phi(\mathbf{x})) = \frac{e^{-(\mathbf{w}_c)^T\phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{-(\mathbf{w}_k)^T\phi(\mathbf{x})}}$$

for $c = 1, \ldots, K$.