Lecture 19: Logistic Regression and Regularization, Kernelized Logistic Regression, Neural Networks

Instructor: Prof. Ganesh Ramakrishnan

Recap: Minimizing negative Log-likelihood for LR

1. 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 $\Pr(y|\mathbf{w}, \phi(.))$, rather than the 'true' distribution $\Pr(y|\mathcal{D})$.

$$H(P(D)) \leq E(\mathbf{w}) = \mathbf{E}_{\Pr(y|\mathcal{D})}\left[-\log \Pr(y|\mathbf{w}, \phi(.))\right] \quad (1)$$

$f_w(x^i) = \Pr(y=1|x^{(i)}) = \frac{1}{1+\exp(-w^T\phi(x^{(i)}))}$

2. The Cross-entropy Loss function:

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

(Slides have gradient descent in this form)

with some simplification,

$$E(\mathbf{w}) = -\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\mathbf{w}^T\phi(\mathbf{x}^{(i)}) - \log\left(1+exp\left(\mathbf{w}^T\mathbf{x}^{(i)}\right)\right)\right)\right] \quad (3)$$

Hint: Useful for Tut 7

(used for grad descent in last class)

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

Recap: Gradient descent for LR

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

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

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

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

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

Recap: 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

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

Recap: Gradient descent for LR

1. The final descent update

LBFGS etc approximate $(\nabla^2 E)^{-1}$

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

Perceptron $\sum_{i \in M} y^{(i)} \phi(x^{(i)})$

(Weka has both implementations)

2. The iterative update rule:

$w^{(k+1)} = w^{(k)} - (\nabla^2 E)^{-1} \nabla E$

Newton update (faster to converge)

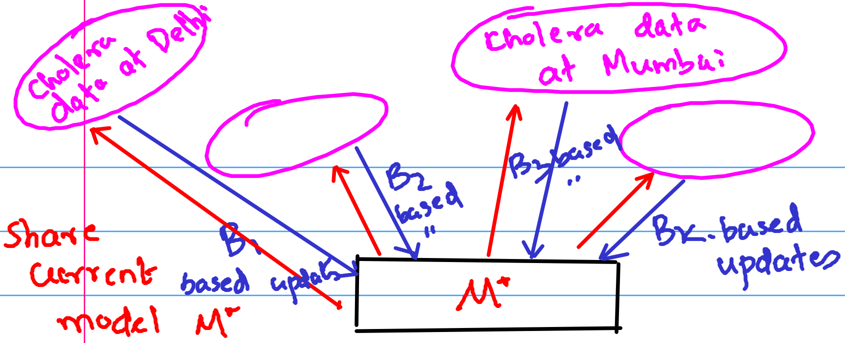
$$\mathbf{w}^{(k+1)} = \mathbf{w}^k + \eta \left[\frac{1}{m} \underbrace{\sum_{i=1}^{m}}_{\text{Deterministically over all pts}} \underbrace{\left(y^{(i)} - f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right) \right)}_{\text{Accounting for extent of misclassification!}} \phi(\mathbf{x}^{(i)}) \right] \tag{9}$$

3. Stochastic version of the same:

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

Batch stochastic: $w^{(k+1)} = w^{(k)} + \eta \frac{1}{|B|} \sum_{B} - - - - -$

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



Mahout: Distributed machine learning platform. (performs batch stochastic for LR)

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* $\left(\left(\mathbf{w}^{(k)}\right)^T \phi\left(\mathbf{x}^{(i)}\right)\right) \neq y^{(i)}$.

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

Direct contrast with perceptron

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 $\left|f_{\mathbf{w}^k}\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right| > 0.5$.

Prob of error > 0.5 minimizes risk (Bayes risk)

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

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. $\rightarrow g(w^T\phi(x) + b)$

For nonparam models g is a fn of x

dx

[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?

ie Bayesian interpretation?

$$Pr(w_i) = \mathcal{N}\left(0, \frac{1}{\lambda}\right)$$

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

Bayesian Inference For **Logistic Regression**

MAP Estimation and regularized LR

$$-\log P(w) = +\log\left(\frac{2\pi}{\lambda}\right)^{m/2} + \frac{\lambda}{2}\|w\|_2^2$$

can be ignored (independent of w)

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 $\pm\frac{3}{\sqrt{\lambda}}$
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. $\Rightarrow \Pr(\mathbf{w}) = \frac{1}{\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2}$

Now derive MAP estimate for w

$$w_{MAP} = \arg\max_w \Pr(w|D) = \arg\max_w \frac{\Pr(w)\Pr(D|w)}{\cancel{\Pr(D)}} = \arg\max_w \Pr(w)L(w;D)$$

$$= \arg\max_w \log\Pr(w) + \log(L(w;D)) = \arg\min_w \left(\underbrace{-\log\Pr(w) - \log(L(w;D))}_{m}\right)$$

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: $\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, L(\mathcal{D};\mathbf{w})$

   $$= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}}$$
3. Now the MAP for LR: $\tilde{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, \Pr(\mathbf{w}) L(\mathcal{D};\mathbf{w}) = \underset{w}{\operatorname{argmax}} \frac{1}{m}\left(\log \Pr(w) + \log L(D, w)\right)$

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: $\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, L(\mathcal{D};\mathbf{w})$

   $$= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{1-y^{(i)}}$$
3. Now the MAP for LR: $\tilde{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \Pr(\mathbf{w}) L(\mathcal{D};\mathbf{w}) =$

   $$\underset{\mathbf{w}}{\operatorname{argmax}} \underbrace{\frac{1}{\left(\frac{2\pi}{\lambda}\right)^{\frac{m}{2}}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|_2^2}}_{\Pr(\omega)} \underbrace{\prod_{i=1}^{m} \left(f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_{\mathbf{w}}\right.}_{L(D;\omega)}\left.\left(\mathbf{x}^{(i)}\right)\right)^{(1-y^{(i)})}$$

MAP estimation and regularized LR

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

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

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

2. **TO** Min of the <u>Regularized cross entropy</u>

$$= \underset{w}{\operatorname{argmin}} -\frac{1}{m}\sum y^{(i)} \log f_w(x^{(i)}) + (1-y^{(i)}) \log(1-f_w(x^{(i)}))$$

$$+ \frac{\lambda}{2m}\|w\|^2$$

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:

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

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*

Gradient descent update :

Gradient descent for Regularized LR

$$w^{(k+1)} = w^{(k)} + \frac{\eta}{m} \sum_i \left(y^{(i)} - f_w(x^{(i)})\right) \phi(x^{(i)}) - \frac{\lambda \eta}{m} w^{(k)}$$

Gradient descent for Regularized LR

1. The final descent update

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

2. The iterative update rule:

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

3. Stochastic version of the same:

Inspires other models of update that work well in practice

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

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

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

Reserved class for K^{th}

For 2 class case we used $K=1$

Assuming: Classification based on $\arg\max_k \Pr(Y=k|\phi(x))$

① Why not $w_1 \ldots w_K$ (K vectors?)

② What is geometric interpretation?

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

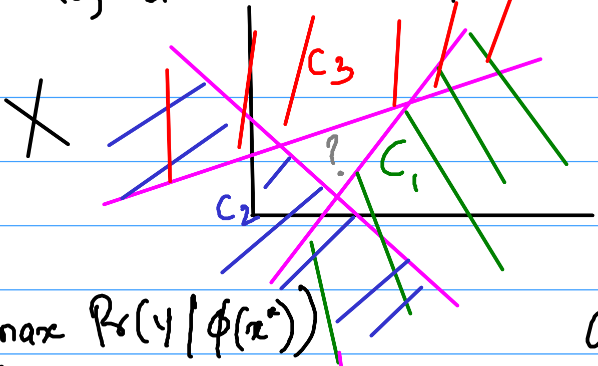
$$p(Y = c|\phi(\mathbf{x})) = \frac{e^{-(\mathbf{w}_c)^T \phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{-(\mathbf{w}_k)^T \phi(\mathbf{x})}}$$

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

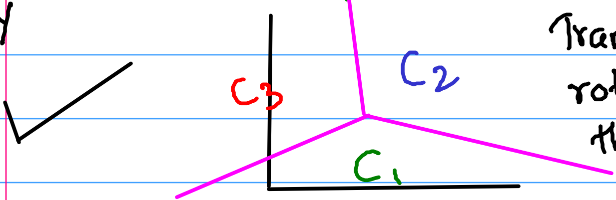
Ans to Q1: The two are equivalent (Tut 7). For grad descent in the second formulation I update K vectors & ∴ more numerical error than updating K-1 vectors which are "differences" of w_c with w_K (Tut 7)

Geometric interpretation?
(of decision on test pt)

Not 1-vs rest



$$y^* = \underset{y}{\arg\max} \Pr(y \mid \phi(x^*))$$



Gradient descent: Translation (for b) and rotation of limbs of the starfish!

Logistic Regression Kernelized

1. We have already seen (a) Cross Entropy loss and (b) Bayesian interpretation for regularization
2. The Regularized (Logistic) Cross-Entropy Loss function (minimized wrt $\mathbf{w} \in \Re^p$):

$$E(w) = -\left[\underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_w\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)}_{\sum_i L(\omega,\, x^{(i)},\, y^{(i)})}\right] + \frac{\lambda}{2m}\|\mathbf{w}\|_2^2 \tag{18}$$

$\|\omega\|_2^2 = \langle \omega, \omega \rangle$

3. Equivalent dual kernelized objective[3] (minimized wrt $\alpha \in \Re^m$):

$$E(\omega) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log\left(\frac{1}{1+\exp\sum_j \alpha_j K(x^{(i)}, x^{(j)})}\right) + (1-y^{(i)})\log\left(\ldots\ldots\right)\right) + \lambda/2m\|w\|^2 ?$$

$$f(x^i) = \frac{1}{1+\sum_j \alpha_j K(x, x^j)}$$

$$\omega^T\phi(x) \Rightarrow \sum_j \alpha_j K(x, x^{(j)})$$

[3] Representer Theorem and http://perso.telecom-paristech.fr/~clemenco/Projets_ENPC_files/kernel-log-regression-svm-boosting.pdf

Logistic Regression Kernelized

1. We have already seen (a) Cross Entropy loss and (b) Bayesian interpretation for regularization
2. The Regularized (Logistic) Cross-Entropy Loss function (minimized wrt $\mathbf{w} \in \Re^p$):

$$E(w) = -\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log f_w\left(\mathbf{x}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\mathbf{w}}\left(\mathbf{x}^{(i)}\right)\right)\right)\right] + \frac{\lambda}{2m}\|\mathbf{w}\|_2^2 \quad (18)$$

3. Equivalent dual kernelized objective[3] (minimized wrt $\alpha \in \Re^m$):

$$E_D(\alpha) = \left[\sum_{i=1}^{m}\left(\sum_{j=1}^{m} -y^{(i)}K\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right)\alpha_j + \frac{\lambda}{2}\alpha_j K\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right)\alpha_j\right) + \log\left(1+\sum_{j=1}^{m}\alpha_j K\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right)\right)\right] \quad (19)$$

How?

Decision function $f_{\mathbf{w}}(\mathbf{x}) = \dfrac{1}{1+\exp\left(\underbrace{\sum_{j=1}^{m}\alpha_j K\left(\mathbf{x},\mathbf{x}^{(j)}\right)}_{w^T\phi(x)}\right)}$

[3] Representer Theorem and http://perso.telecom-paristech.fr/~clemenco/Projets_ENPC_files/kernel-log-regression-svm-boosting.pdf

Some Tutorial 7 Questions

- Prove that the Kernelized Logistic Regression form is equivalent to the original Logistic Regression minimum regularized cross entropy form: 2 Hints
- Show equivalence of the two formulations of Multiclass Logistic Regression.

Logistic Regression Generalized to CRF

1. Multi-class LR: $c \in [1 \dots K]$ has weight vector $[w_{c,1} \dots w_{c,p}]$

$$\Pr(y = c \mid x) = \frac{e^{-w_c^T \phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{-w_k^T \phi(\mathbf{x})}} = \frac{e^{-\tilde{\omega}^T \tilde{\phi}(y,x)}}{\sum_{k=1}^{K} e^{-\tilde{\omega}^T \tilde{\phi}(k,x)}}$$

(0) Construct $\tilde{\omega}$ & $\tilde{\phi}(\cdot)$ . . .

(1) Made $\tilde{\omega}$ independent of class c

(2) Expanded ϕ to $\tilde{\phi}(y,x)$ & take class as argument

Useful when y has some structure / periodicity etc!

[4] http://www.tzi.de/~edelkamp/lectures/ml/scripts/loglinearcrfs.pdf

Logistic Regression Generalized to CRF

1. Multi-class LR: $c \in [1 \dots K]$ has weight vector $[w_{c,1} \dots w_{c,p}]$

$$\Pr(y = c \mid x) = \frac{e^{-w_c^T \phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{-w_k^T \phi(\mathbf{x})}} = \frac{e^{-\tilde{w}^T \phi(\mathbf{x}, y=c)}}{Z(\mathbf{x}, \tilde{w})}$$

   where $\tilde{w} = [w_{1,1} \dots w_{1,p}, \dots w_{c,1} \dots w_{c,p}, \dots w_{K,1} \dots w_{K,p}]$ and $\phi(\mathbf{x}, y) = [\delta(y, 1)\phi(\mathbf{x}), \dots, \delta(y, c)\phi(\mathbf{x}) \dots \delta(y, K)\phi(\mathbf{x})]$ and $\delta(a, b) = 1$ if $a = b$ and $= 0$ otherwise

2. Extended to non-iid inference in Conditional Random Fields[4] with $\mathbf{x} = [\mathbf{x}^{(1)} \dots \mathbf{x}^{(n)}]$ and $\mathbf{y} = [y^{(1)} \dots y^{(n)}]$:

[4] http://www.tzi.de/~edelkamp/lectures/ml/scripts/loglinearcrfs.pdf

Logistic Regression Generalized to CRF

1. Multi-class LR: $c \in [1 \dots K]$ has weight vector $[w_{c,1} \dots w_{c,p}]$

$$\Pr(y = c \mid x) = \frac{e^{-w_c^T \phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{-w_k^T \phi(\mathbf{x})}} = \frac{e^{-\tilde{w}^T \phi(\mathbf{x}, y=c)}}{Z(\mathbf{x}, \tilde{w})}$$

   where $\tilde{w} = [w_{1,1} \dots w_{1,p}, \dots w_{c,1} \dots w_{c,p}, \dots w_{K,1} \dots w_{K,p}]$ and $\phi(\mathbf{x}, y) = [\delta(y,1)\phi(\mathbf{x}), \dots, \delta(y,c)\phi(\mathbf{x}) \dots \delta(y,K)\phi(\mathbf{x})]$ and $\delta(a,b) = 1$ if $a = b$ and $= 0$ otherwise

2. Extended to non-iid inference in Conditional Random Fields[4] with $\mathbf{x} = [\mathbf{x}^{(1)} \dots \mathbf{x}^{(n)}]$ and $\mathbf{y} = [y^{(1)} \dots y^{(n)}]$:

$$\Pr(\mathbf{y} \mid \mathbf{x}) = \frac{e^{-\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})}}{Z(\mathbf{x}, \mathbf{w})}$$

[4] http://www.tzi.de/~edelkamp/lectures/ml/scripts/loglinearcrfs.pdf