Probabilistic discriminative models

Fixed basis functions

Logistic regression

Iterative reweighted least squares

Multiclass logistic regression

Probit regression

Two approaches to find parameters of generalized linear model

- A) Generative and Indirect (already seen)
  - Fit class conditionals and class priors separately (logistic sigmoid/ softmax, maximum likelihood)
  - Apply Bayes theorem
- From the model created, synthetic data can be generated by taking values of from the marginal distribution p(x)
- B) Discriminative and Direct
  - Maximize a likelihood function defined through the conditional distribution $p(C_k/x)$ <--this is a functional form of generalized linear model
  - Normally lesser parameters
  - Might give better prediction
  - Algorithm is called "Iterative reweighted least squares (IRLS)"

Fixed basis functions

- Fixed basis function transformation: Make a fixed nonlinear transformation of inputs using a vector of basis functions $\phi(\mathbf{x})$
- Advantage: Decision boundaries will be linear in the feature space ϕ corresponding to the non-linear decision boundaries in original space
- Handle overlap between class conditional densities ($p(c_k/x)$ are not all 0 or 1) by proper choice of nonlinearity

Logistic regression

- Actually this is a Classification technique
- Posterior probability of class C_1 = logistic sigmoid (linear function of feature vector ϕ)

$$p(C_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right)$$

- $\sigma(\cdot)$ is the logistic sigmoid function (statistics -> logistic regression)
- M dimensional feature space has only M parameters (much less than Gaussian – 2M for mean and M(M+1)/2 for covariance)
- Use Maximum likelihood to find parameter values:
  - Take derivative of logistic sigmoid function: $\frac{d\sigma}{da} = \sigma(1-\sigma)$

- For data set $\{\phi_n, t_n\}$ likelihood function is $p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} \{1 - y_n\}^{1-t_n}$

$$y_n = p(\mathcal{C}_1|\phi_n)$$

- Error function is negative log of likelihood -> cross entropy error fn.

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

$$y_n = \sigma(a_n) \text{ and } a_n = \mathbf{w}^{\mathrm{T}} \phi_n$$

- (using $\frac{d\sigma}{da} = \sigma(1 - \sigma)$,the derivative of logistic sigmoid is cancelled)

- Gradient of error function w.r.t 'w' -> $\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n$

- In words-> contribution by a data point n to the gradient is error y_n- t_n multiplied by basis function vector ϕ_n
- Similar to gradient for sum of squares error function for linear regression
- Note: For linearly separable data sets, maximum likelihood can result in overfitting.
-

Iterative reweighted least squares

- Issue: Non-linearity of logistic sigmoid function => no closed form of solution for logistic regression
- But difference from a quadratic form is only a little.
- Error function here is concave => unique minimum
- Minimize error function using Newton Raphson iteration optimization (this uses local quadratic approximation to log likelihood function)
- Newton Raphson update to minimize error E(w) is

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$$

- H is Hessian matrix (elements are second derivative of E(w) w.r.t 'w'
- -------------
- Applying Newton Raphson to linear regression.
- (Φ is NxM design matrix -> n^{th} row is ϕ_n^{T})
- Gradient of error function: $\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}}\phi_n - t_n)\phi_n = \Phi^{\mathrm{T}}\Phi\mathbf{w} - \Phi^{\mathrm{T}}\mathbf{t}$
- Hessian of error function: $\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \phi_n \phi_n^{\mathrm{T}} = \Phi^{\mathrm{T}}\Phi$
- Newton Raphson update: $\mathbf{w}^{(\mathrm{new})} = \mathbf{w}^{(\mathrm{old})} - (\Phi^{\mathrm{T}}\Phi)^{-1}\left\{\Phi^{\mathrm{T}}\Phi\mathbf{w}^{(\mathrm{old})} - \Phi^{\mathrm{T}}\mathbf{t}\right\}$
- (Quadratic, Std.least sq sol) = $(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{t}$

- <u>Apply Newton Raphson to cross entropy fn. for logistic regression</u>
- Use $\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n$ and $\frac{d\sigma}{da} = \sigma(1-\sigma)$
- Gradient of this is $\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t})$
- Hessian is $\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^{\mathrm{T}} =$
- $= \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi}$
- R is NxN diagonal matrix with elements $R_{nn} = y_n(1-y_n)$
- Hessian depends on 'w' through R

- As $0<y_n<1$ (for logistic sigmoid function), for any vector 'u'
- $u^T H u > 0$ => Hessian matrix is positive definite =>
- Error function is concave function of 'w'
-     Error function => has an unique minimum
- Newton Raphson update model for logistic regression is

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - (\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\Phi^{\mathrm{T}}(\mathbf{y} - \mathbf{t})$$

- Make z = N dimensional vector with elements
- $$\mathbf{z} = \Phi\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t})$$
- $$\mathbf{w}^{(\text{new})} = (\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{R}\mathbf{z}$$

- As Weighing matrix R depends on a changing 'w' the equation
- $(\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{R}\mathbf{z}$ has to be repeatedly applied, with new
- values of weight vector <---- Iterative reweighted least squares (IRLS)

- Elements of R give the variance

- Mean of 't' in logistic regression: $\mathbb{E}[t] = \sigma(\mathbf{x}) = y$

- Variance of 't' : $\mathrm{var}[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 = \sigma(\mathbf{x}) - \sigma(\mathbf{x})^2 = y(1-y)$

Multi class Logistic regression

- Use maximum likelihood to calculate values of 'w'

1) Find derivatives of y_k w.r.t all activations a_j

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$$

I_{kj} identity matrix

2) Likelihood function (using 1 of k) (T is n x k matrix of target variables

$$p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathcal{C}_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

3) Negative logarithm -> $E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk}$

This is the cross entropy error function for multi class classification

,,,,,contd.....

- 4) Take gradient of error function w.r.t 'w_j'
- Using $\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$ and $\sum_k t_{nk} = 1$
- We get $$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\, \phi_n$$
- Above is similar to linear regression -> error multiplied by basis function
- Use Newton Raphson method to get IRLS algorithm for multiclass

Probit Regression

- There are class-conditional distributions that cannot be modelled by logistic sigmoid or by softmax.
- For example when the class conditional distribution is modelled using a Gaussian <u>mixture</u>
- Find other discriminative probabilistic model ->
- With $a = \mathbf{w}^{\mathrm{T}}\phi$ and f(.) as activation function

$$p(t = 1|a) = f(a)$$

- Find a noisy threshold value as: For each i/p ϕ_n, evaluate $a_n = w^T \phi_n$,
- Set target value t_n= 1 if a_n >= θ else t_n=0

- If value of θ is from a probability density p(θ) then corresponding activation function is the cumulative distribution fn.

$$f(a) = \int_{-\infty}^{a} p(\theta)\,\mathrm{d}\theta$$

- <u>Probit function:</u>
- When the density p(θ) is a zero mean, unit variance Gaussian, the cumulative distribution (activation function) is

$$\Phi(a) = \int_{-\infty}^{a} \mathcal{N}(\theta|0,1)\,\mathrm{d}\theta$$

- Has a Sigmoidal shape

- Generalized linear model based on probit activation function is Probit regression
- ---
- Related function is "erf function"

$$\mathrm{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a \exp(-\theta^2/2)\,\mathrm{d}\theta$$

- ---
- Outliers: Probit model is more sensitive to Outliers <-- Bad behavior