CS 5135/6035 Learning Probabilistic Models

Lecture 9: Multivariate Gaussian MLE, Logistic Regression, Newton's Method

Gowtham Atluri

September 20, 2018

Reading Material:

- Jordan, Chapter 13. The Multivariate Gaussian

https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter13.pdf

- Engelhardt, Gaussian Models

https://www.cs.princeton.edu/~bee/courses/scribe/lec_09_09_2013.pdf

- Shalizi, Chapter 12 Logistic Regression

https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf

Learning a MV Gaussian using Maximum Likelihood

- **Scenario:** Height (in cm.) and weight (in kg.) of 200 individuals are collected. Assuming they follow a MV Gaussian distribution, estimate the parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the MV Gaussian.

Row	Weight	Height
1	77.4	182.6
2	58.5	161.3
3	63.1	161.2
4	68.6	177.7
5	59.3	157.8
6	76.7	170.4

Learning a MV Gaussian using Maximum Likelihood

- **Scenario:** Height (in cm.) and weight (in kg.) of 200 individuals are collected. Assuming they follow a MV Gaussian distribution, estimate the parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the MV Gaussian.

- Given a training data $\mathcal{X} = \{\mathbf{x_1}, \dots, \mathbf{x_n}\}$ drawn *i.i.d* from a Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$,

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \frac{1}{\sqrt{det(2\pi\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(x-\mu)^T \boldsymbol{\Sigma}^{-1}(x-\mu)}$$

What are the parameters with which a set of data points $\mathcal{X}$ were generated from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$?

Log-Likelihood for MV Gaussian

- First, we write the *likelihood* $p(\mathcal{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\mathbf{x_1}, \dots, \mathbf{x_n}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_i p(\mathbf{x_i}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (from *i.i.d*)
- We write the log-likelihood $\log p(\mathcal{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_i \log p(\mathbf{x_i}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- We know the pdf for each data point $\mathbf{x_i}$ is

$$p(\mathbf{x_i}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{det(2\pi\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(x_i-\mu)^T \boldsymbol{\Sigma}^{-1}(x_i-\mu)}$$

- As the log-likelihood is a function of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ we denote it as $\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
-

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \sum_{i=1}^{n} \log p(\mathbf{x_i}|\boldsymbol{\mu}\boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x_i} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_i} - \boldsymbol{\mu}) - \frac{n}{2}\log det(2\pi\boldsymbol{\Sigma})$$

MLE for MV Gaussian - determining μ

- Log-likelihood

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \sum_{i=1}^{n} \log p(\mathbf{x_i}|\boldsymbol{\mu}\boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x_i} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_i} - \boldsymbol{\mu}) - \frac{n}{2}\log det(2\pi\boldsymbol{\Sigma})$$

- To find **optimal** $\boldsymbol{\mu}$, take the partial derivative w.r.t. vector $\boldsymbol{\mu}$

$$\nabla_{\mu}\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{i=1}^{n}(-2)\boldsymbol{\Sigma}^{-1}(\mathbf{x_i} - \boldsymbol{\mu}) = \sum_{i=1}^{n}\boldsymbol{\Sigma}^{-1}(\mathbf{x_i} - \boldsymbol{\mu})$$

- Equating this to zero and solve for μ

$$\sum_{i=1}^{n}\boldsymbol{\Sigma}^{-1}(\mathbf{x_i} - \boldsymbol{\mu}) = 0$$

MLE for MV Gaussian - determining μ

- Equating this to zero and solve for μ

$$\sum_{i=1}^{n} \mathbf{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = 0$$

$$\sum_{i=1}^{n} \mathbf{\Sigma}^{-1}\mathbf{x}_i - \sum_{i=1}^{n} \mathbf{\Sigma}^{-1}\boldsymbol{\mu} = 0$$

$$\sum_{i=1}^{n} \mathbf{\Sigma}^{-1}\mathbf{x}_i - n\boldsymbol{\mu}\mathbf{\Sigma}^{-1} = 0$$

$$\sum_{i=1}^{n} \mathbf{\Sigma}^{-1}\mathbf{x}_i = n\boldsymbol{\mu}\mathbf{\Sigma}^{-1}$$

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

MLE for MV Gaussian - determining Σ

- To determine **optimal $\mathbf{\Sigma}$**...
- $$\ell = -\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) - \frac{n}{2}\log det(2\pi\mathbf{\Sigma})$$
- It is convenient to isolate $\mathbf{\Sigma}^{-1}$

$$\ell = -\frac{1}{2}trace\Big(\mathbf{\Sigma}^{-1}\underbrace{\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T}_{\equiv \mathbf{M}}\Big) + \frac{n}{2}\log det(2\pi\mathbf{\Sigma}^{-1})$$

$\mathrm{trace}\,(A) = \sum_i a_{ii}$

- The log-likelihood now is

$$\ell = -\frac{1}{2}trace\Big(\mathbf{\Sigma}^{-1}\mathbf{M}\Big) + \frac{n}{2}\log det(2\pi\mathbf{\Sigma}^{-1})$$

MLE for MV Gaussian - determining Σ

- The log-likelihood now is

$$\ell = -\frac{1}{2}trace\Big(\mathbf{\Sigma}^{-1}\mathbf{M}\Big) + \frac{n}{2}\log det(2\pi\mathbf{\Sigma}^{-1})$$

- To find **optimal $\mathbf{\Sigma}$**, take partial derivative w.r.t matrix $\mathbf{\Sigma}^{-1}$
- Trace and matrix derivatives: $\nabla_A tr(AB) = B^T$; $\nabla_A \log|A| = A^{-T}$
- using $\mathbf{M} = \mathbf{M}^T$, we obtain

$$\nabla_{\mathbf{\Sigma}^{-1}}\ell(\boldsymbol{\mu}, \mathbf{\Sigma}) = -\frac{1}{2}\mathbf{M} + \frac{n}{2}\mathbf{\Sigma}$$

- Equating this to zero matrix and solving for $\mathbf{\Sigma}$ gives the sample covariance

$$\mathbf{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

Comparing Univariate and MV Gaussian ML estimates

Multivariate Gaussian

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

Univariate Gaussian

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}x_i$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

Logistic Regression

- Example
- Problem definition
- Assumption
- Conditional Likelihood
- Maximizing Conditional Likelihood

[http://www.stat.cmu.edu/~cshalizi/uADA/13/lectures/ch12.pdf]

Logistic Regression: Example

- Widely used to model outcome of a categorical *dependent* variable, given the state of continuous *independent* variables
- Petal length of flowers from two different plant species are collected.

Row	PetalLength	Species
1	1.6	setosa
2	1.4	setosa
3	1.3	setosa
4	5.2	virginica
5	5.0	virginica
6	5.2	virginica

- Dependent variable
  - Species
- Independent variable
  - PetalLength

- Determine the probabilities:

$$p(y = setosa|x = 1.5) = ? \qquad p(y = virginica|x = 1.5) = ?$$

Logistic Regression: Problem definition

Problem 1: Univariate x_i and Binary y

- Given a training set $\{(x_i, y_i) : i = 1, 2, \ldots n\}$, $y_i \in \{0, 1\}$, and $x_i \in \mathbb{R}^1$
- Define $p(y = 0|x_i)$ and $p(y = 1|x_i)$

Logistic Regression: Problem definition

Problem 1: Univariate x_i and Binary y

- Given a training set $\{(x_i, y_i) : i = 1, 2, \ldots n\}$, $y_i \in \{0, 1\}$, and $x_i \in \mathbb{R}^1$
- Define $p(y = 0|x_i)$ and $p(y = 1|x_i)$

Problem 2: Multivariate $\boldsymbol{x}_i$ and Binary y

- Given a training set $\{(\boldsymbol{x}_i, y_i) : i = 1, 2, \ldots n\}$, $y_i \in \{0, 1\}$, and $\boldsymbol{x}_i \in \mathbb{R}^d$
- Define $p(y = 0|\boldsymbol{x}_i)$ and $p(y = 1|\boldsymbol{x}_i)$

Problem 3: Multivariate $\boldsymbol{x}_i$ and Categorical y

- Given a training set $\{(\boldsymbol{x}_i, y_i) : i = 1, 2, \ldots n\}$, $y_i \in \{1, 2, \ldots k\}$, and $\boldsymbol{x}_i \in \mathbb{R}^d$
- Define $p(y = 1|\boldsymbol{x}_i), \ldots p(y = k|\boldsymbol{x}_i)$

Logistic regression: Assumption (for univariate x)

- For a binary variable y (i.e., a Bernoulli outcome) and x a continuous varibale, we assume

$$p(y = 1|x, \beta_0, \beta_1) = \sigma(\beta_0 + \beta_1 x) = \frac{1}{1 + exp - [\beta_0 + \beta_1 x]}$$

  where
  - β_0, β_1 are parameters
  - $\sigma(z) = 1/(1 + e^{-z})$ is a nonlinear, sigmoid function
- This model is called **logistic regression**

Logistic regression: Assumption (for univariate x)

- For a binary variable y (i.e., a Bernoulli outcome) and x a continuous varibale, we assume

$$p(y = 1|x, \beta_0, \beta_1) = \sigma(\beta_0 + \beta_1 x) = \frac{1}{1 + exp - [\beta_0 + \beta_1 x]}$$

  where
  - β_0, β_1 are parameters
  - $\sigma(z) = 1/(1 + e^{-z})$ is a nonlinear, sigmoid function
- This model is called **logistic regression**

Logistic regression: Assumption (for univariate x)

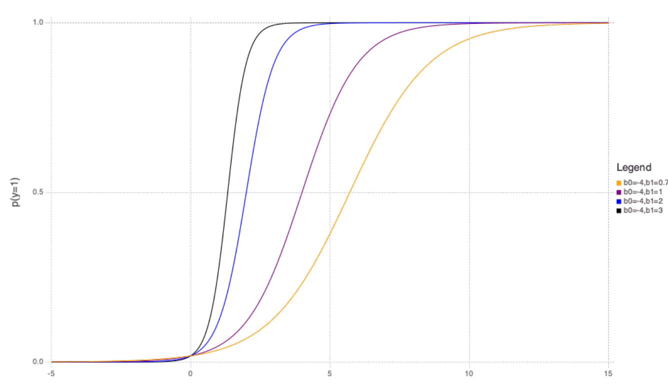
$$p(y = 1|x, \beta_0, \beta_1) = \sigma(\beta_0 + \beta_1 x) = \frac{1}{1 + exp - [\beta_0 + \beta_1 x]}$$

- Alternatively

$$\log \frac{p}{1 - p} = \beta_0 + \beta_1 x$$

- $p/(1 - p)$ is called the odds of the event $y = 1$ and $x = x_i$
  - odds range between 0 and $+\infty$
- $\log[p/(1 - p)]$ is the log odds, also called *logit* function
  - log odds range between $-\infty$ and $+\infty$
- $\beta_0 + \beta_1 x$ is similar to *linear regression*
  - logistic regression is a generalization of regression to predict categorical variables

Logistic regression: Visually



- Learning $p(y = 1)$ for each value of x

Logistic regression: Assumption (for multivariate $\boldsymbol{x}$)

- When $\boldsymbol{x}$ a vector of d continuous varibales, we assume

$$p(y=1|\boldsymbol{x}, \beta_0, \beta_1, \ldots \beta_d) = \sigma(\beta_0 + \sum_i \beta_i x_i) = \frac{1}{1 + exp - [\beta_0 + \sum_i \beta_i x_i]}$$

where

  - $\beta = [\beta_0, \beta_1, \ldots, \beta_d]$ are the parameters
  - $\sigma(z) = 1/(1+e^{-z})$ is a nonlinear function

- Alternatively

$$\log \frac{p}{1-p} = \beta_0 + \sum_i \beta_i x_i$$

Likelihood for Logistic Regression

- Extension of the idea of likelihood
  - Likelihood is denoted as $L(\theta|x)$ or $L(\theta; x)$or $p(x|\theta)$ or $f(x|\theta)$.
- In our case we have data $\{(x_i, y_i) : i = 1, 2, \ldots n\}$
- The likelihood for this case is

$$L(\theta|x, y) = f(x, y|\theta)$$

- From logistic regression $p(y|x, \beta) = \sigma(\beta_0 + \beta_1 x)$
  - i.e., y follows a probability distr. that is different for different x.
  - All these functions share the same parameters θ.
- We can write the joint density of (x, y) as a product of conditional density of $y|x$ and marginal densitiy of x.

$$f(y, x|\theta) = f(y|x, \theta) \times f(x|theta)$$
$$Joint = Conditional \times Marginal$$

Conditional Likelihood

$$f(y, x|\theta) = f(y|x, \theta) \times f(x|theta)$$
$$Joint = Conditional \times Marginal$$

Conditional Likelihood

Conditional Likelihood of θ given data x and y is

$$L(\theta; y|x) = p(y|x) = f(y|x; \theta)$$

Principle of maximum conditional likelihood

Given data consisting of pairs $\{(x_i, y_i) : i = 1, 2, \ldots n\}$, choose a parameter estimate $\hat{\theta}$ that maximizes the joint conditional likelihood expressed as the product

$$\prod_i f(y_i|x_i; \theta)$$

- suffices to assume y_i are independent (x_is need not be indep.)

Maximizing Conditional Likelihood

$$\text{Conditional Likelihood} = \prod_i f(y_i|x_i; \theta)$$

$$\text{Log Conditional Likelihood } \ell = \sum_i \log f(y_i|x_i; \theta)$$

- We can write, $p(y_i = 1|x_i)$ as p_i (success in a Bernoulli trial)

$$\ell = \sum_{i;y_i=1} \log p_i + \sum_{i;y_i=0} \log(1-p_i)$$

- Partial derivative of ℓ w.r.t. a paramter β_j is

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i;y_i=1} \frac{\partial}{\partial \beta_j} \log p_i + \sum_{i;y_i=0} \frac{\partial}{\partial \beta_j} \log(1-p_i)$$

Maximizing Conditional Likelihood

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i;y_i=1} \frac{\partial}{\partial \beta_j} \log p_i + \sum_{i;y_i=0} \frac{\partial}{\partial \beta_j} \log(1-p_i)$$

- For an individual sample, if $y = 1$, then partial derivative

$$\frac{\partial}{\partial \beta_j} \log p = \frac{1}{p} \frac{\partial p}{\partial \beta_j}$$

- For an individual sample, if $y = 0$, then partial derivative

$$\frac{\partial}{\partial \beta_j} \log(1-p) = \frac{1}{1-p}\Big(-\frac{\partial p}{\partial \beta_j}\Big)$$

Let $e = exp[-\sum_{j=0}^{d} \beta_j x_j]$, so

$$p = \frac{1}{1+e} \qquad 1 - p = \frac{1+e-1}{1+e} = \frac{e}{1+e}$$

Maximizing Conditional Likelihood

$$p = \frac{1}{1+e} \qquad 1 - p = \frac{e}{1+e}$$

$$\begin{aligned}
\frac{\partial p}{\partial \beta_j} &= (-1)(1+e)^{-2} \frac{\partial e}{\partial \beta_j} \\
&= (-1)(1+e)^{-2}(e) \frac{\partial}{\partial \beta_j}[-\sum_j \beta_j x_j] \\
&= (-1)(1+e)^{-2}(e)(-x_j) \\
&= \frac{1}{1+e} \frac{e}{1+e} x_j \\
&= p(1-p)x_j
\end{aligned}$$

$$\frac{\partial}{\partial \beta_j} \log p = (1-p)x_j \qquad \frac{\partial}{\partial \beta_j} \log(1-p) = -p x_j$$

Maximizing Conditional Likelihood

We have:

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i;y_i=1} \frac{\partial}{\partial \beta_j} \log p_i + \sum_{i;y_i=0} \frac{\partial}{\partial \beta_j} \log(1-p_i)$$

$$\frac{\partial}{\partial \beta_j} \log p = \frac{1}{p}\frac{\partial p}{\partial \beta_j} \qquad \frac{\partial}{\partial \beta_j} \log(1-p) = \frac{1}{1-p}\Big(-\frac{\partial p}{\partial \beta_j}\Big)$$

Substituting:

$$\frac{\partial}{\partial \beta_j} \log p = (1-p)x_j \qquad \frac{\partial}{\partial \beta_j} \log(1-p) = -px_j$$

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i;y_i=1} (1-p_i)x_{ij} + \sum_{i;y_i=0} -p_i x_{ij} = \sum_i (y_i - p_i)x_{ij}$$

Maximizing Conditional Likelihood

$$\frac{\partial \ell}{\partial \beta_j} = \sum_i (y_i - p_i)x_{ij}$$

We get one equation like this for each parameter β_j.

Not possible to solve for β_j by equating the above equation to 0.

We resort to numerical optimization techniques such as Gradient Descent or Newton's method.

Newton's Method

- Overview
- General algorithm
- Newton's method for Logistic Regression
- Julia code

Newton's method

- A numerical optimization technique used to find the parameter vector $\boldsymbol{w}$ that minimizes an objective function $E(\boldsymbol{w})$.

$$\boldsymbol{w}^* = argmin_{\boldsymbol{w}} E(\boldsymbol{w})$$

- An iterative approach to estimate $\boldsymbol{w}$
  - Starts with an initial estimate $\boldsymbol{w}_1$ (often a random vector)
  - First and the second gradients $\nabla E(w_1)$ and $\nabla^2 E(w_1)$ are computed at this point.
  - Next estimate $\boldsymbol{w}_2$ is estimated as $\boldsymbol{w}_2 \leftarrow \boldsymbol{w}_1 - \nabla E(w_1)/\nabla^2 E(w_1)$
  - Generally $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i-1} - \nabla E(w_{i-1})/\nabla^2 E(w_{i-1})$
  - Stops after a given maxIter or when estimate $\boldsymbol{w}$ or ℓ converges

Newton's method - ∇E and $\nabla^2 E$

$$\nabla E(\boldsymbol{w}_{i-1}) = \Big[\frac{\partial E(\boldsymbol{w}_{i-1})}{\partial \beta_0}, \frac{\partial E(\boldsymbol{w}_{i-1})}{\partial \beta_1}, \ldots, \frac{\partial E(\boldsymbol{w}_{i-1})}{\partial \beta_d}\Big]$$

$$\nabla^2 E(\boldsymbol{w}_{i-1}) = \begin{pmatrix} \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_0^2} & \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_0 \beta_1} & \ldots & \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_0 \beta_d} \\ \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_1 \beta_0} & \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_1^2} & \ldots & \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_1 \beta_d} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_d \beta_0} & \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_d \beta_1} & \ldots & \frac{\partial^2 E(\boldsymbol{w}_{i-1})}{\partial \beta_d^2} \end{pmatrix}$$

- $\frac{\nabla E(w_{i-1})}{\nabla^2 E(w_{i-1})}$
  - Numerator is a vector and a denominator is a matrix
- $(\nabla^2 E(w_{i-1}))^{-1} \nabla E(w_{i-1})$
  - Invert the Hessian matrix $(\nabla^2 E(w_{i-1}))$ and multiply with the gradient $\nabla E(w_{i-1})$

Newton's method

- Recall that, if $E(\boldsymbol{w})$ is convex, it is equivalent to finding $\boldsymbol{w}^*$ such that $\nabla E|_{\boldsymbol{w}^*} = 0$

Taylor series

It is a representation of a function as an infinite sum of terms that are calculated from the values of the function's derivatives at a single point'

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots$$

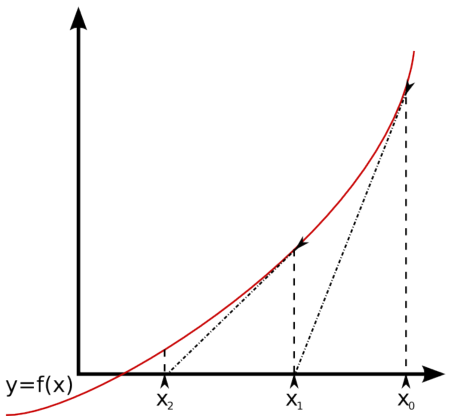
- Let $F(\boldsymbol{w}) = \nabla E(\boldsymbol{w})$. Taking Taylor expansion at the optimum solution $\boldsymbol{w}^*$

$$F(\boldsymbol{w}^*) = F(\boldsymbol{w}) + (\boldsymbol{w}^* - \boldsymbol{w})\nabla F(\boldsymbol{w}^*) + \text{ negligible terms}$$

- Because $F(\boldsymbol{w}^*) = \nabla E(\boldsymbol{w}^*) = 0$, we know

$$0 \approx F(\boldsymbol{w}) + (\boldsymbol{w}^* - \boldsymbol{w})\nabla F(\boldsymbol{w}^*) \implies \boldsymbol{w}^* \approx \boldsymbol{w} - \frac{F(\boldsymbol{w})}{\nabla F(\boldsymbol{w})} = \boldsymbol{w} - \frac{\nabla E(\boldsymbol{w})}{\nabla^2 E(\boldsymbol{w})}$$

Newton's method - interpretation



Newton's method: a general algorithm

Step 1: Pick initial value $\boldsymbol{w}_1$

Step 2: $maxIter = 10000$

Step 3: **for** $i = 2 : maxIter$

Step 4: $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{i-1} - \frac{\nabla E(\boldsymbol{w}_{i-1})}{\nabla^2 E(\boldsymbol{w}_{i-1})}$

Step 5: **if** $|\ell_i - \ell_{i-1}| < \epsilon$ terminate; **end**

Step 6: **end for**

Newton's method: an example

minimize $(x-c)^2$ or $argmin_x(x-c)^2$

- $f'(x) = 2(x-c)$
- $f''(x) = 2$
- $x_1 = x + 0 - \frac{f'(x)}{f''(x)} \implies x_1 = x_0 - \frac{2(x_0-c)}{2} = c$
- Newton's method may find the minimum solution in one step.
- Second derivative must exist

Newton's method: Advantages and Disadvantages

Advantages

- Converges quadratically towards a stationary point.

Comparision with Gradient Descent:

$$\lambda = \frac{1}{\nabla^2 E(\boldsymbol{w}_{i-1})}$$

Disadvantages

- Does not necessarily coverge toward a minimizer
- Diverges if the starting approximation is too far
- Requires second-rder information $\nabla^2 E(\boldsymbol{w}_{i-1})$
- Not suited if $\nabla^2 E(\boldsymbol{w}_{i-1})$ is not invertible

Newton's Method for Logistic Regression

First derivatives

$$\frac{\partial \ell}{\partial \beta_0} = \sum_i (y_i - p_i)$$

$$\frac{\partial \ell}{\partial \beta_1} = \sum_i (y_i - p_i) x_{i1}$$

Second derivatives

$$\frac{\partial^2 \ell}{\partial \beta_0^2} = -\sum_{i=1}^n p_i(1-p_i)$$

$$\frac{\partial^2 \ell}{\partial \beta_1^2} = -\sum_{i=1}^n x_i^2 p_i(1-p_i)$$

$$\frac{\partial^2 \ell}{\partial \beta_0 \beta_1} = -\sum_{i=1}^n x_i p_i(1-p_i)$$

General case:

First derivative

$$\frac{\partial \ell}{\partial \beta_j} = \sum_i (y_i - p_i) x_{ij}$$

where x_{ij} is the j^{th} attribute in the i^{th} sample.

Second derivative

$$\frac{\partial^2 \ell}{\partial \beta_j \beta_k} = -\sum_{i=1}^n x_{ij} x_{ik} p_i(1-p_i)$$

$$\frac{\partial^2 \ell}{\partial \beta_j^2} = -\sum_{i=1}^n x_{ij}^2 p_i(1-p_i)$$

where x_{ij} is the j^{th} attribute in the i^{th} sample.

Julia code - log likelihood and probability computation

```
# compute p(y=1|x_i)
function compute_p(x,b)
    p = 1./(1.+e.^(-x*b));
    return p;
end
```

```
## compute_p (generic function with 1 method)
```

```
function compute_l(x,y,b)
    p = compute_p(x,b);
    prob = y.*log.(p) + (1-y).*log.(1-p);
    l = sum(prob[.!isnan.(prob)]);
    return l;
end
```

```
## compute_l (generic function with 1 method)
```

Julia code - first and second derivaties

```
function compute_first_derivatives(x,y,p)
    d1 = zeros(2);
    d1[1] = sum(y.-p);
    d1[2] = sum((y.-p).*x[:,2]);
    return d1;
end
```

```
## compute_first_derivatives (generic function with 1 method)
```

```
function compute_second_derivatives(x,y,p)
    d2 = zeros(2,2);
    d2[1,1] = sum(p.*(1.-p));
    d2[2,2] = sum((x[:,2].^2).*p.*(1.-p));
    d2[1,2] = sum(x[:,2].*p.*(1.-p));
    d2[2,1] = d2[1,2];
    return d2;
end
```

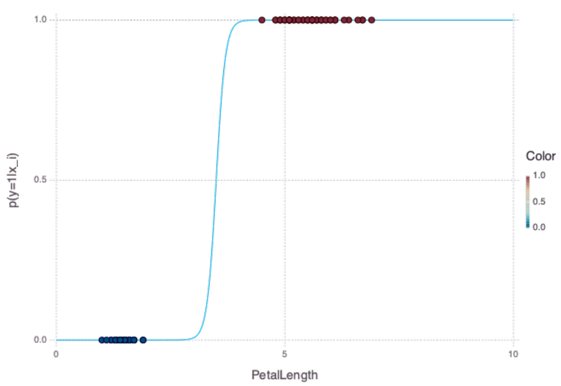
```
## compute_second_derivatives (generic function with 1 method)
```


Julia code - Newton's method

```
function newtons_lr(x,y)
    max_itr = 20; # maximum num. iterations
    b = [-4 1]'; # random initialization
    l = compute_l(x,y,b); # compute log-likelihood
    for i=1:max_itr
      p = compute_p(x,b); #compute prob.
      d1 = compute_first_derivatives(x,y,p);
      d2 = compute_second_derivatives(x,y,p);
      b_new = b.+inv(d2)*d1; #update betas
      l_new = compute_l(x,y,b_new);
      if(abs(l-l_new)<0.00001) break; end;
      l = l_new;
      b = b_new;
    end
    return b;
end
```

```
## newtons_lr (generic function with 2 methods)
```


Julia code - Newton's method (result)



Other scenarios