Chapter 3 : Linear Models for Regression

The simplest form of linear regression models are also linear functions of the input variables. However, we can obtain a much more useful class of functions by taking linear combinations of a fixed set of nonlinear functions of the input variables, known as *basis functions*. Such models are linear functions of the parameters, which gives them simple analytical properties, and yet can be nonlinear with respect to the input variables.

Basis function intuition : Transformations (Fixed non-linear functions of x that can be used in modelling)
Examples : Log (scale issues), Negative Binomial (overdispersed/count), Poly, Splined, Gaussian ($exp(-z^2/2)$), Sigmoid(z) similar to tanh, Fourier (expansion in sinusoidal functions)

Intercept : Also called a *bias parameter* given it allows for any fixed offset in the data

Maximum Likelihood and Least Squares :

We define target as sum of deterministic function in x and Gaussian noise.
Based on LR assumptions we define conditional probability of target given x. We evaluate expectation on both sides and error is removed, assume independence to avail product form for likelihood. We take log to avoid product and exp computation. Deriving we reach formula :

We maximize wrt w. We know that maximization of the likelihood function under a conditional Gaussian noise distribution for a linear model is equivalent to minimizing a sum-of-squares error function given by $E_D(w)$

Setting gradient wrt w to zero we have :
$w_{ML} = (\phi^T\phi)^{-1}\phi^T t$ which are known as the *normal equations* for the least squares problem. Here ϕ is called the *design matrix* and the quantity
$\phi^+ = (\phi^T\phi)^{-1}\phi^T$ is called *Moore-Penrose pseudo-inverse (left)* of the matrix ϕ which equates to ϕ^{-1} when phi is invertible square matrix.

A note on the weird beta issue

In practice, a direct solution of the normal equations can lead to numerical difficulties when $\phi^T\phi$ is close to singular. In particular, when two or more of the basis vectors are co-linear, or nearly so (highly correlated), the resulting parameter values can have large magnitudes. Such near degeneracies will not be uncommon when dealing with real data sets. The resulting numerical difficulties can be addressed using the technique of singular value decomposition, or SVD *(Press et al., 1992; Bishop and Nabney, 2008)*. Note that the addition of a regularization term ensures that the matrix is non-singular, even in the presence of degeneracies.

Multicollinearity over-inflates the standard errors associated with the respective regression coefficients, causing the p-values to become very sensitive to changes in model specification, resulting in the whole process becoming unstable. Consequently, multiple competing models may be produced, making it difficult to be confident about any inference drawn from the various models. In contrast to MLR, singular value decomposition (SVD) is immune to multicollinearity because it produces a set of orthogonal composite variables that are completely uncorrelated. Although not a statistical technique per se, SVD underpins other techniques such as principal component analysis (PCA) and partial least squares correlation analysis (PLSCA). PLSCA in particular appears to have considerable potential with regard to the analysis of small data sets that exhibit multicollinearity (e.g. sports performance data sets). Because PLSCA incorporates SVD, it has the great advantage that it is both immune to multicollinearity, and unlike MLR, can cope with situations where the number of predictor variables exceeds the number of observations.

Sequential Learning

Batch techniques, such as the maximum likelihood solution, which involve processing the entire training set in one go, can be computationally costly for large data sets.

Sequential learning takes in 1 line at a time and updates model parameters (Bayesian?)

Both the **single step updating**, in which all data are used together to update the prior and arrive at the posterior, and the **Bayesian sequential (also called recursive) updating**, in which data are used one at a time to obtain a posterior which becomes the prior of the successive iteration, must give exactly the same result. This is one of the pillars of Bayesian statistics: **consistency**.

We can obtain a sequential learning algorithm by applying the technique of stochastic gradient descent, also known as **sequential gradient descent**. SGD algorithm updates the parameter vector w using
$w^{(\tau+1)} = w^{(\tau)} - \eta\nabla E_n$ where ∇ is analogous to learning rate. If we subsitute error function E_n with sum of squares it becomes least mean square or **LMS algorithm**.

Side note 1 - Don't decay the learning rate, increase the batch size (Google Brain)
https://arxiv.org/pdf/1711.00489.pdf

Side note 2 - Markov state models (MSMs) : They model the dynamics of a system as a series of memoryless, probabilistic jumps between a set of states.

Side note 3 : A *loss function* (minimize) is a part of a *cost function* (minimize) which is a type of an *objective function* (needed to optimize)

Side point 4 :
OLS (Ordinary) - can't handle heteroscedasticity and multicollinearity
WLS (Weighted) - handles heteroscedasticity not multicollinearity (weight for the ith case is inversely proportional to $Var(\epsilon_i)$)
GLS (Generalized) - handles both heteroscedasticity and multicollinearity (need to know form of errors independently)
IRLS (Iteratively reweighted) - When little is known about the covariance structure of the errors independently of the data
TLS (Total) - Treats the covariates and response variable in a more geometrically symmetric manner than OLS. It is one approach to handling the "errors in variables" problem
*LMS (Least Mean): Based on gradient descent (batch or stochastic)
http://cs229.stanford.edu/notes2020fall/notes2020fall/cs229-notes1.pdf

Regularized least squares

Ridge (q=2) : Known in the machine learning literature as *weight decay* because in sequential learning algorithms, it encourages weight values to decay towards zero, unless supported by the data. In statistics, it provides an example of a parameter *shrinkage method* because it shrinks parameter values towards zero.
Solving for w in objective function updated with the penalty term
$w_{Ridge} = (\lambda I + \phi^T\phi)^{-1}\phi^T t$

Lasso (q=1) : Return *sparse* model
Q=0.5 (*curved diamond*) gives stronger lasso-like behavior and Q=4 (*curved rectangle*) gives slight ridge like behavior?

Multiple output regression

Matrix form makes this very simple. MLE essentially results in :
$w_{multiple} = (\phi^T\phi)^{-1}\phi^T T$ where T is matrix of all individual target vectors.

The Bias-Variance Decomposition

The phenomenon of over-fitting is really an unfortunate property of maximum likelihood and does not arise when we marginalize over parameters in a Bayesian setting.

Consider a frequentist viewpoint of the model complexity issue, known as the bias- variance trade-off.
Bias = f(Mean prediction - True distribution)
Variance = f(Prediction - Mean prediction)

We see that the expected squared difference between target and the regression function can be expressed as the sum of two terms. The first term, called the *squared bias*, represents the extent to which the **average prediction over all data sets differs from the desired regression function**. The second term, called the *variance*, measures the extent to which the solutions for **individual data sets vary around their average**, and hence this measures the extent to which the prediction function is sensitive to the particular choice of data set.
Very flexible models : Low Bias, High Variance
Very Rigid models : High Bias, Low Variance

Bayesian Linear Regression

Bayesian treatment of linear regression works by introducing a prior probability distribution over the model parameters w.

Note : In Bayesian probability theory, if the posterior distribution $p(\theta|x)$ is in the same probability distribution family as the prior probability distribution $p(\theta)$, the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function $p(x|\theta)$.

Next we compute the posterior distribution, which is proportional to the product of the likelihood function and the prior. Due to the choice of a conjugate Gaussian prior distribution, the posterior will also be Gaussian.

Conjugate prior form :

$$p(w) = N(w|m_0, S_0)$$

Posterior distribution form :
$p(w|t) = N(w|m_N, S_N)$ where
$m = S_N(S_0^{-1}m_0 + \beta\phi^T t)$ and
$S_N^{-1} = S_0^{-1} + \beta\phi^T\phi$

This implies if we consider an infinitely broad prior $S_0 = \alpha^{-1}I$ with $\alpha \to 0$, the mean m_N of the posterior distribution reduces to the maximum likelihood value $w_M L$

It is understood that if data points arrive sequentially, then the posterior distribution at any stage acts as the prior distribution for the subsequent data point.

Assumption : Consider a particular form of Gaussian prior in order to simplify the treatment. Specifically, we consider a zero-mean isotropic Gaussian governed by a single precision parameter α and calculate log of posterior, apply maximization on it equates to minimization of the sum-of-squares error function with the addition of a quadratic regularization term with $\lambda = \alpha/\beta$ where alpha comes from assumption in prior and beta comes from assumption in noise

Visualization for MAP :