Problem Statement

- Design an algorithm that predicts the bankruptcy state of a company
- Classifies the companies into groups that represent the bankruptcy state
- Only three possible states are defined for the sake of simplicity
- Gaussian discriminant analysis approach used for the machine learning problem
- Bayes optimal solution

Data

- Financials of 200 companies
- Ratios used:
  - Working Capital / Total Assets
  - Retained Earnings / Total Assets
  - EBIT / Total Assets
  - Market Value of Equity / Book Value of Total Liabilities
  - Sales/ Total Assets

Prior Classification

- Initial classification using Z-score
- $Z = 1.2T_1 + 1.4T_2 + 3.3T_3 + 0.6T_4 + 0.999T_5$

Where T, T2, T3, T4, T5 are ratios mentioned previously

- Prior probabilities calculated from this classification
- This is used as the training dataset

GDA: Algorithm

- Probability that we're dealing with class *k* while observing currently point x is:

  ***P*(*k*|x)=*P*(*k*)∗*P*(x|*k*)/*P*(x)**
- i.e ***P*(*k*|x)=*P*(*k*)∗*P*(x|*k*)**
- *P*(*k*) is prior probability that the native class for x is *k*
- But,

$$PDF(x|k) = \frac{e^{-d/2}}{(2\pi)^{p/2}\sqrt{|\mathbf{S}|)}}$$

- **d** - squared Mahalanobis dist $D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}.$
- **S** – covariance matrix *between the discriminants*, observed within that class.
- **p** – dimension of p-variate distribution

GDA: Bayes Optimal Solution

- $PDF(x\,|\,k)$ computed for each of the classes.
- $P(k) * PDF(x\,|\,k)$ normalized dividing by the sum of $P(k) * PDF(x\,|\,k)$s over all classes.
- $P(k\,|\,x) = P(k) * PDF(x\,|\,k) / [P(k) * PDF(x\,|\,k) + P(l) * PDF(x\,|\,l) + P(m) * PDF(x\,|\,m)]$
- Point x is assigned by LDA to the class for which $P(k\,|\,x)$ is the highest.

Testing

- Algorithm reclassifies data
- Training data of length 180
- Predicted data for 34 companies
- 82.5% match between predicted and actual data
- Accuracy increases as data points increase

References

- **Gaussian Discriminant Analysis, Naïve Bayes** by Marco Chiarandini (University of Southern Denmark)
- **Bayes Optimality in Linear Discriminant Analysis**, by Onur C. Hamsici and Aleix M. Martinez , The Ohio State University
- http://stats.stackexchange.com/

Option Pricing using Finite Difference Method

- Finite difference methods (also called finite element methods) are used to price options by approximating the (continuous-time) differential equation that describes how an option price evolves over time by a set of (discrete-time) difference equations.

- The discrete difference equations may then be solved iteratively to calculate a price for the option.

- We have taken European call and put options for pricing using this method.

Option Pricing Using The Implicit Finite Difference Method – Algorithm

- **Step 1** Discretization of differential equation

For the implicit method the Black-Scholes-Merton partial differential equation,

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf$$

Different Approximations for different of the equation –

- Backward Difference -->

$$\frac{\partial f}{\partial t} = \frac{f_{i,j} - f_{i-1,j}}{\delta t}$$

Continue

- Central Approximation for – $$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\delta S}$$

- Standard Approximation- $$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{(\delta S)^2}$$

- **STEP 2** Equation Formulation

$$\frac{f_{i+1,j} - f_{i,j}}{\delta t} + rj\delta S \frac{f_{i,j+1} - f_{i,j-1}}{2\delta S} + \frac{1}{2}\sigma^2 (j\delta S)^2 \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{(\delta S)^2} = rf_{ij}$$

Equation Formulation

- Which reduces to a finite difference equation

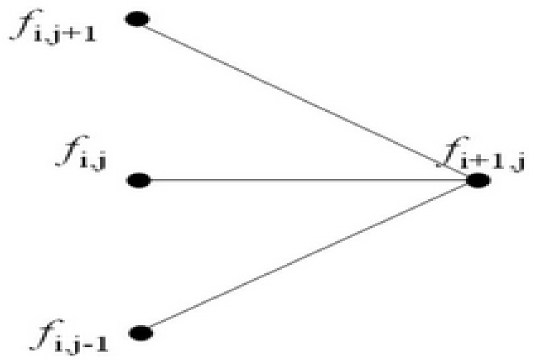
$$a_j^* f_{i,j-1} + b_j^* f_{i,j} + c_j^* f_{i,j+1} = f_{i+1,j}$$

- Where the value of these constants is below-

$$\begin{aligned} a_j^* &= \frac{1}{2}\delta t\left(rj - \sigma^2 j^2\right) \\ b_j^* &= 1 + \delta t\left(\sigma^2 j^2 + r\right) \\ c_j^* &= \frac{1}{2}\delta t\left(-rj - \sigma^2 j^2\right) \end{aligned}$$

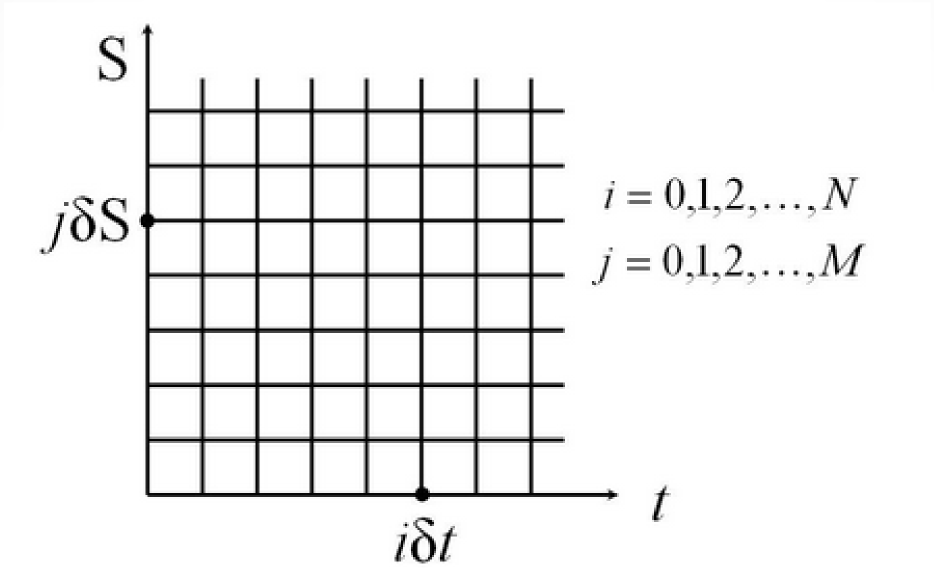
Grid Formulation

- These equations are solved working backwards in time.



- However when all the equations at a given time point are written simultaneously there are M-1 equations in M-1 unknowns. Hence the value for f can be calculated at each node uniquely.

Actual Time and Price Grid



- **Step -3** Calculate the payoff of the option at specific boundaries of the grid of potential underlying prices

- **Step-4** Iteratively determine the option price at all other grid points, including the point for the current time and underlying price.

Option Pricing Using The Explicit Finite Difference Method – Algorithm

- **Step 1** Discretization of differential equation

For the implicit method the Black-Scholes-Merton partial differential equation,

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf$$

Different Approximations for different of the equation –

- Forward Difference -->

$$\frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{\delta t}$$

Continue

- Central Approximation for – $$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\delta S}$$
- Standard Approximation- $$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{(\delta S)^2}$$
- **STEP 2** Equation Formulation

$$\frac{f_{i,j} - f_{i-1,j}}{\delta t} + rj\delta S \frac{f_{i,j+1} - f_{i,j-1}}{2\delta S} + \frac{1}{2}\sigma^2 (j\delta S)^2 \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{(\delta S)^2} = rf_{ij}$$

Equation Formulation

- Which reduces to a finite difference equation

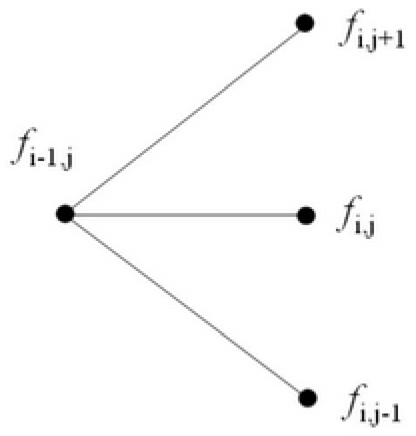
$$f_{i-1,j} = a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1}$$

- Where the value of these constants is below-

$$\begin{aligned} a_j &= \frac{1}{2}\delta t\left(\sigma^2 j^2 - rj\right) \\ b_j &= 1 - \delta t\left(\sigma^2 j^2 + r\right) \\ c_j &= \frac{1}{2}\delta t\left(\sigma^2 j^2 + rj\right) \end{aligned}$$

Grid Formulation

- **Step 3** Calculate the payoff of the option at specific boundaries



- **Step 4** Given values of $f_{i,j+1}$, $f_{i,j}$, and $f_{i,j-1}$, then values for $f_{i-1,j}$ can explicitly and easily be calculated. This figure shows that given the value of the option at the boundary conditions, then all interior points of the pricing grid can be calculated.

Results of Matlab code

Input Variables --
Strike - 60, S0 - 50 , Risk free rate - 0.05,
Volatility - 0.2 , Svector - 0 to 100 increment of 1 ,
Tvector - 0 to 1 increment of 0.01 , OptionType - Call/Put

	Call	Put
Black Scholes Price	1.6237	8.6975
Explicit Method	1.6209	8.6951
Implicit Method	1.5862	8.6954

Truncation Error

Assumption - Del (x)= sigma*[3*del- t]½.

Derivation:

T(j, n)= [Û(j,n+1)]/Del(t) – 0.5 sig^2 [Û(j+1,n)-2 Û (j, n) + Û (j-1,n)]/(del x)^2

Where Û (j, n)→ analytical price at point (n, j)

Using Taylor's series expansion we can write;

T(j, n) = (Û(t)- 0.5*sig^2*Û(xx))+ 0.5*Û(t, t) del(t) –(1/24) .sig^2. Û(xxxx).(del x)^2 +O((Del x)^4) + O((del t)^2).

Where Û(xx): denote double differentiation w.r.t space.

Û(tt): double differentiation w.r.t time. [This is "0" according to the B-S PDE]

Differentiating Û(t) w.r.t to "t" we get Û(tt) =(.5*sig^2)^2Û (xxxx)

Truncation Error

So, T(j,n) can be approximately written as

T(j, n) = (1/8) sig^2*Û(xxxx) [sig^2*del(t)-(1/3) (del x)^2]

To make it zero we set Del (x)= sig* sqrt(3* del(t)).

Stability Analysis

The explicit finite difference is stable if the following condition holds:

[(Del-X)^2]/(Sig)^2≥ Del-T

Derivation:

We can show that for the finite difference method to be stable â ≤ 0.5.

Here, â = 0.5*(sig)^2*(del-t)/(del-x)^2

Hence, del-t≤ (del-x)^2/(sig)^2

Here, del-X= Sig*(3*del-t)^(.5) .Therefore by choosing this del-X we can make the system always stable

Thank You ..