Senior Project Proposal:
Risk Assessment Tool for Collateralized Loans

Introduction

I believe incorporating my studies of both Computer Science and Economics in my senior project to be a fruitful intellectual challenge. Professor Stanley Eisenstat in the Computer Science Department, who holds a Ph.D. in Mathematics and from whom I took Systems Programming and Computer Organization, has agreed to supervise the project. Sigridur Benediktsdottir in the Economics Department, with whom I took Financial Markets last year, has agreed to co-advising the Economics components of the project. Sigga holds a B.S. in Computer Science and a Ph.D. in Economics.

Economic Motivation

Sigga spent the last eight months in her home country of Iceland, working with the legal team assigned to investigate the factors that led to the collapse of Iceland’s financial sector last year. By interviewing key economic players – from the Prime Minister to bank presidents – and digging through databases of transaction data, she discovered a fundamental flaw in the lending policies of many Icelandic banks: though each lending desk within a given bank tended to follow proper diversification guidelines when issuing a collateralized loan, aggregating loan sheets across the entire bank revealed that a large fraction of the bank’s loans were backed by the same or similar assets, for example stock in Icelandic Air.

This flawed decision-making had two major repercussions. The first was that the risk of each collateralized loan was actually much higher than the banks realized, since the collateralized assets were highly correlated. The second result was that when the economy slowed and the value of these assets fell, the terms of the loan required banks to post a margin call, asking lenders to put up additional collateral. This was often done by (1) putting up more of the same asset as collateral, (2) defaulting on the loan, or (3) selling the initial collateralized asset to pay off some of the loan. The first and second options left banks with more of an asset whose value was declining and the third option further depressed the asset’s value. Additionally, falling asset values could result in lower ratings for the bank’s financial instruments, and, subsequently, higher financing costs for the bank. Under these conditions, banks often chose not to demand additional collateral from borrowers, resulting in even riskier loans.

Description of Tool

On the simplest level, this is an issue of communication and coordination between the various desks within a bank, but considering that each loan has a different duration to maturity, risk of default, and probability of prepayment, the problem becomes highly dynamic. I plan to develop a proof-of-concept computational tool that would predict the risk of a bank’s collateralized loans under different market liquidity conditions. It would attempt to answer the following question: how would a liquidity crisis affect a bank’s lending book? It would then display the answer in an intuitive format for human interpretation.

When a potential borrower request a loan, the bank’s loan officer would input the borrower’s name, the value of the loan, and the asset he is offering as collateral. The system would query the bank’s centralized records to determine what other loans he has on the book and
produce a human-readable output to describe the risk in accepting the loan. One of the bank’s risk managers could also query the system to determine how much of the bank’s loans are collateralized by highly-correlated stocks. Sigga sees that such a tool would be useful to regulators in countries such as Iceland, who could use it as a diagnostic tool in assessing a bank– or a sector’s– economic stability.

Implementation

I plan to write the tool in Python. Having used it for other projects, I am comfortable with the language and it is often easy to find useful prewritten modules. For example, TkInter\(^1\) is the standard Python GUI editor included with the Python installation, but there are many other choices as well\(^2\). Sphinx\(^3\) is a tool for creating documentation and reports, matplotlib\(^4\) is a 2D plotting library that produces publication-quality figures, and both of these tools can be easily integrated.

I have not yet decided whether it is better to build the tool as a stand-alone window or browser-based application. I found prewritten modules for building both GUI and HTML interfaces. Since Sphinx can output attractive HTML, I will begin by exploring that option. After our discussion, I think it would be best to begin with a single user interface designed for a loan officer, but with extra options to generate output that would satisfy a risk manager as well. Since the data will not contain more than two dimensions, it is probably simplest to maintain the database as a file system.

A common measure of risk is called value at risk (\(VaR\)). For a given portfolio, probability, and time horizon, \(VaR\) is defined as the threshold value such that the probability that the loss on the portfolio over the given time horizon exceeds this value is the given probability level. That is, given some confidence level \(\alpha \in (0,1)\), the \(VaR\) of the portfolio at the confidence level \(\alpha\) is given by the smallest number \(l\) such that the probability that the loss \(L\) exceeds \(l\) is not larger than \((1 - \alpha)\); \(VaR_\alpha = \inf\{l \in \mathbb{R} : P(L > l) \leq 1 - \alpha\}\). But this measure does not adequately account for cross-correlation between assets. The theoretical challenge behind this project will be to find a way to incorporate both dimensions of risk into a single, instructive measure.

Timeline

Professor Eisenstat and I plan to meet weekly to evaluate my progress and discuss implementation decisions. Before these meetings, I will record short status updates for my own reference when writing the final report. I will be responsible for teaching him any necessary Economics and Finance material. I will meet with Sigga about as often to design the details of the tool. During some phases of the project we will meet more or less frequently. Below is the timeline I propose for completing the project:

Read articles and book chapters (Week 1: Friday, February 12)
  • Gain background in risk assessment theory (Week 1)

---

\(^1\) http://wiki.python.org/moin/TkInter
\(^2\) http://wiki.python.org/moin/GuiProgramming
\(^3\) http://sphinx.pocoo.org/
\(^4\) http://matplotlib.sourceforge.net/
\(^5\) http://en.wikipedia.org/wiki/Value_at_risk
• Learn approaches to computing VaR (Week 1)
• Understand how cross-correlation increases risk (Week 1)
• Think about how to incorporate both VaR and cross-correlation in a single measure (Week 1)

Design and build interface (Weeks 2 to 4: Friday, March 5)
• Decide what information should be included in the output (Week 2)
• Based on that, determine what information should be included in the input (Week 2)
• Search online for possible input templates (Week 2)
• Sketch layout of input and output windows (Week 2)
• Code input and output windows with placeholders for data (Weeks 3 & 4)

Spring Break (Weeks 5 to 6)

Design and populate the database (Weeks 7 to 8: Friday, April 2)
• Determine what data is required based on demands of the interface (Week 7)
• What is the best way to represent the data? (Week 7)
• Distributions of the data (Week 7)
• Generate random data (Week 8)
• How to read from the database (Week 8)
• How to represent the data in memory (Week 8)

Code link between interface and database (Weeks 9 to 10: Friday, April 16)
• Process inputs (Week 9)
• Get relevant data from database (Week 9)
• Perform risk calculations (Week 9)
• Output into template (Week 10)

Extensions in time permits: Data input and secondary interface (Week 11: Friday, April 23)
• When a loan officer accepts a loan, update the database to include the loan (Week 11)
• Separate loan officer’s interface from risk manager’s interface (Week 11)

Write documentation (Week 12: Friday, April 30)
• Package the final code for the tool (Week 12)
• Write explanation of the implementation (Week 12)
• Use weekly status updates to document progress (Week 12)
• Generate web pages and post them (Week 12)

References