Title

Kernel Density Estimation Clustering Algorithm with an Application in Characterizing Volatility Smiles

Author
del Valle, Arturo

Publication Date
2014

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Kernel Density Estimation Clustering Algorithm with an Application in Characterizing Volatility Smiles

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Arthur del Valle

2014

Abstract of the Thesis

Kernel Density Estimation Clustering Algorithm with an Application in Characterizing Volatility Smiles

by

Arthur del Valle

Master of Science in Statistics

University of California, Los Angeles, 2014

Professor Nicolas Christou, Co-chair

Professor Rick P. Schoenberg, Co-chair

An algorithm is devised for clustering observations based on the densities of points within each individual observations. The Kernel Density Estimation Clustering Algorithm (KCA) performs a search on the graph of the observations' group memberships, where group memberships determines the KDEs that in turn drive changes in the objective function. Option pricing theory is used to demonstrate the utility of the algorithm. Volatility smiles, 2-dimensional plots relating option strike prices and their implied volatilities, are the basis for grouping options.

The thesis of Arthur del Valle is approved.

Hongquan Xu

Rick P. Schoenberg, Committee Co-chair

Nicolas Christou, Committee Co-chair

University of California, Los Angeles

2014

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In their foundational paper, Black and Scholes establish a relationship between the price of an option contract and the price of an underlying security on which the option is based [1]. One of the parameters for the Black-Scholes function, the volatility, is not directly observed and empirical evidence suggests that it is not static over time (Black and Scholes stated so themselves). However, having observed the market price for the option, it becomes possible to solve for the volatility. The implied volatility (IV), as it is referred to, has been studied extensively. A large amount of literature directs its attention at producing new forecasting models that can further minimize error, sidestepping the understanding of underlying causation or complex relationships [11], [17]. Even where an understanding of the relationships between volatility, term-structures, etc., is sought, analysis is generally conducted on either a single option-chain or an option on an index [2].

Our analysis applies a clustering method on subsets of the option-chains of various securities. The resulting groups can be used to further explore prediction schemes and to seek new relationships. The option-chain characteristics that are of interest are collectively known as a volatility smile. Each volatility smile is constructed with a subset of options sharing an underlying security, expiration date, and the time at which their prices were capture. To fully leverage the minute idiosyncrasies of volatility smiles, clustered groupings are based on kernel density estimates (KDE). This necessitates the development of a framework for clustering KDEs; we refer to the resulting algorithm as the KDE Clustering Algorithm (KCA). The focus of this study is on the method itself. The search for the aforementioned complex relationships are left for future research.

The following document is organized as follows: In chapter 2, we begin by introducing

some of the language and concepts used. We then proceed to further detail some necessary background in chapter 3. Certain aspects of Option Pricing Theory are discussed and some relevant literature reviews are conducted. In this chapter we also briefly discuss the EM algorithm and the K-Means algorithm, clustering algorithms that influenced the development of the KCA. In chapter 4, we introduce the KCA. In addition to the iterative algorithm, use of the KCA requires a discretization of the standard KDE algorithm; this too is covered in chapter 4. In chapter 5, an options dataset is described, input into the KCA, and analyzed. Chapter 6 summarizes the work and suggests future modifications and analyses.

CHAPTER 2

Concepts and Nomenclature

Stock options, or option contracts, are contracts that allow, but do not require, the owner to purchase (or sell) a fixed amount of an underlying security at a set price. The price at which the security may be purchased is called the strike price; the underlying security is often referred to simply as the underlying. A contract allowing for the purchase of the securities is called a call option while a contract allowing for the sale of securities is called a put option. American Options allow for such a transaction to occur before or up to the expiration date while European options allow for the such an exercis to occur only at the set expiration date.

Option chain (or simply chain, if the context is clear) is the name given to the snapshot that a trader retrieves containing the prices of options for a given underlying with all expiration dates and strike prices currently being traded. We use the name option sub-chain (or just sub-chain) to refer to the subsets of a chain that share expiration dates. E.g., if two distinct expiration dates are currently being traded, each chain will consist of two sub-chains. It should be noted that all of the sub-chains may or may not share the same set of strike prices; this is influenced by a number of factors including trade volume, security price, and expected volatility.

The sub-chain's strike prices can be plotted against the implied volatility to create a visualization called a volatility smile. An example of a volatility smile is plotted in figure 2.1 (p. 4). The volatility smile is characterized by higher IVs for options that are further in-the-money or further out-of-the-money. In-the-money refers to any call option that has a strike prices below the value of its underlying security; out-of-the-money refers to options not in-the-money, at-the-money is reserved for the option whose strike price is closest to the price of the underlying.

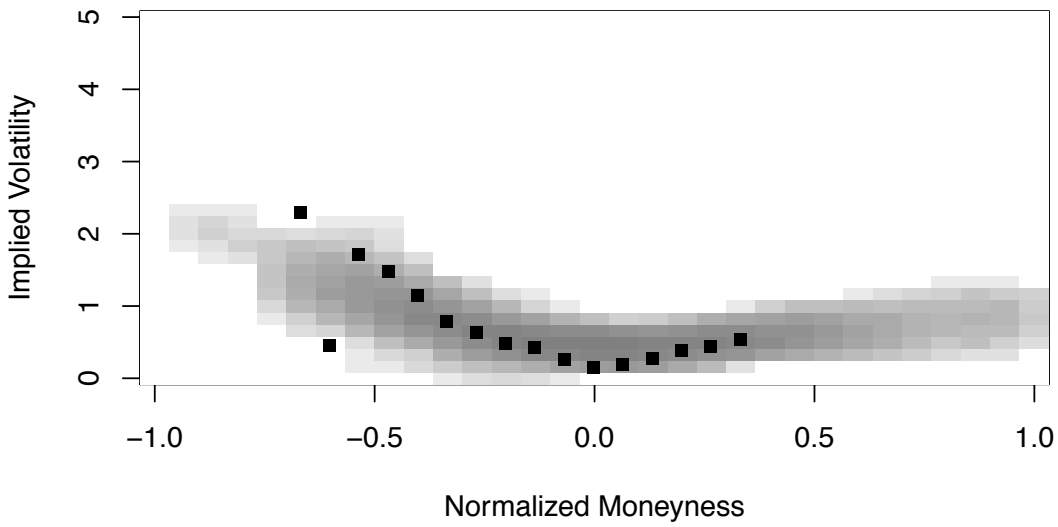

Figure 2.1: Example of volatility smile for the WMT data in Table 5.1 (p. 22) with the second KDE that result from the KCA run in Chapter 5.

Strike Price	Norm Strike	Option Price	Norm Moneyness	Impl Volatility
27.50	0.0000000	25.225	0.6690670	2.298449
30.00	0.0666667	22.550	0.6024003	0.459819
32.50	0.1333333	20.200	0.5357337	1.711768
35.00	0.2000000	17.700	0.4690670	1.481238
37.50	0.2666667	15.150	0.4024003	1.149564
40.00	0.3333333	12.600	0.3357337	0.786023
42.50	0.4000000	10.100	0.2690670	0.631301
45.00	0.4666667	7.600	0.2024003	0.481424
47.50	0.5333333	5.150	0.1357337	0.422026
50.00	0.6000000	2.665	0.0690670	0.256186
52.50	0.6666667	0.510	0.0024003	0.150920
55.00	0.7333333	0.035	-0.0642663	0.189730
57.50	0.8000000	0.010	-0.1309330	0.271303
60.00	0.8666667	0.010	-0.1975997	0.379565
62.50	0.9333333	0.005	-0.2642663	0.443748
65.00	1.0000000	0.005	-0.3309330	0.531933

Table 2.1: Sub-chain for option on Walmart stock (WMT) with expiration date 03-19-2011, traded on 03-11-2011. The date's stock price is $52.59 and the risk-free return rate is 0.78%.

CHAPTER 3

Background

This section is divided into 2 parts: the first gives some background on option pricing, the latter describes the EM algorithm. Understanding option pricing motivates the use of the KCA on option sub-chains. Understanding the EM helps introduce the KCA.

3.1 Option Pricing

The work of Black and Scholes established a theoretical foundation for the pricing of options. [1]. Black & Scholes determined that if markets are in fact complete and efficient, a basket comprised of a number of risk-less assets and a number of the underlying securities could be used to perfectly hedge a long/short position on an option. Given that this basket is risk-less, its return would have to equal that of the risk-less asset in order to negate the possibility of an arbitrage opportunity. Making assumptions on the distribution of the security's return and features of the market, the Black-Sholes formula solves the differential equation that defines the strategy for eliminating the risk. The formula is as follows:

$$w(x,t) = xN(d_1) - ce^{r(t-t*)}N(d_2) \tag{3.1}$$

$$d_1 = \frac{\ln x/c + \left(r + \frac{1}{2}v^2\right)(t*-t)}{v\sqrt{t*-t}} \tag{3.2}$$

$$d_2 = d_1 - v\sqrt{t*-t} = \frac{\ln x/c + \left(r - \frac{1}{2}v^2\right)(t*-t)}{v\sqrt{t*-t}}, \tag{3.3}$$

and is a function of the following parameters: x, the price of the underlying security; t and $t*$, the current time and time of expiration, respectively (where a difference of 1 year is unity); c, the strike price of the contract; r the annualized return of the risk free asset; and v^2 the variance of the return. The underlying's price, strike prices, and time until expiration

are all easily observable; agreeing to certain assumptions, the risk-free rate can be set to a minimally-risky, easily-observable instruments such as treasury bonds. Black and Scholes describe v^2 as the limit of the underlying's return as its period of measurement goes to zero– the instantaneous volatility; a simplification necessary to quantify the uncertainty of the underlings price movement until expiration, but a quantity that is not directly observable. Although they give it a static value, evidence has shown that the volatility is time-varying [17].

Once market prices have been observed for options, the BS formula can be inverted to calculate the volatility that would generate such prices; this is called implied volatility and it has been the focus of much research. The implied volatilities have garnered additional interest because the options in a sub-chain, given that they share the same underlying security at the same point in time, should have the same associated volatility. Instead, they posses a smile of the sort shown in figure 2.1 (p. 4). If a single value were implied for the volatility, it could be interpreted as the expected volatility, as opposed to historical volatility. This has led many researcher to model the underlying's volatility in the context of options, particularly since the use of IVs has been shown to outperform the use of historical volatility in forecasts [3]. A large portion of the literature focuses on fitting a diffusion parameter, a jump parameter, or combination of both, and thus creating a forecasted point and spread estimate based on recent IVs; these models are referred to as stochastic volatility models and are akin to time-series analyses [17]. Another portion of the literature has focused on creating weighted averages of the implied volatilities, essentially just using recent IV data [11].

The focus of these studies has been forecasting and they have not leveraged data beyond that of the security in question. As the literature on stochastic volatility and implied volatility show, often the shape of the volatility smile is not even of interest. Yet certain dynamics of the smile's shape are known to be related to the price and volatility, and attempts should be made to incorporate these into forecasts. For example, it is well-established that the smile is flatter the further it is from expiration and that securities that have experienced recent volatility also experience a decline in price [3]. If options or underlyings exhibit additional

similarities in their dynamics during period in which their smiles posses similarities, models fitted with this in mind could potentially improve forecasting performance. This is how we motivate the need to cluster securities and their option chains based on the characteristics of their smiles' densities.

3.2 Expectation Maximization

The EM algorithm is a framework used for missing data problems [4]. A common use of the algorithm is clustering; the group memberships of the observations are the missing variables. Clustering with the EM algorithm focuses on maximizing the likelihood $\log \mathcal{L}(\theta|x) = P(x|\theta)$ where θ is a function of x, namely θ is divided into parameters for each of k groups, $\theta_1, \ldots, \theta_k$, and these are to be maximized according to the group membership. The group memberships are missing and typically labeled z or y_{mis}. Maximizing $\mathcal{L}$ by maximizing each θ_i for every possible permutation of $\{z_1, \ldots, z_n\}$ is , for all but trivial problems, intractable. Instead of solving every permutation, the EM algorithm maximizes θ_i given all $\{x_1, \ldots, x_n\}$ where each x_i is weighted by it's probability of being assigned to that group, $P(z_i|\theta, x)$. The new fully-extended objective function is

$$Q(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{l=1}^{k} \log \mathcal{L}(\theta_l|x_i, z_i = j) P(z_i = j|\theta_l, x_i) \tag{3.4}$$

Since the function is not likely, or can not be-shown, to be convex, the parameter space for θ is searched iteratively, optimizing θ in $\mathcal{L}(\theta|x, z_i = j)$ while fixing the θ in $P(z_i = j|\theta, x)$– $Q(\theta^{t+1}|\theta^t)$; ultimately it is the likelihood that we are trying optimize. We also note that many terms are fixed given i, j, l and therefore do not affect the optimization problem; *i.e.* θ_l is unaffected by z_i if $l \neq j$. The new

$$Q(\theta^{t+1}|\theta^t) = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{l=1}^{k} \mathbb{1}_{j=l} \log \mathcal{L}(\theta_l^{t+1}|x, z_i = j) P(z_i = j|\theta_l^t, x) \tag{3.5}$$

However, this observation alone does not make the problem solvable analytically; instead, manipulation of the underlying likelihood and probability are required so that the log-likelihood becomes a sum of easily-optimizable components. Examples of such manipulations abound for the exponential family since all members have a convex parameter space;

in such cases, the resulting maximum likelihood problem for θ^{t+1} becomes a relatively simple function of x and θ^t. This is repeated until the difference θ^{t+1} - θ^t converges.

Additionally, versions of the EM algorithm exists whose features more closely mirror those of the KCA. In one version, at each iteration, maximization is only conducted on a subset of the parameters, θ. This feature of this variant algorithm, called the Expected Conditional Maximization Algorithm, is also used in our algorithm. Meng and Rubin, the authors of the ECM, note that the ECM is a subset of the Generalized EM, a generalization of the EM algorithm that only requires an improvement in likelihood instead of a full maximization [12], [8]. Once again, the EM and EM-like frameworks only serve as a inspiration for the KCA; many of the EM's properties do not apply to the KCA.

CHAPTER 4

The Algorithm

Here we describe the KCA and its use of KDE's. The algorithm seeks to cluster volatility smiles according to their two-dimensional densities. The following processes are used to achieve this: 1) the necessary metrics characterizing each volatility smile are devised. One of the metrics used is the implied volatility. The other metric, which we refer to as the normalized moneyness, requires further explanation. 2) kernel-density estimation algorithms are discretized and programmed in a manner conducive to the manipulation necessary for the algorithm. 3) and finally the process by which sub-chains are classified and the effect that their classification has on the group's density estimations.

4.1 Normalized Moneyness

The volatility smile is plotted on a two-dimensional space with implied volatility on the vertical axis (y) and some transformation of the strike price on the horizontal axis (x). The volatility is already a normalized quantity. However, in order for a comparison to be made between stocks with different strike prices, a normalization procedure is necessary for the x-axis. The characteristic smile typically has it's lowest point (with respect to the y-axis) where the strike price is closest to the current price of the underlying security. As such, we strive to make this our origin/zero and additionally to normalize the horizontal spread of the volatility smile points so that securities with vastly different absolute prices can have similar smiles

We define the normalized strike price, normalized stock price, and normalized moneyness

as:

$$c_n = \frac{c - c_{\min}}{c_{\max} - c_{\min}} \tag{4.1}$$

$$x_n = \frac{x - c_{\min}}{c_{\max} - c_{\min}} \tag{4.2}$$

$$m_n = x_n - c_n = \frac{x - c}{c_{\max} - c_{\min}} \tag{4.3}$$

where $c_{\min}$ and $c_{\max}$ is the lowest and highest strike prices for expiration date t, on the Monday (for ease of notation, we will occasionally refer to this as day$_0$) following the expiration date $t - 2$, where t simply denotes the standard expiration dates in order. c_n clearly extends from 0 to 1; the range for x_n is not known *a priori*.

To further understand the choice of spread, we detail some characteristics of how the CBOE issues options. For the CBOE's standard equity options, strike prices for a new chain are set after the expiration of a current chain so that at any time, four distinct expiration dates are available. Depending on the current month and the quarterly cycle to which a security is assigned, the sub-chain second-closest to maturity may begin to be offered two months prior to expiration. If at any point, the underlying security's price exceeds the top or bottom bounding strike prices, options with new strike prices are added. Whether or not this is the first day that a sub-chain is traded, we assume that the CBOE procedure does not vary greatly or systematically between option chains; presumably, the sub-chain's range is set as a function of the expected movement of the underlying security's value.

Given the presumed nature of the CBOE's procedures, all smiles now share relatively similar horizontal ranges and points with certain features. Namely, the low point will typically be near zero, and relatively extreme values of -1 and 1 are achieved when the value of the underlying stock is equal to the lowest or highest (respectively) strike prices. m_n values with absolute value above 1 are possible given that new strike prices maybe added after day$_0$.

It is possible that if the sub-option had already existed for one or two months and the underlying security experienced consistent gains or losses, the majority of the sub-chain will be located to the left or right of zero. Although this may merit further investigation, we assume that the effects of these scenarios are minimal (due to a reduced probability of

sustained gains or losses) and offset by each other.

4.2 Kernel Density Estimation

Kernel density estimation (KDE) is a non-parametric method for estimating the density of a distribution [7]. Using KDE to describe volatility smiles provides the flexibility to capture idiosyncrasies that parametric densities could not easily capture; although a comparison with complex mixture models and/or other non-parametric methods would be warranted.

Given a set of n points, $\{x_1, \ldots, x_n\}$, the kernel density estimate is

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i), \tag{4.4}$$

where K is a kernel function. In practice, particularly for use in visualizations, a KDE can be discretized; a grid is created for the domain of interest and for each area a representative point has its density calculated. This process is used to create the visualizations of KDEs; where a maximum detail bound exists because of screen and printing resolutions. If the $K = N(\mu, \sigma)$, the visualization amounts to discretizing the normal distribution and adding the respective amounts to the grid bins surrounding the points x_i. As an example, discretizing the normal distribution into 3 bins can be achieved by raising the central bin by 0.68 units, and each of the adjacent bins by 0.16. The width of each bin is 2σ and the 0.2% beyond $\pm 3\sigma$ can be ignored. The denominator for the resulting probability mass function is simply n. It is exactly this process that we take advantage of for modifying our KDEs.

Since the addition/subtraction of a sub-chain to a KDE of sub-chains adds/removes mass in a predictable fashion (the discretization of the normal distribution only needs to be performed once), KDE modification can be performed at low computational cost. Additionally, calculation of the likelihood/joint-probability is also simplified; if a sub-chain contains n points, the values for n bins are simply retrieved and multiplied together. Any loss of precision due to binning can minimized by treating the bin size as a parameter and cross-validating.

Algorithm Kernel Density Estimate Clustering Algorithm

```
Randomly initialize {z_1, ··· , z_n} where z_i ∈ {1, ··· , K}
loop (t ∈ ℕ)
    for all i ∈ random permutation of {1, . . . , n} do
        o_k ← Q(θ^{t+1} = k|θ^t) for k ∈ {1, . . . , K}
        k_max ← argmax_k o_k
        if k_max ≠ z_i then
            z_i ← k_max
            Break for all
        end if
    end for
    Break loop                                  ▷ Only reached if all i are attempted
end loop
```

4.3 Kernel Density Estimation Clustering Algorithm

We now have two continuous metrics, implied volatility and normalized moneyness, as well as a method for representing the non-parametric densities for the cluster groups. Calculating the metrics for each option in each sub-chain, we can plot a volatility smile for each sub-chain and create two-dimensional KDEs for any group of sub-chains. The next step is determining the group-membership.

Group membership is determined by the using our KDE Clustering Algorithm (KCA), a combinatorial optimization algorithm. We begin by randomly assigning each observational unit (from here on, simply "observation"), a sub-chain, to one of K groups. Each group has a KDE calculated and stored. An observation is then randomly chosen to have its contribution to the KDEs changed and the objective function values are computed for the K groups. All of the possible changes can be represented in graph. In the graph, vertices represent vectors of discrete choices (represented by integers) and an edge between a pair of vertices represents that one vertex can become the other by manipulating a subset of parameters; *i.e.* if one parameter change is allowed per iteration, the vector $(1, 1, 1)$ has an edge $(1, 1, 0)$, but not

$(1, 0, 0)$.

Among the vertices that can be reach from our current position, if there is an increase of the objective function as a result of the assignment of the chosen observation to a group other than its original one, the change is made and a new iteration is begun. If no increase is possible, another observation is randomly chosen. Representing this in a graph is only slightly more complex and does not result in any difference in implementation (one could simply add weights to the graph's paths as well as a variable that determines the parameter of optimization. A path between vertices that differ only by their parameter of optimization would have 0 weight and the search could move freely between them). The algorithm exits when all n observations can no longer be moved without causing a decrease in the objective function value. The algorithm is written is psuedocode on page 13. Although we describe the algorithm with the possibility of one parameter/observation-class class being changed, multiple parameters can be changed at a time.

The KCA takes inspiration from the EM/EMC algorithms since like them, we are confronted with the circular problem of specifying group membership based on the KDEs that are themselves a function of group membership. Clustering with the EM algorithm often involved the use of mixture models where a small set of parameters are optimized in a continuous parameter space. The EM's ability to maximize a localized optimization problem while continuing to increase the complete likelihood can be viewed as its most important features. When the parameter space is a discrete one, much of this complexity can be done away with. Instead the influence of the EM algorithm comes from its use of soft assignment. Specifically, the objective function consisting of a weighted log-likelihood where the weights are "soft" or "fractional" assignments rather then hard assignments is the key. The counterpart of the objective function detailed in equation 3.5 (p. 8), is

$$Q(\theta) = \sum_{i=1}^{n} \log \mathcal{L}(\theta_i | x_i, \theta_{-i}) P(x | \theta_i, \theta_{-i}) \tag{4.5}$$

where θ_{-i}, all parameters except for the ith, is fixed in both the likelihood and the probability because we are maximizing the conditional expectation. i is the observation index that is selected randomly. The random order in which we choose the parameter to optimize make

this algorithm a stochastic optimizer. Whereas the likelihood for EM cannot be maximized without fixing the conditional probability, the finite parameter space of the KCA allows for the updating of both simultaneously.

The comparison to the EM algorithm was initiated for heuristic purposes; in reality, this algorithm (better yet, its objective function) should be framed in an information theoretic context. If hard-assignments were to be made, KCA would more closely resemble the K-means algorithm. Kearns, et. al., analyze the difference between the soft and hard assignments of the EM algorithm and the K-means algorithm [9]. Among the differences, K-means minimizes distortion and this leads to a trade-off between the balance of class membership and group accuracy. Kearns, et. al., also state that K-means will consistently find groups that have less overlap (in fact, this phenomenon was observed during early testing when both soft and hard assignment were tried).

Finally, momentarily ignoring the conditioning that we perform, it is easy to see that for fixed θ,$Q(\theta) = \sum \log\ \mathcal{L}(\theta|x)P(x|\theta) = \sum \log \mathcal{L}(\theta|x)\mathcal{L}(\theta|x)$ is essentially the negative entropy of the group-memberships of each observation (remember, together x and θ define a set of surfaces). In the context of this interpretation, the KCA can be viewed as trying to reduce the entropy of the group classifications. Since KDEs are non-parametric by definition, it is only by parameterizing the KDEs using the inclusion/exclusion of points that we are even afforded a sample space to traverse.

CHAPTER 5

Analysis

Option strike prices, bid/ask prices, and other necessary characteristics were obtained from the Wharton Research Database Service (WRDS) [15]; We will deal exclusively with American Call options. Merton shows that the BS formula can be applied to American calls options since the the optimal decision exercise point will continue to be at the expiration date (granted, he states that this applies to stocks that do not pay a dividend, we proceed despite using stocks that do pay dividends) [13]. Chicago Board of Exchange (CBOE) options on New York Stock Exchange (NYSE) securities are used. Close prices are used to value the securities. The value of an option is taken to be the mean of the best bid and the best offer. The prices of the underlying securities were also obtained from WRDS and the London Interbank Offered Rate (LIBOR) (for $/USD; 12 mo. rate) was procured from EconStats [16][5]. The LIBOR is a benchmark interest rate, the average rate at which prominent London banks lend to each other; it is used as the rate of risk-free borrowing/lending [10] (U.S. Treasury Bill rates are also used in this capacity [3] [14]; differences resulting from the use of one risk-free rate over the other were not examined).

Option chains were retrieved for stocks of the NYSE. Initially, stock prices and option chains were obtained for 846 distinct stocks for the year 2011. To reduce the computational burden, a sample of 40 stock tickers was taken and used for the analysis; the tickers are listed in table 5.1 (22). The expiration dates and trading dates were also reduced. Three standard expiration dates are used (standard, denoting options that expire the Saturday following the third Friday of each month). These are: March 19, 2011; April 16, 2011; and May 21, 2011. The trading dates used are the weeks of March 7 and April 4. Options for the first expiration date are no longer available during the second period. For the majority

of securities and standard expiration dates, options do not begin trading until the Monday following the expiration of the contract that precedes the expiration date by two months. For this reason, during the first period (week of March 7), options expiring on May 21, 2011 are not yet available (where these options are available, they were purposefully excluded). As such, the combination of ticker, trading date, and expiration date, results in 800 sub-chains. In total there are 9,876 option quotes for an average of approximately 12.5 quotes per sub-chain.

Given the experimental nature of the algorithm, relatively small values were chosen for the amount of clusters and bins, although these should be tested at different levels. K is set to 3, the x-axis is divided into 29 equally-wide bins ranging from -1 to 1, and the y-axis is divided into 29 bins ranging from 0 to 5. The kernel is defined as follows,

$$K(x,y) = \begin{cases} 1, & \text{if } x = x_i \cap y = y_i \\ 0.25, & \text{if } \{x - x_i \in (-1,1) \cup y - y_i \in (-1,1)\} \setminus \{x = x_i \cap y = y_i\} \\ 0, & \text{otherwise,} \end{cases} \tag{5.1}$$

where x and y are the implied volatility ($\hat{v}^2$) and normalized moneyness (m_n), respectively. The addition of each point to a KDE increases the sum of all bins by 3; the sum is necessary to normalize the bin counts so that they add to 1. Pseudocounts of 0.1 were added to each bin so that $P(x|\theta)$ is never 0. Additionally, a bin is added at each end of the range; points are never assigned to these bins but they are added in order to simplify the accounting of the count sums.

The KAC was run 5 times with the aforementioned algorithmic parameters. In the bottom panel of figure 5.1 (p. 19) we can see the log-likelihoods of the randomly initiated groupings and their gradual improvements as the iterations pass. Although the decrease in log-likelihood appears to occur at a fairly similar rate for each of the 5 runs, multiple runs should be utilized to mitigate any sensitivity to initialization. The extent of this sensitivity is unknown; during the earlier iterations, few indices need to be tried before an improvement in log-likelihood is found. The top panel of figure 5.1 shows the amount of indices that were searched during each iteration. Before the 400th iteration, the searches rarely require more

than 5 attempted indices. By the 500th iteration, a 5-wide moving average rarely dips below double digits and after the moving average breaks triple digits, a local minimum is usually found in around 20 iterations. The total amount of attempted/successful switches vary from a low of approximately 8e3 to as much as 18e3. More than half of the switches occur in the last 10% of iterations and the last iteration alone is responsible for between 5% and 10% of switches (recall, the condition for convergence is that all potential switches result in a decrease in likelihood).

We now proceed to analyze the groupings created by the clustering. The run with the highest likelihood, represented by the triangle in figure 5.1, produces the densities in 5.2 (p. 20). As expected, all 3 densities have high density in the area of low IV and zero normalized-moneyness; the difference between the 3 groups appear at the ends of the smiles. Group 1, at the top of 5.2 (p. 20), has the flattest smallest volatility smile, that is to say, the average difference between an option deep in-the-money (or out-of-the-money) and an option with 0 normalized moneyness is relatively low. Group 3 has a larger average difference and group 2 is somewhere between group 3 and group 1.

As a demonstrate of the utility creating these groups, a decision tree is created using only a handful of the covariates retrieved for the analysis. The resulting tree is shown in figure 5.3 (p. 21) and makes use of the variables representing days until maturity and normalized stock price, x_n, labeled as V4 and V7, respectively. Some of the splits found by the decision tree algorithm are consistent with already known dynamics of the implied volatility skew; others merit further analysis: Group 1, the group with the flattest smile, contains more sub-chains that are further from expiration; among the sub-chains with an less than 24 days until expiration, those that have deviated most from the initial price (moneyness of 0; remember, strikes are essentially set so that security price on day 1 is in the middle of the range), have smiles with higher implied volatility. The large areas covered by in-the-money options for groups 1 and 3 indicate that more groups should be tried; this may potentially split the groups created in this example. In any case, the amount of data should be increased to allow for more patterns to emerge. This may result in observable patterns beyond what is already know about volatility smile dynamics.

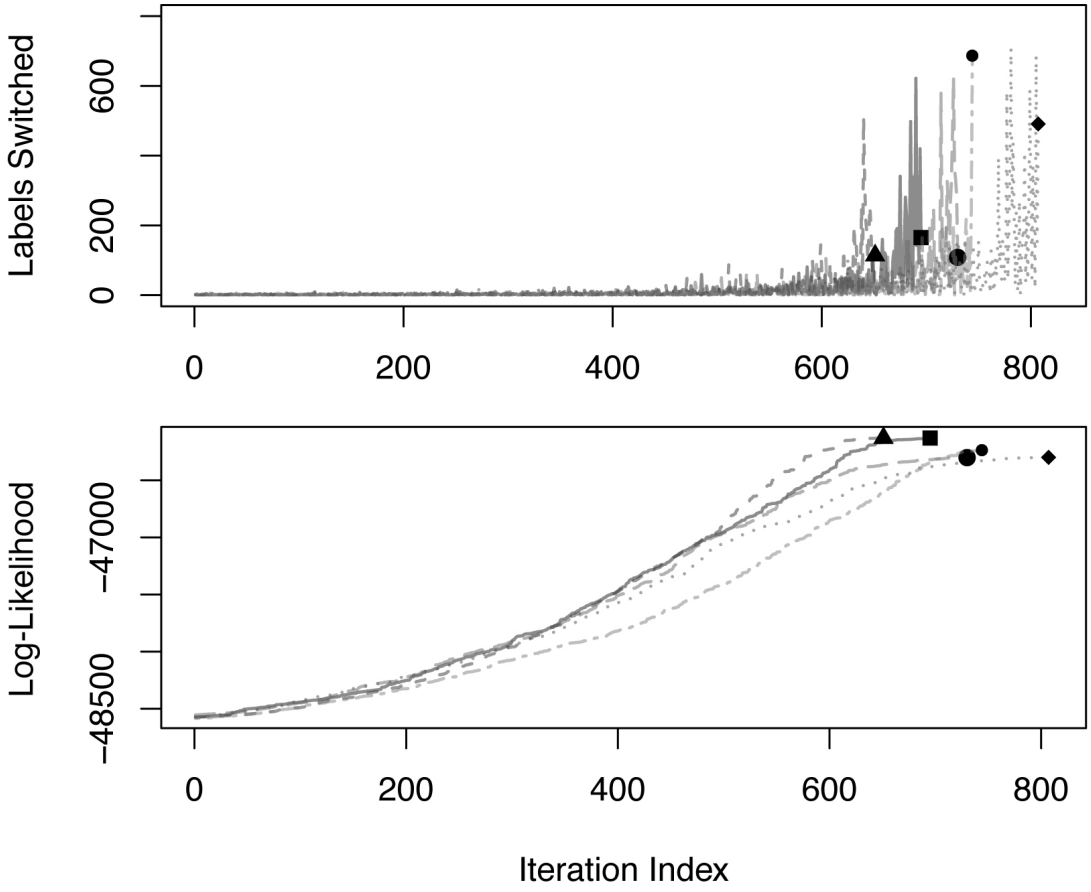

Figure 5.1: Characteristics of KCA convergence for 5 independent runs. The top frame displays the number of attempted parameter modifications per iteration. The bottom frame shows the progressive increase in log-likelihood. The bold points indicate the points of convergence.

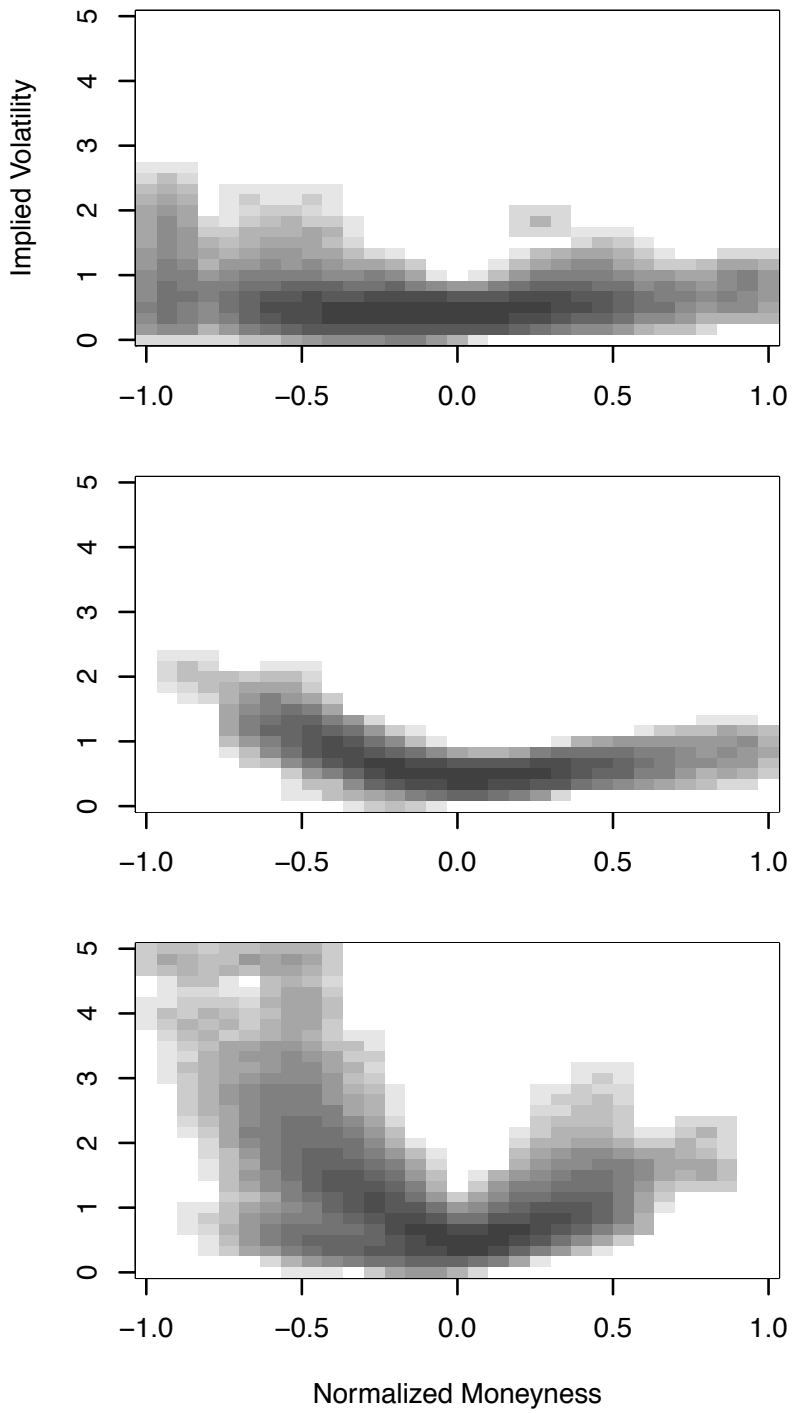

Figure 5.2: Densities for the 3 groups based on the assignments by the KCA.

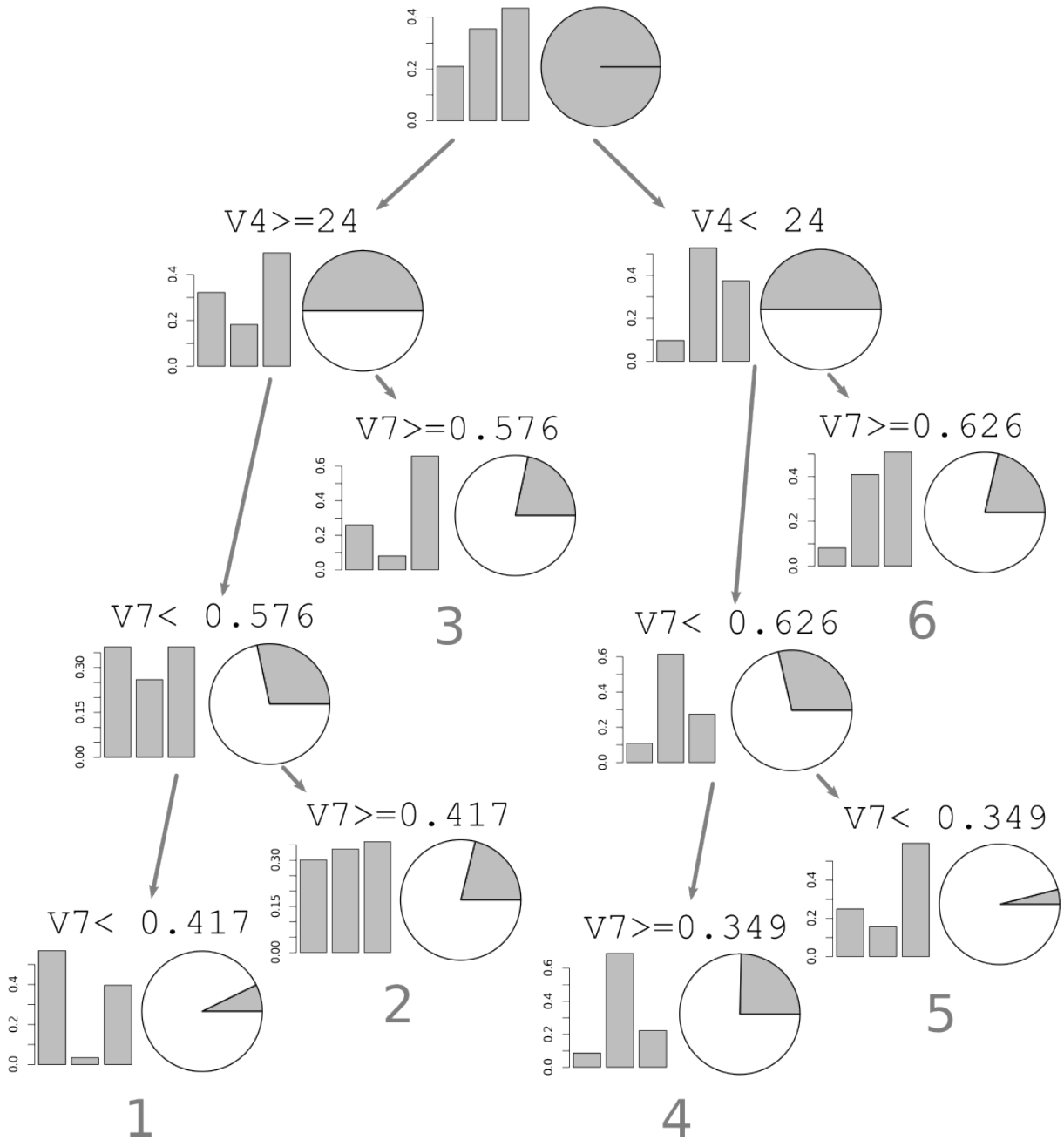

Figure 5.3: Decision tree for predicting group membership. V4 represents the days until the sub-chain's expiration and V7, the normalized price for the underlying security.

ACT	BCO	BP	BTU	CBT	CNQ	CRS	CVC
DDR	DLX	ECA	ELX	ESI	FHN	GB	GME
GSK	HFC	ING	JPM	NBL	NLY	ORB	OSK
PGR	PHG	PHH	PQ	PRU	REG	RF	ROC
ROG	SM	TKR	TTC	UPS	VAR	WCC	WMT

Table 5.1: List of included NYSE stocks

CHAPTER 6

Conclusion

Much of the previous research on implied volatility has focused on the forecasting of the future implied volatility of a single security given the data of its recent volatility. The relationship among the options in the same sub-chain is largely ignored. Nowhere is this more evident then in research that produces weighted implied volatilities based on their options' moneyness. A conclusion of this literature has been to simply use the implied volatility of the option with strike price closest to the underling's price, discarding potentially valuable data [11]. By using the sub-chain as the observational unit that we cluster with KCA, we are able to analyze the similarities between smiles for the security that posses different or similar trading dates, expiration dates, industries, and many other attributes associated with the underlying securities. The latent groupings may provided additional insight based on this very data discarded by these aforementioned forecasting methods. Kernel density estimates were chosen to describe volatility smiles because of the ease with which they can capture slight and unexpected features.

The use of kernel density estimates demands a new framework; the formulated iterative scheme shows some promising results but there are still many details left to be explored. The choice of $k = 3$ and the simple kernel function were used for convenience but would certainly merit further inspection. Just as for other iterative algorithms, the choice of k should be chosen via cross-validation; the number of bins used in the discretization of the density and kernel function should likewise be cross-validated, although here the optimal choice may not be as clear. Additionally, throughout this document we focus on a single run of the KCA algorithm, in practice one would be advised to run the algorithm multiple times given the sensitivity to initialization and the potential for the algorithm to get stuck in local maxima.

Neither of these problems are unique to the KCA, the EM algorithm and K-means are also sensitive to initialization, although the degree to which the KCA suffers from this should be examined. One might suspect that due to the graphical nature of the optimization and large amount of parameters these problems would be pronounced in the KCA, but visual inspection of the resulting densities from various runs showed fairly consistent results. Measuring the consistency of results could also be formalized via the use of Kullback-Leibler divergence or by analyzing the tendency for certain observations to be grouped together. Also, we should not dwell too much on the similarities and differences between the KCA and other iterative methods; ultimately they are solving distinct problems: clustering using mixture models finds groupings of points; the KCA groups densities. Instead, competing non-parametric frameworks could be developed, such as a clustering algorithm based on splines.

Finally, the use of non-parametric techniques does create a computational burden, but the construction of the KCA allows for a number of optimizations. First, the cost of adding and removing an observation from a pair of densities can be reduced by 1) having the densities of each observation pre-calculated and stored, and 2) computing deltas for the new likelihoods by storing the set of observations that are affected by changes in each individual bin. The leveraging of multithreaded programming can also provide a reduction in the time required for a run. At the very least, each iteration consists of reassigning an observation to one of $k-1$ groups and selecting the assignment with the highest likelihood. These calculations can be performed in parallel. In the later iterations of a run, multiple observation need to be searched before an increase in likelihood is found and the iteration proceeds. For this, a process can be devised so that latter iterations take advantage of unused processor time to test multiple observations at a time. The addition of these parallelizations could make the KCA accessible even for very large datasets and the flexibility of KDEs would allow researchers to capture differences that are missed by other methods.

References

[1] Black, F. and Scholes, M. (1973), "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy*, 81, 637-654.

[2] Cont, R. and Jose da Fonseca, J. (2002), "Dynamics of implied volatility surfaces," *Quantitative Finance*, 2, 45-60.

[3] Day, T. E. and Craig L. M. (1992), "Stock market volatility and the information content of stock index options," *Journal of Econometrics*, 52, 267-287.

[4] Dempster, A. P. , Laird, N. M. , and Rubin, D. B. (1977) "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society* Ser B, 39, 1-38.

[5] EconStats. (2013). LIBOR in US Dollar, 1987-2013 [Data]. Retrieved from http://www.econstats.com/r/indexrlib.htm

[6] Equity Option Specs. (2013). CBOE, http://www.cboe.com/Products/EquityOptionSpecs.aspx

[7] Hastie, T. , Tibshirani, R. , and Friedman, J. (2009) *The Elements of Statistical Learning*, 2 E, New York: Springer, 208-209.

[8] Hinton, G. (1999),"A view of the EM algorithm that justifies incremental, sparse, and other variants," *Learning in graphical models*, Netherlands: Springer, 495-520.

[9] Kearns, M. , Mansour Y. , and Ng. A. Y. (1998), "An information-theoretic analysis of hard and soft assignment methods for clustering," *Learning in graphical models*, Netherlands: Springer, 495-520.

[10] Kou, S. G. (2002) "A Jump-Diffusion Model for Option Pricing," *Management Science*, 48, 1086-1101.

[11] Mayhew, S. (1995), "Implied Volatility," *Financial Analysts Journal*, 51, 4, 8-20.

[12] Meng, X. and and Rubin, D. B. (1993), "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika* 80, 2, 267-278.

[13] Merton, R. C. (1973), "Theory of Rational Option Pricing," *The Bell Journal of Economics and Management Science*, 4, 141-183.

[14] Larry J. Merville, L. J. and Dan R. Pieptea, D. R. (1989) "Stock-Price Volatility, Mean-Reverting Diffusion, and Noise," *Journal of Financial Economics*, 24, 193-214.

[15] OptionMetrics Ivy DB. (2013). Option Information Datasets, 1996-2013. [Data]. Retrieved from Wharton Research Data Service.

[16] OptionMetrics Ivy DB. (2013). Security Information Datasets, 1996-2013. [Data]. Retrieved from Wharton Research Data Service.

[17] Shephard, N. (2005), "Stochastic Volatility," unpublished manuscript, Nuffield College, Economics Group.