FinM 331/Stat 339 Financial Data Analysis, Winter 2010

Floyd B. Hanson, Visiting Professor

Email: fhanson@uchicago.edu

Master of Science in Financial Mathematics Program
University of Chicago

Lecture 7

6:30-9:30 pm, 15 February 2010, Ryerson 251 in Chicago

7:30-10:30 pm, 15 February 2010 2010 at UBS in Stamford

7:30-10:30 am, 16 February 2010 at Spring in Singapore

* **Gong Xi Fa Cai!**

Unfortunately conflicts with Second Day of the Chinese New Year holiday in Singapore, but this is an international program subject to many holidays. Sorry.

7. Method of Moments, Options, Calibration, Implied Volatility, and NonParametric Regression:

- ***7.1. Method of Moments:***

The **method of moments estimation (MME)** of parameters $\vec{p} = [p_i]_{1\times m}$ of a given distribution $F_{\mathbf{X}}(x;\vec{p})$ of the RV $\mathbf{X}$ by matching sample moments to true moments of $F_{\mathbf{X}}(x;\vec{p})$. If there are **m** parameters, then a least **m** moments are ideally required,

$$\mu_{\mathbf{X}}^{(k)}(\vec{p}) = \mathrm{E}[X^k] \ \ for \ \ k = 1\!:\!m, \tag{7.1}$$

where, for example, $\mu_{\mathbf{X}}^{(1)}(\vec{p}) = \mu$ the usual distribution mean.

Let $\vec{X} = [X_i]_{1\times n}$ be a sample of IID RV observations from the assumed known distribution, then the estimated sample means are

$$\widehat{\mu}_n^{(k)} = \frac{1}{n}\sum_{i=1}^{n} X_i^k \ \ for \ \ k = 1\!:\!m, \tag{7.2}$$

where the sample mean $\widehat{\mu}_n = \widehat{\mu}_n^{(1)}$ and the biased sample variance is $S_n^2 = \widehat{\mu}_n^{(2)} - \widehat{\mu}_n^2$.

The estimates of the parameters $\widehat{p}_n \simeq \vec{p}$ are determined by the moment method equations, assuming a solution exists for m equations in m unknowns,

$$\mu_X^{(i)}(\widehat{p}_n) = \widehat{\mu}_n^{(i)} \ \ for \ \ i = 1:m \tag{7.3}$$

∘ *7.1.1. Poisson Moment Method Example:*

Suppose the population distribution is a static Poisson so that the RV $X \stackrel{\text{dist}}{=} P(\Lambda)$ with single parameter $\Lambda = \mathrm{E}^{(p)}[X]$. Let the $\vec{X} = [X_i]_{1\times n}$ be the **Poisson IID (IIPD) sample**, then estimated MM Poisson parameter is

$$\widehat{\Lambda}_n = \overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{7.4}$$

The estimated parameter $\widehat{\Lambda}_n$ is an RV that is Poisson distributed, since

$$\mathrm{Prob}\Big[\widehat{\Lambda}_n = K\Big] = \mathrm{Prob}\left[\sum_{i=1}^{n} X_i = nK\right] = e^{-n\Lambda}\frac{(n\Lambda)^{nK}}{(nK)!}, \tag{7.5}$$

as will be shown. Knowing the estimate distribution is needed to estimate the error of the estimate.

This can shown directly by first considering the $n=2$ as a **discrete convolution**,

$$\begin{aligned}
\mathrm{Prob}[X_1+X_2=2K] &\overset{\mathrm{iid}}{\underset{*}{=}} e^{-2\Lambda}\sum_{k_1=0}^{\infty}\frac{\Lambda^{k_1}}{k_1!}\sum_{k_2=0}^{\infty}\frac{\Lambda^{k_2}}{k_2!}\mathcal{I}_{\{k_1+k_2=2K\}} \\
&\overset{\mathrm{k_1\le 2K}}{=\!=} e^{-2\Lambda}\sum_{k_1=0}^{2K}\frac{\Lambda^{k_1}\Lambda^{2K-k_1}}{k_1!(2K)!} \\
&= e^{-2\Lambda}\frac{\Lambda^{2K}}{(2K)!}\sum_{k_1=0}^{2K}\binom{2K}{k_1} \\
&\overset{\mathrm{bin}}{=} e^{-2\Lambda}\frac{\Lambda^{2K}}{(2K)!}(1+1)^{2K} \\
&= e^{-2\Lambda}\frac{(2\Lambda)^{2K}}{(2K)!},
\end{aligned} \tag{7.6}$$

so the rest for general n follows by **induction though binomial additivity.**

It is nice to be able to derive something through first principles as on the prior page, but there is a simpler way using the **moment generating function (MGF)** which has the advantage of easily decomposing up the IID RV parts. Since for a single RV when $n = 1$, the exponential transform MGF for the Poisson distribution yields

$$\mathrm{MGF}^{(p)}[X_i] = \mathrm{E}^{(p)}\left[e^{tX_i}\right] = e^{-\Lambda}\sum_{k=0}^{\infty}\frac{\Lambda^k e^{tk}}{k!} = e^{\Lambda\left(e^t-1\right)}, \tag{7.7}$$

an exponential of an exponential. Hence of the IID sum MGF is

$$\mathrm{MGF}^{(p)}\left[\sum_{i=1}^{n} X_i\right] = \mathrm{E}^{(p)}\left[\exp\left(t\sum_{i=1}^{n} X_i\right)\right] \overset{\text{loe}}{\underset{\text{iid}}{=}} \prod_{i=1}^{n} \mathrm{E}^{(p)}\left[e^{tX_i}\right]$$
$$= \prod_{i=1}^{n} e^{\Lambda\left(e^t-1\right)} \overset{\text{loe}}{=} e^{n\Lambda\left(e^t-1\right)}, \tag{7.8}$$

which is the same form as (7.7), except with the parameter Λ replaced by $n\Lambda$ and thus proving (7.5), so $\mathrm{E}^{(p)}\left[\sum_{i=1}^{n} X_i\right] = n\Lambda$. Consequently, we have that $\widehat{\Lambda}_n$ is **unbiased**, i.e.,

$$\mathrm{E}^{(p)}[\widehat{\Lambda}_n] = \frac{1}{n}\mathrm{E}^{(p)}\left[\sum_{i=1}^{n} X_i\right] = \frac{1}{n}n\Lambda = \Lambda. \tag{7.9}$$

Also, the **estimated parameter sample variance** is

$$\mathrm{Var}^{(p)}\left[\widehat{\Lambda}_n\right] = \frac{1}{n^2}\mathrm{Var}^{(p)}\left[\sum_{i=1}^{n} X_i\right] = \frac{1}{n^2} n\Lambda = \frac{\Lambda}{n} = \mathrm{SE}^2\left[\widehat{\Lambda}_n\right] \quad (7.10)$$

and the **standard error** is

$$\mathrm{SE}\left[\widehat{\Lambda}_n\right] = \sqrt{\mathrm{Var}^{(p)}\left[\widehat{\Lambda}_n\right]} = \frac{\Lambda}{\sqrt{n}}, \quad (7.11)$$

which goes to zero as $n \to \infty$. However, $\mathrm{SE}[\widehat{\Lambda}_n]$ can be used as an estimate of the order of the error in $\widehat{\Lambda}_n$ and using the estimate for the standard error gives an **actual estimate of the standard error**, i.e.,

$$\mathrm{SE}\left[\widehat{\Lambda}_n\right] \simeq \frac{\widehat{\Lambda}_n}{\sqrt{n}}. \quad (7.12)$$

As $n \to \infty$, the $\widehat{\Lambda}_n$ can be asymptotically **approximated by a normal distribution** using the central limit theorem,

$$\widehat{\Lambda}_n \overset{\text{dist}}{\sim} \mathcal{N}(\Lambda, \Lambda/\sqrt{n}) \quad (7.13)$$

and can be approximated by the 2-sigma or 95% confidence interval rule,

$$\widehat{\Lambda}_n \simeq \Lambda(1 \pm 2\widehat{\Lambda}_n/\sqrt{n}). \quad (7.14)$$

∘ *7.1.2. Normal Moment Method Example:*

Now suppose we have a normal distribution $\mathcal{N}(\mu, \sigma)$ and an sample of n **IID normal (IIND) observations** $\vec{X}$. The moment method involves two parameters $\vec{p} = (\mu, \sigma^2)$ and two true moments, the first moment

$$\mu_X^{(1)}(\vec{p}) = \mathrm{E}^{(n)}[X] = \mu \tag{7.15}$$

and the second moment

$$\mu_X^{(2)}(\vec{p}) = \mathrm{E}^{(n)}[X^2] = \sigma^2 + \mu^2. \tag{7.16}$$

As before the **mean estimate** is

$$\widehat{\mu}_n \equiv \overline{X}_n, \tag{7.17}$$

but now the **variance estimate** is

$$\widehat{\sigma}_n^2 \equiv \overline{(X^2)}_n - \overline{X}_n^2. \tag{7.18}$$

The **mean estimate is unbiased** since

$$\mathrm{E}^{(n)}[\overline{X}_n] = \mathrm{E}^{(n)}[\widehat{\mu}_n] = \frac{1}{n}\mathrm{E}^{(n)}[X_i] = \mu. \tag{7.19}$$

The **variance of the mean estimate** is then

$$\begin{aligned}\mathrm{Var}^{(n)}[\widehat{\mu}_n] &= \mathrm{E}^{(n)}\left[(\overline{X}_n - \mu)^2\right] \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{E}^{(n)}[(X_i-\mu)(X_j-\mu)] \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sigma^2 = \frac{\sigma^2}{n} = \mathrm{SE}^2[\widehat{\mu}_n]\end{aligned} \tag{7.20}$$

and the **standard error of the mean estimate** is

$$\mathrm{SE}[\widehat{\mu}_n] = \frac{\sigma}{\sqrt{n}}. \tag{7.21}$$

It can also be shown that

$$\widehat{\mu}_n = \overline{X}_n \stackrel{\text{dist}}{=} \mathcal{N}(\mu, \sigma/\sqrt{n}). \tag{7.22}$$

so **normally distributed for <u>any n</u>, not just asymptotically normal distributed**.

For the normal distribution $F_X^{(n)}(x;\mu,\sigma^2)$, the MGF for a single IIND RV is

$$\begin{aligned}\mathrm{MGF}^{(n)}[X_i] = \mathrm{E}^{(n)}\left[e^{tX_i}\right] &= \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty} dx\, e^{-\frac{(x-\mu)^2}{2\sigma^2}+tx} \\ &= e^{\mu t+\sigma^2 t^2/2},\end{aligned} \tag{7.23}$$

so the for the sample mean,

$$\begin{aligned}\mathrm{MGF}^{(n)}[\overline{X}_n] = \mathrm{E}^{(n)}\left[e^{t\sum_{i=1}^n X_i/n}\right] &\overset{\text{ind}}{=} \prod_{i=1}^n \mathrm{E}^{(n)}\left[e^{tX_i/n}\right] \\ &= \prod_{i=1}^n e^{\mu t/n+\sigma^2(t/n)^2/2} = e^{\mu t+(\sigma^2/n)t^2/2},\end{aligned} \tag{7.24}$$

justifying the estimated mean distribution in (7.22).

Further, Rice (2007; p. 263 with cross-references) shows that the estimated variance scaled by $\mathrm{SE}^2[\widehat{\mu}_n]$ behaves as a **Chi-squared** distribution with $n-1$ degrees of freedom, i.e.,

$$\frac{\widehat{\sigma}_n^2}{\sigma^2/n} \overset{\text{dist}}{\sim} \chi_{n-1}^2. \tag{7.25}$$

○ 7.1.3. Gamma Moment Method Example:

The **gamma distribution** is a generalization of the exponential distribution. Its density on $(0, \infty)$ is given by

$$f_X^{(g)}(x;\lambda,\alpha) = \frac{\lambda^\alpha x^{\alpha-1}}{\Gamma(\alpha)} e^{-\lambda x}, \quad (7.26)$$

where $\alpha > 0$ and $\lambda > 0$ for integrability, while its normalization depends on the **gamma function** $\Gamma(\alpha)$

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx, \quad (7.27)$$

such that $\Gamma(1) = 1$ and $\Gamma(\alpha+1) = \alpha\Gamma(\alpha) = \alpha!$. The true moments can be deduced from the gamma function MGF, letting X be a gamma distributed IID RV and $t < \lambda$, so from the MGF moment coefficients,

$$MGF^{(g)}[X] = \mathrm{E}^{(g)}[e^{tX}] = \frac{\lambda^\alpha}{\Gamma(\alpha)} \int_0^\infty x^{\alpha-1} e^{-(\lambda-t)x} dx$$
$$= \frac{\lambda^\alpha}{\Gamma(\alpha)(\lambda-t)^\alpha} \int_0^\infty y^{\alpha-1} e^{-y} dy = \frac{1}{(1-t/\lambda)^\alpha} \quad (7.28)$$
$$\sim 1 + \frac{\alpha t}{\lambda} + \frac{\alpha(\alpha+1)t^2}{2\lambda^2} = 1 + \mu_X^{(1)}(\vec{p})t + \mu_X^{(2)}(\vec{p})\frac{t^2}{2}.$$

Reading the moments from the expansion in the last line of (7.28), we have the true the first moment,

$$\mu_X^{(1)} = \mathrm{E}^{(g)}[X] = \frac{\alpha}{\lambda} \tag{7.29}$$

and the second moment

$$\mu_X^{(2)} = \mathrm{E}^{(g)}[X^2] = \frac{\alpha(\alpha+1)}{\lambda^2}. \tag{7.30}$$

Since the moment method produces moments first and the parameters can be derived second, we need the inverse of the nonlinear relationship between the distribution defined parameters and the moments,

$$\lambda = \frac{\mu_X^{(1)}}{\mu_X^{(2)} - \left(\mu_X^{(1)}\right)^2} \tag{7.31}$$

and

$$\alpha = \lambda \mu_X^{(1)} = \frac{\left(\mu_X^{(1)}\right)^2}{\mu_X^{(2)} - \left(\mu_X^{(1)}\right)^2}. \tag{7.32}$$

Letting $\vec{X} = [X_i]_{1\times n}$ be a set of **IID gamma distributed (IIGD) observations**, the parameter estimates are

$$\widehat{\lambda}_n = \frac{\widehat{\mu}_n}{\widehat{\mu}_n^{(2)} - (\widehat{\mu}_n)^2} = \frac{\widehat{\mu}_n}{\widehat{\sigma}_n^2} \tag{7.33}$$

and

$$\widehat{\alpha}_n = \frac{\widehat{\mu}_n^2}{\widehat{\sigma}_n^2}. \tag{7.34}$$

{Remark: Due to the complexity of estimated parameters, $\widehat{\lambda}_n$ and $\widehat{\alpha}_n$, in relation to the estimated moments, $\widehat{\mu}_n$ and $\widehat{\mu}_n^{(2)}$, finding the distribution of the parameter estimates for estimating the parameter errors is too difficult. Rice (2007, pp. 263-266) suggests bootstrapping a large family of samples $\{\vec{X}_j : j = 1 : M\}$ to simulate empirical distributions of $\widehat{\lambda}_n$ and $\widehat{\alpha}_n$, using that to estimate the errors.}

However, the parameter distribution of the moment estimates can be calculated using the MGF, but the product distribution needs to be generated. Hence, for the sample mean $\widehat{\mu}_n = \overline{X}_n$,

$$\begin{aligned}\mathrm{MGF}^{(g)}[\overline{X}_n] = \mathrm{E}^{(g)}\left[e^{t\sum_{i=1}^{n} X_i/n}\right] &\stackrel{\text{ind}}{=} \prod_{i=1}^{n} \mathrm{E}^{(g)}\left[e^{tX_i/n}\right] \\ &= \prod_{i=1}^{n} \frac{1}{(1-t/(n\lambda))^{\alpha}} = \frac{1}{(1-t/(n\lambda))^{n\alpha}},\end{aligned} \tag{7.35}$$

so

$$\widehat{\mu}_n = \overline{X}_n \stackrel{\text{dist}}{\sim} F_X^{(g)}(x; n\lambda, n\alpha). \tag{7.36}$$

Thus

$$\mathrm{E}^{(g)}[\overline{X}_n] = \frac{\alpha}{\lambda}, \tag{7.37}$$

$$\mathrm{E}^{(g)}\left[\overline{X}_n^2\right] = \frac{n\alpha(n\alpha+1)}{(n\lambda)^2} = \frac{\alpha}{\lambda}\left(\frac{\alpha}{\lambda} + \frac{1}{n\lambda}\right) \tag{7.38}$$

and

$$\mathrm{Var}^{(g)}[\overline{X}_n] = \frac{\alpha}{\lambda}\left(\frac{\alpha}{\lambda} + \frac{1}{n\lambda}\right) - \left(\frac{\alpha}{\lambda}\right)^2 = \frac{\alpha}{n\lambda^2}. \tag{7.39}$$

The standard error of the estimated first moment is then,

$$\mathrm{SE}\left[\overline{X}_n\right] = \frac{1}{\lambda}\sqrt{\frac{\alpha}{n}}. \qquad (7.40)$$

∘ *7.1.4. Consistent Parameter Estimates:*

Definition of Consistency: *A parameter estimate $\widehat{p}_n$ of a single parameter p from sample size n is a **consistent estimate** if $\widehat{p}_n$ converges in probability to p as $n \to \infty$,*

$$\mathrm{Prob}[|\widehat{p}_n - p| > \varepsilon] \to 0 \quad as \quad n \to \infty, \qquad (7.41)$$

for any $\varepsilon > 0$.

The weak law of large numbers is support for this definition. Conisistency in probability justifies the use of the standard error of the estimate, $\mathrm{SE}[\widehat{p}_n] = \sigma/\sqrt{n}$ where $\sigma = \sigma(p)$ is the true standard error. An **approximate form of consistency** is the use of the estimated standard error such that

$$\mathrm{SE}[\widehat{p}_n] \simeq \widehat{\mathrm{SE}}[\widehat{p}_n] \equiv \sigma(\widehat{p}_n)/\sqrt{n}, \qquad (7.42)$$

assuming continuity of $\sigma(p)$ then $\widehat{p}_n \to p$ implies $\sigma(\widehat{p}_n) \to \sigma(p)$, in theory.

{MME versus MLE Remarks: The method of moments parameter estimation (MME) is very old and predates the maximum likelihood method of estimating parameters. Maximum likelihood method estimation was introduced by the legendary statistician R. A. Fisher as an improvement over the method of moments to have a higher probability of closer estimations, as the name maximum likelihood suggests, so MLE is said to be more efficient. The method of moments can produce improper answers with small samples, so large samples are important. The same can be said for a larger number of parameter, since it may be difficult to solve the system of moment equations.}

◦ *7.1.5. Generalized Moment Method:*

The are a number of variants of the ordinary moment method and one was introduced by Lars Peter Hansen (1982) of the University of Chicago called the **generalized moment method (GMM)** which uses a general function of the observation RV X and the parameter $\vec{p}$, with expectation

$$\vec{\mu}(\vec{p}) = \mathrm{E}_X[\vec{g}(X, \vec{p})], \tag{7.43}$$

such that the otherwise general $\vec{g}$ has **mean zero**,

$$\vec{\mu}(\vec{p}) = \mathrm{E}_X[\vec{g}(X, \vec{p})] = \vec{0}, \tag{7.44}$$

in analogy to the critical point conditions of MLE.

The sample mean is the usual,

$$\widehat{\mu}_n(\vec{p}) = \frac{1}{n}\sum_{i=1}^{n} \vec{g}(X_i, \vec{p})] \tag{7.45}$$

and since by the law of large numbers $\widehat{\mu}_n(\vec{p}) \longrightarrow \vec{\mu}(\vec{p})$ as $n \longrightarrow \infty$, leading to the parameter estimation condition,

$$\widehat{\mu}_n(\widehat{p}_n) \simeq \vec{0}. \tag{7.46}$$

The optimal objective is model on the idea of robust, **weighted least squares** with the sample size dependent, positive definite weight matrix W_n such that the **optimal estimated parameter vector** is given by

$$\widehat{p}_n = \underset{\vec{p}}{\operatorname{argmin}}\left[\widehat{\mu}_n^\top(\vec{p}) W_n \widehat{\mu}_n(\vec{p})\right]. \tag{7.47}$$

A good, efficient choice is the inverse of the $\vec{g}$-covariance matrix,

$$\begin{aligned}\mathrm{Cov}\left[\vec{g}(X,\vec{p})\vec{g}^\top(X,\vec{p})\right] &= \mathrm{E}\left[\vec{g}(X,\vec{p})\vec{g}^\top(X,\vec{p})\right] \\ &\simeq \frac{1}{n}\sum_{i=1}^{n} \vec{g}(X_i,\widehat{p}_n^{(0)})\vec{g}^\top(X_i,\widehat{p}_n^{(0)}). \qquad (7.48) \\ &= \overline{\vec{g}\vec{g}^\top}_n^{(0)} \equiv (W_n^{(0)})^{-1},\end{aligned}$$

recalling that $\vec{g}$ has mean zero, where $\widehat{p}_n^{(0)}$ is a starting or current parameter estimate. The parameter estimate using such a weight matrix estmate can be computed using a robust or nonlinear regression or an optimal search method.[a]

[a] A short description of additional conditions and other information can be found at ***Wikipedia: Generalized method of moments.*** For code, go to MATLAB Central File Exchange, ***GMM: gmmestimation.m by Cao Zhiguang (2006).*** Be aware that “GMM” is also used as an acronym for Gaussian mixture model in MATLAB and elsewhere.

PS: Another often used advanced moment method is the **efficient moment method (EMM)** which successively combines the efficiency of the maximum likelihood estimation (MLE) and simulation moment method (SMM). For a compact, readable account and SAS example code see ***Efficient Method of Moments Estimation of a Stochastic Volatility Model.*** The EMM was used in a paper often quoted in class for results showing the importance of both jumps and stochastic volatility: T. G. Andersen, L. Benzoni and J. Lund, “An Empirical Investigation of Continuous-Time Equity Return Models,” *J. Fin.*, vol. 57, 2002, pp. 1239–1284.

- **7.2. Vanilla, European Options:**[a]
  - ***7.2.1. Black-Scholes European Call and Put Options:***

With **delta-hedging**[b] to eliminate the risk due to volatility terms and arbitrage-free conditions restricting portfolio growth to the risk-free rate, r, we are effectively dealing with the modified, underlying stock price $S(t)$ diffusion SDE,

$$dS(t) = S(t)(r dt + \sigma dW(t)), \quad S(0) = S_0, \tag{7.49}$$

where both r and the volatility σ are assumed to be constant, although that is not necessary.

---

[a] In part, adapted from Carmona ('04) Ch. 4.; Hull (6th Ed., '06); D. Higham ('04); Hanson's *Applications in Financial Engineering*, Chapter 10; CBOE's Stock Options brochure *http://www.cboe.com/LearnCenter/pdf/understanding.pdf*.

[b] The Greek delta, $\Delta = \partial C/\partial S$ is used to **hedge** away the uncertainty risk, where for example C is the call option price and S is the stock price.

Black and Scholes (Spring 1973) derived a solution for a **European style call option** contract to buy the stock for a **strike price K** at a specified contract maturity date paying the **call option contract price** or **premium** $C(S(t),t;K,T,r,\sigma)$ at current time t when underlying price $S(t)$, all governed by the final gain or strike payoff function

$$C(S(T),T;K,T,r,\sigma)=G(S(T),K)=\max[S(T)-K,0]. \quad (7.50)$$

Presumably, $G(S_0,K)=0$, i.e., $S_0<K$, otherwise there would be no incentive to sell the contract to the buyer, the buyer betting that the stock price will rise over K and if $G(S(T),K)=0$ the rational buyer would walk away from the contract since the stock could be purchased more cheaply in the market.

Note that index options[a] are different, mainly that there is a cash settlement replacing the opportunity to buy stock at K and thus is closer to real betting.

[a] Again, see CBOE's Understanding Index Options brochure: *http://www.cboe.com/LearnCenter/pdf/understandingindexoptions.pdf* .

Black and Scholes' well-known solution formula uses the solution of the SDE (also a backward problem for a PDE) and is

$$C^{(\mathrm{bs})}(S,t;K,T,r,\sigma) = SF_X^{(\mathrm{n})}(d_1;0,1) \\ -e^{-r(T-t)}KF_X^{(\mathrm{n})}(d_2;0,1), \tag{7.51}$$

where

$$d_1 = d_1(S/K,T-t,r,\sigma) \\ \equiv \big(\log(S/K)+(r+\sigma^2/2)(T-t)\big)/\big(\sigma\sqrt{T-t}\big)\,; \\ d_2 = d_2(S/K,T-t,r,\sigma) \\ = d_1(S/K,T-t,r,\sigma)-\sigma\sqrt{T-t}, \tag{7.52}$$

noting the natural dependence is on the **time to maturity** $\tau \equiv T-t$, also called the **time-to-go**, and the **moneyness**[a] , the ratio $M = M(\tau) \equiv Se^{r\tau}/K$, so that $\log(M) = \log(S/K)+r\tau$ is the **log-moneyness**.

---

[a] Moneyness can also be defined as the reciprocal $K/(Se^{r\tau})$ when the focus is on K as for a put option. See, for instance, Prof. R. Lee's (2004) thorough paper on ***Implied Volatility***.

Note that $Se^{r(T-t)}$ is the **future value** of the current price S to maturity from t compounded at the risk-free rate r or alternately $e^{-r(T-t)}K$ is the **present value** of the strike price K, available at maturity T, but discounted at the risk-free rate r back to present time t[a]. If $M = 1$ then the option is **at the money (ATM)**, else if $M > 1$ then it is **in the money (ITM)** for a call option, else $M < 1$ then it is **out the money (OTM)** for a call option, but ITM for the put option. Note that at exercise, $M = S(T)/K$, so then ITM or ATM mean $K \leq S(T)$. Also, ITM is not the same as in the profit "**ITP**", since that requires ***a net profit or*** $\mathbf{Profit} = S - K - \mathbf{Premium} > 0$ for the call option.

[a]In general, the discount rate β is different than the interest rate r as the present value differs from the future value. They both can depend on time t.

Thus, **for financial and numerical purposes**, we may define a more computational finance form of the call option price

$$\begin{aligned}\widetilde{C}^{(\mathrm{bs})}(M,\tau;\sigma) &\equiv C^{(\mathrm{bs})}\left(e^{-r\tau}KM, T-\tau; K, T, r, \sigma\right)/\left(e^{-r\tau}K\right) \\ &= MF_X^{(\mathrm{n})}(d_1;0,1) - F_X^{(\mathrm{n})}(d_2;0,1),\end{aligned} \tag{7.53}$$

where $d_1 = \log(M)/\widetilde{\sigma} + \widetilde{\sigma}/2$ and $\widetilde{\sigma} \equiv \sqrt{\sigma^2\tau}$ which is the scaled volatility. You can verify that the scaled call price goes to the correct limit as $\tau \longrightarrow 0^+$.

The corresponding **European put option** is a contract to sell stock to the contract maker at K at T under an asymmetric version of the payoff,

$$\mathcal{P}(S(T),T;K,T,r,\sigma) = G(-S(T),-K) = \max[K-S(T),0], \tag{7.54}$$

with solution,

$$\begin{aligned}\mathcal{P}^{(\mathrm{bs})}(S,t;K,T,r,\sigma) = &-SF_X^{(\mathrm{n})}(-d_1;0,1) \\ &+e^{-r(T-t)}KF_X^{(\mathrm{n})}(-d_2;0,1).\end{aligned} \tag{7.55}$$

Note that moneyness for the put should literally be the opposite of money for the call, i.e. $1/M$ makes more sense for the put option.

Equations (7.51) and (7.55) connect a maximum and replicated portfolio derivation called the **put-call parity**,

$$\mathcal{P}^{(\mathrm{bs})}(S,t)+S=C^{(\mathrm{bs})}(S,t)+e^{-r(T-t)}K, \tag{7.56}$$

suppressing parameter arguments.

However, as we have previously discussed, the Black-Scholes model, despite its extensive service in quantitative finance for almost 37 years, has many deficiencies, like unrealistic constant coefficients (though Merton's (also Spring 1973) justification paper generalized it to variable coefficients and many other things), lack of fat tails subsequent poor risk assessment, skewness, jumps, stochastic volatilities, etc.

∘ ***7.2.2. Market Calibration and Implied Volatility:***

One work-around the deficiencies with Black-Scholes formula, is to **find a volatility that better fits market values of the instrument of interest**, say the European call option. Hence, given market data $C^{(\mathrm{mkt})}(K_i, T_i)$ for a discrete number of strikes $K_i$and corresponding maturities T_i f or any given call option, the financial engineer will make an estimate of the volatility, and possibly other parameters, that is implied by option market rather than the underlying stock market.

When the underlying stock price data is used to to estimate the underlying volatility, then the **log-return** $\mathrm{LR}_i \equiv \log(S_{i+1}/S_i)$ is used, with estimated mean

$$\overline{\mathrm{LR}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{LR}_i. \tag{7.57}$$

The **unbiased estimated volatility**

$$\widehat{\sigma}^{(\text{hist})} = \sqrt{\frac{1}{(n-1)\Delta t}\sum_{i=1}^{n}(\text{LR}_i - \overline{\text{LR}})^2} \qquad (7.58)$$

is called the **historical volatility**; note that in the difference approximation to the asset SDE (SΔE), $\text{E}[\text{LR}_i] = (\widehat{\mu} - \widehat{\sigma}^2/2)\Delta t$ rather than the risk-neutral $(r - \widehat{\sigma}^2/2)\Delta t$.

However, the call market prices are not usually given directly, but, for instance in the delayed quotes at the **Chicago Board of Options Exchange (CBOE)**[a], they are given in terms of the latest bid and ask quotes, so usually one takes the **average of the bid and the ask quotes** for $C^{(\text{mkt})}(K_i, T_i)$ for each contract pair (K_i, T_i).

[a]CBOE *Delayed Market Quotes* page is found for downloadat the URL: *http://www.cboe.com/delayedquote/QuoteTableDownload.aspx*.

The option market implied estimate is the so-called **Black-Scholes implied volatility (IV)**, $\sigma_i^{(\mathrm{iv})}$, by solving the **inverse problem**[a], matching the BS call price to the market call price,

$$C^{(\mathrm{bs})}\left(M_i, T_i; \sigma_i^{(\mathrm{iv})}\right) = C^{(\mathrm{mkt})}(K_i, T_i), \tag{7.59}$$

defining $\sigma_i^{(\mathrm{iv})}$ for each $i = 1:n$, given options data $\{K_i, T_i, S_0\}$, where $M_i = S_0/K_i$, $C^{(\mathrm{bs})}$is given in (7.51), and for fixed r and $t = 0$ as the current time.

One problem in estimating volatility or variance is that they can not be directly observed but must be deduced from other observations like stock or option prices. There are also many methods for estimating implied volatility including Newton's method, maximum likelihood, kernel smoothing, and Monte Carlo, but for a single scalar variable like σ the derivative-free root-finder `fzero` of **MATLAB** could be used.

[a]Note that $\mathrm{vega} = `\nu` = \partial C^{(\mathrm{bs})}/\partial\sigma > 0$, a volatility sensitivity measure. Hence, the inverse should exist for Black-Scholes. See D. Highham (2004), *An Introduction to Financial Option Valuation*, p. 101 and 132; also for a simple justification of put-call parity.

However, there is not that much strike-maturity data, so **pooled data is sometimes used**, e.g., short maturity, medium maturity and long maturity options, or long-run historical data. Getting historical data has been harder to get, e.g., for European options, in the public domain, unless available in a company or business school.

○ *7.2.3. Risk-Neutral Option Pricing and Implied Volatility:*

While a relatively simple solution to European call or put option pricing problem with delta ($\Delta^{(\mathrm{bs})} \equiv \partial C^{(\mathrm{bs})}/\partial S$) hedging, the multiple sources of randomness in jump-diffusions or stochastic-volatility jump-diffusions do not allow for delta hedging to eliminate the purely diffusive risks. However, a risk-neutral formulation of the discounted, expected, conditional payoff simulates the principal properties of delta hedging. In addition, the arbitrage-free condition must be used by setting the instantaneous mean rate to the risk-free rate,

$$\mathrm{E}[dS(t)|S(t)=s]/(s dt)=r, \tag{7.60}$$

e.g., $\mu = r$ for linear diffusions or $\mu + \lambda\overline{\nu} = r$ for linear compound-jump-diffusions.

Thus, for linear diffusions and more general cases, the **current risk-neutral (RN) European style call or put option prices** are given by

$$\begin{bmatrix} \mathcal{C} \\ \mathcal{P} \end{bmatrix}^{(\mathrm{rn})}\left(s, t; K, T, r, \vec{\theta}\right) = e^{-r(T-t)} \cdot \mathrm{E}^{(\mathrm{rn})}[G(\pm S(T), \pm K) | S(t) = s], \tag{7.61}$$

where again $G(\pm S, \pm K) = \max[\pm(S-K), 0] \equiv [\pm(S-K)]_+$ is the payoff function and $\vec{\theta}$ is the vector of other model parameters.

◦ *7.2.4. Risk-Neutral Option Pricing Application to Compound-Jump-Diffusion (CJD) Underlying Asset:*

As a good example of a genuine risk-neutral options model, consider the risk-neutral version of the constant-coefficient, zero-one jumps on $(t, t + \Delta t]$ **compound-Poisson, jump-diffusion (CJD) SDE** asset price model (Merton, 1976) underlying the option,

$$dS^{(\mathrm{rn})}(t) = S(t)((r - \lambda\overline{\nu})dt + \sigma dW(t)) + \nu(Q)dP(t;Q)), \quad (7.62)$$

where the IID $\nu(Q) = \exp(Q) - 1$, $\overline{\nu} = \mathrm{E}_Q[\nu(Q)]$ and the required risk-neutral property is $\mathrm{E}[dS(t)|S(t)] = rS(t)dt$. Converting to the log-return variable $Y(t) = \log(S(t))$ using the hybrid independent continuous and jump process stochastic chain, leads to a state independent right-hand-side,

$$dY^{(\mathrm{rn})}(t) = (r - \sigma^2/2 - \lambda\overline{\nu})dt + \sigma dW(t) + QdP(t;Q), \quad (7.63)$$

where $Q = \log(1 + \nu(Q))$ has been used, provided $\nu(Q) > -1$.

Integrating from current time t to final, contract exercise time T, with $\tau = T - t$, and exponentiating yields,

$$S^{(\mathrm{rn})}(T) = S(t)\exp\left((r - \sigma^2/2 - \lambda\overline{\nu})\tau + \sigma W(\tau) + \sum_{j=1}^{P(\tau)} Q_j\right). \quad (7.64)$$

Since the time-to-maturity time-interval $(t, T]$ is not infinitesimal nor is sort of small a the trading day in years, we have to count the number of jumps in $(t, T]$ and the probable number of jumps in $(t, T]$ is the same as the number of jumps in $(0, T-t] = (0, \tau]$, so $P(T) - P(t) = P(\tau)$ and

$$\int_t^T Q dP(t) = \sum_{j=1}^{P(\tau)} Q_j \mathcal{I}_{\{P(\tau)>0\}} = \sum_{j=1}^{P(\tau)} Q_j, \quad (7.65)$$

with the no-jump convention that $\sum_{j=1}^{0} Q_j \equiv 0$ and $\mathcal{I}_{\{\mathcal{S}\}}$ is the indicator function for set $\mathcal{S}$.

The property of both diffusion and time-homogeneous Poisson processes that their increments depend only on the time step is called the **stationary property**, then $W(T)-W(t)=W(\tau)$ and $P(T)-P(t)=P(\tau)$. Next the scaled diffusion form, $W(\tau)=\sqrt{\tau}Z$ where Z is a mean-zero, variance-one normal RV. For notational simplicity, the $P(\tau)=k$ jump-sum is $\mathcal{S}_k \equiv \sum_{j=1}^{k} Q_j$ of jump-amplitudes.

Substituting formula (7.64) for $S(T)$ into the risk-neutral formula for the call option price with parameter vector $\vec{p}=[K,T,r,\vec{\theta}]$, along with normal density, Poisson distribution and IID RV expectation, yields,

$$C^{(\mathrm{rn})}(s,t;\vec{p})=e^{-r\tau}\sum_{k=0}^{\infty}p_k(\lambda\tau)\mathrm{E}_{\mathcal{S}_k}\Bigg[\int_{-\infty}^{\infty}dz\, f_Z^{(\mathrm{n})}(z;0,1) \cdot\max\Big[s\,e^{(r-\sigma^2/2-\lambda\bar{\nu})\tau+\sigma\sqrt{\tau}z+\mathcal{S}_k}-K,0\Big]\Bigg]. \tag{7.66}$$

Next, the payoff maximum operator will be eliminated by finding the **break even point (BEP)** for the payoff by finding the zero Z_0 of the first argument

$$s\,e^{(r-\sigma^2/2-\lambda\overline{\nu})\tau+\sigma\sqrt{\tau}Z_0+\mathcal{S}_k}-K=0, \tag{7.67}$$

whose solution is

$$\begin{aligned} Z_0 &\stackrel{\text{alg}}{=} -\big(\log(s/K)+(r-\sigma^2/2-\lambda\overline{\nu})\tau+\mathcal{S}_k\big)/\sqrt{\sigma^2\tau} \\ &\quad -d_2+(\lambda\overline{\nu}\tau+\mathcal{S}_k)/\sqrt{\sigma^2\tau} \qquad (7.68) \\ &\equiv -d_{2,k}\equiv -d_{1,k}+\sigma\sqrt{\tau}, \end{aligned}$$

borrowing from **Black and Scholes normal argument notation**, since we are following the risk-neutral procedure for the Black-Scholes formula modified for jumps.

Substitution back into the current version of risk-neutral call option price, cutting off the left tail of the integral,

$$C^{(\mathrm{rn})}(s,t;\vec{p}) = e^{-r\tau}\sum_{k=0}^{\infty} p_k(\lambda\tau)\mathrm{E}_{\mathcal{S}_k}\left[\int_{-d_{2,k}}^{\infty} dz f_Z^{(\mathrm{n})}(z;0,1)\left(s\,e^{(r-\sigma^2/2-\lambda\overline{\nu})\tau+\sigma\sqrt{\tau}z+\mathcal{S}_k}-K\right)\right]$$

$$\stackrel{\mathrm{alg}}{=} \sum_{k=0}^{\infty} p_k(\lambda\tau)\mathrm{E}_{\mathcal{S}_k}\Bigg[s\,e^{-(\sigma^2/2+\lambda\overline{\nu})\tau+\mathcal{S}_k}\int_{-d_{2,k}}^{\infty} dz f_Z^{(\mathrm{n})}(z;0,1)e^{\sigma\sqrt{\tau}z} -e^{-r\tau}K\int_{-d_{2,k}}^{\infty} dz f_Z^{(\mathrm{n})}(z;0,1)\Bigg]. \quad (7.69)$$

Either using the complete the square technique or letting $y = z - \sigma\sqrt{\tau}$, along with $\exp(\sigma Z)$ normal integral formula, then

$$C^{(\mathrm{rn})}(s,t;\vec{p}) = \sum_{k=0}^{\infty} p_k(\lambda\tau)\mathrm{E}_{\mathcal{S}_k}\Big[s\, e^{\mathcal{S}_k - \lambda\bar{\nu}\tau} F_Z^{(\mathrm{n})}(d_{1,k};0,1) - e^{-r\tau} K F_Z^{(\mathrm{n})}(d_{2,k};0,1)\Big], \quad (7.70)$$

where we have used the identities, $-d_{2,k} \equiv -d_{1,k} + \sigma\sqrt{\tau}$ and $\int_{-d}^{\infty} = \int_{-\infty}^{d}$ for even integrable integrands. Finally, we form the compound-jump-diffusion risk-neutral European call option price as a compound-Poisson mixture of Black-Scholes call option prices,

$$C^{(\mathrm{rn})}(s,t;\vec{p}) = \sum_{k=0}^{\infty} p_k(\lambda\tau) \cdot \mathrm{E}_{\mathcal{S}_k}\Big[C^{(\mathrm{bs})}\big(s e^{\mathcal{S}_k - \lambda\bar{\nu}\tau}, t; K, t+\tau, r, \sigma\big)\Big], \quad (7.71)$$

where, for example, in the case of a single uniform jump amplitude model, $\vec{\theta} = [\sigma, \lambda, a, b]^{\top}$. The formula reduces to the Black-Scholes if $\lambda = 0$ and $\vec{\theta} = [\sigma]$.

Due to the complex nature of this call option price formula with Poisson and IID RV expectations, perhaps **Monte Carlo simulations**[a] would be most practical, especially since a Poisson simulation of the number of jumps would keep the Poisson sum finite and the Poisson sum and the IID RV expectation could be combined. For instance, following Zhu and Hanson (2005), we can replace the Poisson sum and the sum $\mathcal{S}_k$ in the single uniform distribution case, with sample IID Poisson variates P_i for $i = 1:n$ and standard RVs $U_{i,j}$ for $j = 1:P_i$, on **(a,b)** by the estimate

$$\widehat{\mathcal{S}}_i = \sum_{j=1}^{P_i} Q_{i,j} = \sum_{j=1}^{P_i} (a + (b-a)U_{i,j}) = aP_i + (b-a)\sum_{j=1}^{P_i} U_{i,j}. \quad (7.72)$$

---

[a]Zhu and Hanson (2005) give elaborate Monte Carlo procedures with variance reduction techniques for European options in a jump-diffusion model with uniformly distributed jump-amplitudes, showing that jump-diffusion options are worth more than Black-Scholes diffusion options, in the paper at *http://www.math.uic.edu/ hanson/pub/CDC2005/cdc05zhweb.pdf* .

Then for the Poisson sample size n, the Monte Carlo estimate of the call option price, starting at $t=0$, is simply,

$$\widehat{C}_n = \frac{1}{n}\sum_{i=1}^{n} C^{(\mathrm{bs})}\left(S_0 e^{\widehat{\mathcal{S}}_i - \lambda\overline{\nu}T}, 0; K, T, r, \sigma\right) \equiv \frac{1}{n}\sum_{i=1}^{n} C_i^{(\mathrm{bs})}, \quad (7.73)$$

where $C_i^{(\mathrm{bs})}$ is IID compound Poisson variate along with $\widehat{\mathcal{S}}_i$, so

$$\widehat{C}_n \to C^{(\mathrm{rn})}(S_0, 0; K, T, r, \sigma) \;\; as \;\; n \to \infty \quad (7.74)$$

with probability one, with standard deviation,

$$\sigma_{\widehat{C}_n} = \frac{1}{\sqrt{n}}\sqrt{\mathrm{Var}[C_i^{(\mathrm{bs})}]} \simeq \frac{1}{\sqrt{n}}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(C_i^{(\mathrm{bs})} - \widehat{C}_n)^2}, \quad (7.75)$$

where in the last term the unbiased sample variance estimate was used.

There is more to the Monte Carlo application than these basic estimates, i.e., there are variance and bias reduction techniques to improve performance and accuracy.

○ ***7.2.5. Nonparametric, Multivariate Kernel Regression:***[a]

Kernel smoothers are useful for smoothly fitting data to a curve or surface, particularly when the user wants to do some continuous operations on the curve, like plotting and finding optima. Whereas, splines fit smooth curves by numerical interpolation by matching values and derivatives at data points using a low degree polynomial (cubics are often used, fitting up to second derivatives or more) interpolation. The kernel smoothers are related to the kernel density estimators, except that kernel smoothing regression gives an estimation of an expectation the response scalar variable, the $\boldsymbol{y}$, relative to the explanatory $\boldsymbol{m}$-vector, the $\vec{\boldsymbol{x}}$.

[a] This and other sections comes from Carmona ('04) Chapter 4, but the kernel smoothing part is not recommended for students.

For **independent or explanatory vector**, the distribution is represented by a **normalized kernel**, $K_{\vec{X}}$, with common scaled bandwidth b_x, the estimated smooth function for a sample of n independent observations, $\{\vec{X}_i, Y_i | i = 1 : n\}$, has the form

$$y \overset{\text{dist}}{\simeq} \phi(\vec{x}; b_x) = \sum_{i=1}^{n} Y_i K_{\vec{X}}\left(\frac{\vec{x} - \vec{X}_i}{b_x}\right) \Bigg/ \sum_{j=1}^{n} K_{\vec{X}}\left(\frac{\vec{x} - \vec{X}_j}{b_x}\right), \quad (7.76)$$

where the kernel $K_{\vec{X}}(\vec{\xi})$ is some model proper (i.e., integrates to one on the domain) density like normal, uniform or triangular and is used with a **standardized argument** $\vec{\xi}$ centered about some data point $\vec{x}_i$ and normalized with the scale of the bandwidth b_x for better computational properties. Standardized variables reduces the effects of floating point truncation errors. The normal kernel is often used because of supporting theory. Also, due to centered arguments, usually the kernel is assumed to symmetric, i.e., $K_{\vec{X}}(-\vec{\xi}) = K_{\vec{X}}(\vec{\xi})$.

Actually, the **smoothed function** $\phi(\vec{x}; b_x)$ is basically a simulation of the conditional expectation of a dependent or response variable y conditioned on the independent or explanatory variable $\vec{x}$, since

$$\begin{aligned} \mathrm{E}[Y|\vec{X}=\vec{x}] &= \int_{-\infty}^{+\infty} y f_{Y|\vec{X}}(y|\vec{X}=\vec{x})dy \\ &= \int_{-\infty}^{+\infty} y f_{\vec{X},Y}(\vec{x},y)dy/f_{\vec{X}}(\vec{x}), \end{aligned} \tag{7.77}$$

by a **Bayes' rule for densities**,

$$f_{Y|\vec{X}}(y|\vec{X}=\vec{x}) = \frac{f_{\vec{X},Y}(\vec{x},y)}{f_{\vec{X}}(\vec{x})}, \tag{7.78}$$

following from the definition of conditional probability.[a]

[a]Hanson (2007), Preliminaries Online Appendix B, p. B26.

For motivation, consider the **univariate kernel estimation** of $f_X(x)$,

$$f_X(x) \simeq \widehat{f}_X(x) = \frac{1}{nb_x} \sum_{i=1}^{n} K_X\left(\frac{x - X_i}{b_x}\right). \tag{7.79}$$

Assuming that the **joint kernel is separable**, i.e.,

$$K_{X,Y}(\xi, \eta) = K_X(\xi) \cdot K_Y(\eta), \tag{7.80}$$

and that the joint density has the estimate,

$$f_{X,Y}(x, y) \simeq \widehat{f}_{X,Y}(x, y) = \frac{1}{nb_x b_y} \sum_{i=1}^{n} K_{X,Y}\left(\frac{x - X_i}{b_x}, \frac{y - Y_i}{b_y}\right), \tag{7.81}$$

then

$$\widehat{f}_{X,Y}(x, y) = \frac{1}{nb_x b_y} \sum_{i=1}^{n} K_X\left(\frac{x - X_i}{b_x}\right) K_Y\left(\frac{y - Y_i}{b_y}\right). \tag{7.82}$$

Also, by the conditional expectations,

$$\begin{aligned} f_X(x)\mathrm{E}[Y|X=x] &= \int_{-\infty}^{+\infty} y f_{X,Y}(x,y)dy \\ &\simeq \frac{1}{nb_x b_y}\sum_{i=1}^{n} K_X\!\left(\frac{x-X_i}{b_x}\right) \\ &\quad \cdot \int_{-\infty}^{+\infty} y K_Y\!\left(\frac{y-Y_i}{b_y}\right) dy \\ &= \frac{1}{nb_x}\sum_{i=1}^{n} Y_i K_X\!\left(\frac{x-X_i}{b_x}\right), \end{aligned} \tag{7.83}$$

since, in the y-integral, letting $\eta=(y-Y_i)/b_y$,

$$\int_{-\infty}^{+\infty} y K_Y\left(\frac{y-Y_i}{b_y}\right) dy = b_y\left(Y_i + \int_{-\infty}^{+\infty} \eta K_Y(\eta)d y\right) = b_y Y_i, \tag{7.84}$$

by the fact that K_Y is also a symmetric and proper density like K_X.

Finally by reassembling our formulas, we have the desired motivational result,

$$\mathrm{E}[Y|X=x] \simeq \sum_{i=1}^{n} Y_i K_X\left(\frac{x-X_i}{b_x}\right) \Bigg/ \sum_{j=1}^{n} K_X\left(\frac{x-X_j}{b_x}\right), \qquad (7.85)$$

the denominator sum coming from using another approximation $f_X(x) \simeq \widehat{f}_X(x)$ consistent with the joint density estimated approximation.

The kernel smoothing regression formula (7.85) has been implemented using a univariate Gaussian kernel by Yi Cao as **`ksr.m`** and posted on the **`MATLAB`** Central File Exchange at

Univariate Kernel Regression MATLAB code.

Cao refers to kernel estmators such as (7.85) as **Nadaraya-Watson kernel regressions**.

The **multivariate kernel smoothing case** is more complicated, or rather tedious, due to dimensional complexity, **even if the kernel is separable in vector variable** $\vec{x}=[x_j]_{1\times m}$ with corresponding observations $X=[X_{i,j}]_{n\times m}=[\vec{X}_i]_{n\times 1}$, i.e., in the separable case,

$$K_{\vec{X}}\left((\vec{x}-\vec{X}_i)./\vec{b}_x\right)=\prod_{k=1}^{m}K_{X_k}\left(\frac{x_k-X_{i,k}}{b_{x_k}}\right). \tag{7.86}$$

for $i=1:n$ vector observations, where the vector bandwidth scaling is $\vec{b}_x=[b_{x_j}]_{1\times m}$, and $./$ denotes the element by element division of MATLAB. Substituting (7.86) into (7.85) in place of the univariate kernel gives the needed objective.

$$\mathrm{E}[Y|\vec{X}=\vec{x}]\simeq\sum_{i=1}^{n}Y_iK_{\vec{X}}\left((\vec{x}-\vec{X}_i)./\vec{b}_x\right)\Bigg/\sum_{j=1}^{n}K_{\vec{X}}\left((\vec{x}-\vec{X}_j)./\vec{b}_x\right), \tag{7.87}$$

Cao also has implemented a multivariate kernel regression code, again the kernel is Gaussian kernel, and is called `ksrmv.m` at

Multivariate Kernel Regression MATLAB code.

∘ *7.2.6. Kernel Regression for Implied Volatility (IV) Computations:*

In the **Black-Scholes implied volatility (BSIV) inverse problem**, here starting at $t = 0$, given options data $\{K_i, T_i, S_0\}$, the match is

$$C_0^{(\mathrm{bs})}\left(M_i, T_i; \sigma_i^{(\mathrm{iv})}\right) = C_0^{(\mathrm{mkt})}(M_i, T_i), \tag{7.88}$$

defining $\sigma_i^{(\mathrm{iv})}$ for each $i = 1:n$, where the moneyness variable for $t = 0$ $M_i \equiv S_0/K_i$[a] for $k = 1:n$ helps to reduce the problem dimensionality. Similarly, the $C_0^{(\mathrm{bs})}$ and $C_0^{(\mathrm{mkt})}$ are defined with $t = 0$ to suppress the current time, so $\tau_i = T_i$ from (7.59). We also assume here the risk-free rate r_i is taken from the current **U.S. Federal Reserve Target Rate**, so is fixed in this computation.

[a]Recall, sometimes the reciprocal K/S_0 is used for moneyness, a more suitable form for put options.

One can solve try to solve the inverse problem (7.88) directly by a root-finding method like **fzero.m** or several of the variations of Newton's method, obtaining the estimate $\widehat{\sigma}_i^{(\mathrm{bsiv})}(M_i, T_i)$.

Then the kernal smoothing using the $\widehat{\sigma}_i^{(\mathrm{bsiv})}(M_i, T_i)$ data gives us the smoothed BSIV estimate

$$\widehat{\sigma}_{ks}^{(\mathrm{bsiv})}(M,T) = \sum_{i=1}^{n} \widehat{\sigma}_i^{(\mathrm{bsiv})}(M_i, T_i) K_M\left(\frac{M - M_i}{b_M}\right) K_T\left(\frac{T - T_i}{b_T}\right) \Bigg/ \sum_{j=1}^{n} K_M\left(\frac{M - M_j}{b_M}\right) K_T\left(\frac{T - T_j}{b_T}\right), \tag{7.89}$$

using the usual kernel smoothing procedure (7.87), designated by the ks in $\widehat{\sigma}_{ks}^{(\mathrm{bsiv})}(M,T)$. This estimate should be suitable for plotting an estimated BSIV volatility surface with a 2D-grid for (M,T), and this can also be used for the CJD model replacing the BS model. Gaussian kernels are acceptable for both M and T, so the kernel can be of the form $K_X(x) = \mathtt{normpdf}(x, 0; 1)$ in **MATLAB** notation.

A second approach tries to handle data error robustly by combining **Nadaraya-Watson smoothing with a least squares, approach**. A recent least squares kernel smoothing estimator is by Fengler et al. (2003)[a] seemed to be more appropriate for (7.88) IV inverse problem and efficient using a robust weighting $W(M_i)$ depending on the moneyness and regression on a weighted least squares with respect to the strike price K_i and time to maturity T_i. That is, the BSIV least squares estimate.

$$\widehat{\sigma}_{ls}^{(\mathrm{bsiv})}(M,T) = \underset{\sigma}{\mathrm{argmin}}\left[\sum_{i=1}^{n}\left(C_i^{(\mathrm{mkt})} - C_i^{(\mathrm{bs})}(\sigma)\right)^2 W(M_i) \cdot K_M\left(\frac{M-M_i}{b_M}\right) K_T\left(\frac{T-T_i}{b_T}\right)\right], \tag{7.90}$$

where $C_i^{(\mathrm{mkt})} = C^{(\mathrm{mkt})}(K_i, T_i)$, $C_i^{(\mathrm{bs})}(\sigma) = C_0^{(\mathrm{bs})}(M_i, T_i; \sigma)$, $M_i = S_0/K_i$ and $M = S_0/K$. This can also be used for a volatility surface with 2D-grid for either BS model or CJD model.

[a]M.R. Fengler and Q. Wang, Fitting the Smile Revisited: A Least Squares Kernel Estimator for the Implied Volatility Surface, *Least Squares Kernel Estimator paper*.

The semi-bandwidth terms, b_M and b_T, should be reasonable estimates of the variability of M and T, respectively. A convenient selection of the bandwidth scaling is the bandwidth of the data, i.e.,

$$b_X = \texttt{std}(\vec{X}, 0; 1) \tag{7.91}$$

where $\vec{X} = [X_i]_{n\times 1}$, such that X_i is either $\{M_i \text{ or } \widetilde{M}_i = 1/M_i\}$, or T_i. An alternate optimal formulation is that of Bowman and Azzalini (1997)[a] using median values, so that

$$b_X = \texttt{median}(\texttt{abs}(\vec{X} - \texttt{median}(\vec{X})))/0.6745 * (4/3/n)^{0.2}. \tag{7.92}$$

---

[a] A.W. Bowman, A. Azzalini (1997), Applied Smoothing Techniques for Data Analysis: the Kernel Approach With S-Plus Illustrations, Oxford University Press, Oxford, UK. See also Y. Cao (2008), `ksr.m`, `ksrlin.m` or any other of Cao's series on kernel smoothing regression in **MATLAB Central** for a code applications of Bowman & Azzalini's bandwidth scaling or Nadaraya-Watson's least squares smoothing.

A convenient weighting function, suggested by use of **robustfit.m** is the **'fair'** weighting function mentioned on (6.5; L6p3), i.e.,

$$W(M) = 1/(1 + \mathtt{abs}(M)). \tag{7.93}$$

The moneyness weights should give less weight to the less traded ITM options and the Fengler et al. (2003) suggest using

$$W(\widetilde{M}) = \mathrm{atan}(\pm\beta(1 - \widetilde{M}))/\pi + 0.5, \tag{7.94}$$

where they use the reciprocal moneyness $\widetilde{M} = 1./M$ here and in the kernel, with speed control $\beta \simeq 9$ and where the $(\pm)$ is the usual sign for calls $(+)$ and puts $(-)$.

One of my Singapore FINM 331 Winter 2009 students, Rudy Sitter, put his volatility surface code for Black-Scholes implied volatility on MATLAB Central as **VolSsurface.m** from one of his class projects. ***BSIV Volatility Surface Code LINK***[a].

[a] In fact, much of the updates to this lecture comes from the feedback of the Winter 2009 students in Singapore, in particular, Stephen Huang, and other campuses. There seem to be a healthy intersection between this course and Professor R. Lee's numerical methods class.

∘ *7.2.7. CBOE Market Quotes:*

The market European call (or put) option price can be obtained from the **CBOE Delayed Market Quotes**, Quote Table Download page[a] using an appropriate market symbol[b]. The first two items listed in the first column of the quote table will be the 2- digit **year** and the **exercise month** followed by the strike price.

---

[a] ***http://www.cboe.com/delayedquote/QuoteTableDownload.aspx*** (See description on how change comma-delimited format to Excel, if wanted.

[b] E.G., for S&P 500 Index option SPX (Caution: some companies also use that symbol and you have to avoid getting the index itself, rather than the option.) the product specification is at ***http://www.cboe.com/products/indexopts/spx_spec.aspx***.

The **expiration date**[a], in case of the **SPX option** is **the Saturday after** the third Friday of the month (e.g., if **09 Feb**[b] is the expiration month then **21** Feb. is the expiration date.). **Prices are listed in CBOE points and come with a current multiplier of $100, so for example 200.00 points is $20,000.00. SPX is a European style option** as is **XEO is an option on the S&P100 Index**, while **OEX options on that index are American**, early exercise, style. Index options are different from stock options in many ways. Mileage, i.e., specifications will vary for other options.

[a] See Hull (2006, 6th Edition) for very practical description the CBOE quote table of hard to find information, pp. 187 & 316-317 (recommended).

[b] A fragment of the top left corner of the SPX quote table looks like

SPX (S&P 500 INDEX) Feb 15 2009 @ 13:45 ET	826.84	-8.35		
Calls	LastSale	Net	Bid	Ask
09 Feb 200.00 (SPV BD-E)	635.10	0.0	619.50	622.40

so a market call estimate would $C^{(\mathrm{mkt})}(20000, 4/252)$, counting 4 trading days **due to the market Monday Holiday**. Note for the SPX, there are no weekly options (ticker: JX[A,B,D,E]) and weeklys are different from monthlys and long-term versions.

SPX (S&P 500 INDEX)	826.84	-8.35				
Feb 15 2009 @ 13:45 ET						
Calls	Last Sale	Net	Bid	Ask	Vol	Open Int
09 Feb 200.00 (SPV BD-E)	635.10	0.0	619.50	622.40	0	1
09 Feb 300.00 (SPV BT-E)	0.0	0.0	519.60	522.30	0	0
09 Feb 325.00 (SPV BE-E)	0.0	0.0	494.50	497.40	0	0
09 Feb 350.00 (SPV BJ-E)	0.0	0.0	469.60	472.40	0	0
09 Feb 375.00 (SPV BO-E)	0.0	0.0	444.60	447.40	0	0
09 Feb 400.00 (SZU BT-E)	421.05	0.0	419.60	422.40	0	70
09 Feb 425.00 (SZU BE-E)	403.45	0.0	394.60	397.40	0	98
09 Feb 450.00 (SZU BJ-E)	377.10	0.0	369.60	372.40	0	100
09 Feb 475.00 (SZU BO-E)	0.0	0.0	344.60	347.40	0	0
09 Feb 490.00 (SZU BR-E)	0.0	0.0	329.50	332.20	0	0
09 Feb 500.00 (SYU BT-E)	326.00	0.0	319.60	322.40	0	2875
09 Feb 525.00 (SYU BE-E)	330.30	0.0	294.60	297.40	0	98
09 Feb 550.00 (SYU BJ-E)	265.00	0.0	269.60	272.40	0	130
09 Feb 560.00 (SYU BL-E)	269.00	0.0	259.60	262.40	0	50
09 Feb 575.00 (SYU BP-E)	356.00	0.0	244.60	247.50	0	75
09 Feb 580.00 (SYU BY-E)	0.0	0.0	239.60	242.50	0	0
09 Feb 590.00 (SYU BR-E)	0.0	0.0	229.60	232.50	0	0
09 Feb 600.00 (SYG BT-E)	231.05	0.0	219.60	222.50	0	1959
09 Feb 610.00 (SYG BB-E)	221.15	0.0	209.60	212.50	0	2
09 Feb 620.00 (SYG BD-E)	0.0	0.0	199.60	202.50	0	0
09 Feb 625.00 (SYG BE-E)	202.90	0.0	194.60	197.60	0	53
09 Feb 630.00 (SYG BF-E)	0.0	0.0	189.70	192.60	0	0
09 Feb 635.00 (SYG BG-E)	0.0	0.0	184.70	187.60	0	0
09 Feb 640.00 (SYG BH-E)	0.0	0.0	179.70	182.60	0	0
09 Feb 645.00 (SYG BI-E)	0.0	0.0	174.70	177.60	0	0
09 Feb 650.00 (SYG BJ-E)	182.50	+9.30	169.70	172.70	5	146
09 Feb 655.00 (SYG BK-E)	0.0	0.0	164.70	167.70	0	0
09 Feb 660.00 (SYG BL-E)	167.00	+12.50	159.80	162.60	3	13
09 Feb 665.00 (SYG BM-E)	165.45	0.0	154.80	157.70	0	20
09 Feb 670.00 (SYG BN-E)	0.0	0.0	149.80	152.80	0	0
09 Feb 675.00 (SYG BO-E)	154.60	0.0	144.80	147.70	0	95
09 Feb 680.00 (SYG BP-E)	0.0	0.0	140.00	142.80	0	0
09 Feb 685.00 (SYG BQ-E)	0.0	0.0	135.00	137.80	0	0

Figure 7.1: **CBOE Quote Table** for S&P 500 Index Options with only call option columns (put columns supressed) from ***page 1*** of **Delayed Quote Download** page from February 15, 2009.

09 Feb 1050.00 (SPQ BJ-E)	0.0	-0.05	0.0	0.05	2	20430
09 Feb 1055.00 (SPQ BK-E)	0.10	0.0	0.0	0.10	0	1293
09 Feb 1060.00 (SPQ BL-E)	0.05	0.0	0.0	0.10	1	827
09 Feb 1065.00 (SPQ BM-E)	0.10	0.0	0.0	0.10	0	257
09 Feb 1070.00 (SPQ BN-E)	0.05	-0.05	0.0	0.10	1	2488
09 Feb 1075.00 (SPQ BO-E)	0.05	0.0	0.0	0.05	0	6145
09 Feb 1080.00 (SPQ BP-E)	0.10	0.0	0.0	0.05	0	1405
09 Feb 1085.00 (SPQ BQ-E)	0.05	0.0	0.0	0.10	0	513
09 Feb 1090.00 (SPQ BR-E)	0.05	0.0	0.0	0.05	0	347
09 Feb 1095.00 (SPQ BS-E)	0.05	0.0	0.0	0.05	0	185
09 Feb 1100.00 (SPT BT-E)	0.05	0.0	0.0	0.05	0	19166
09 Feb 1105.00 (SPT BA-E)	0.05	0.0	0.0	0.05	0	50
09 Feb 1110.00 (SPT BB-E)	0.05	0.0	0.0	0.05	0	337
09 Feb 1115.00 (SPT BC-E)	0.05	0.0	0.0	0.05	0	20
09 Feb 1120.00 (SPT BD-E)	0.05	0.0	0.0	0.05	0	2013
09 Feb 1125.00 (SPT BE-E)	0.05	0.0	0.0	0.05	0	8318
09 Feb 1130.00 (SPT BF-E)	0.05	0.0	0.0	0.05	0	3050
09 Feb 1140.00 (SPT BH-E)	0.05	0.0	0.0	0.05	0	1396
09 Feb 1150.00 (SPT BJ-E)	0.05	0.0	0.0	0.05	0	19733
09 Feb 1160.00 (SPT BL-E)	0.05	0.0	0.0	0.05	0	48
09 Feb 1175.00 (SPT BO-E)	0.05	0.0	0.0	0.05	0	1729
09 Feb 1180.00 (SPT BP-E)	0.25	0.0	0.0	0.05	0	43
09 Feb 1200.00 (SZP BT-E)	0.05	0.0	0.0	0.05	0	482
09 Feb 1225.00 (SZP BE-E)	0.05	0.0	0.0	0.05	0	939
09 Feb 1250.00 (SZP BJ-E)	0.50	0.0	0.0	0.05	0	21
09 Feb 1275.00 (SZP BO-E)	1.20	0.0	0.0	0.05	0	30
09 Feb 1300.00 (SXY BT-E)	0.20	0.0	0.0	0.05	0	18003
09 Feb 1325.00 (SXY BE-E)	0.65	0.0	0.0	0.05	0	504
09 Feb 1350.00 (SXY BJ-E)	0.0	0.0	0.0	0.05	0	0
09 Feb 1375.00 (SXY BO-E)	0.0	0.0	0.0	0.05	0	0
09 Feb 1400.00 (SXZ BT-E)	0.40	0.0	0.0	0.05	0	25
09 Feb 1450.00 (SXZ BJ-E)	0.0	0.0	0.0	0.05	0	0
09 Feb 1500.00 (SXM BT-E)	0.05	0.0	0.0	0.05	0	2802
09 Mar 200.00 (SPV CD-E)	707.60	0.0	618.00	620.80	0	360
09 Mar 300.00 (SPV CT-E)	0.0	0.0	518.10	520.90	0	0
09 Mar 325.00 (SPV CE-E)	0.0	0.0	493.10	495.90	0	0

Figure 7.2: **CBOE Quote Table** for S&P 500 Index Options with only call option columns from ***page 4*** of **Delayed Quote Download** page from February 15, 2009.

09 Mar 1450.00 (SLQ CJ-E)	1.60	0.0	0.0	1.00	0	253
09 Mar 1500.00 (SQP CT-E)	1.25	0.0	0.0	1.00	0	1
09 Mar 1550.00 (SQP CJ-E)	0.0	0.0	0.0	1.00	0	0
09 Apr 200.00 (SPV DD-E)	0.0	0.0	615.80	620.30	0	0
09 Apr 300.00 (SPV DT-E)	0.0	0.0	516.10	520.60	0	0
09 Apr 350.00 (SPV DJ-E)	0.0	0.0	466.30	470.80	0	0
09 Apr 375.00 (SPV DO-E)	0.0	0.0	441.50	446.00	0	0
09 Apr 400.00 (SZU DT-E)	0.0	0.0	416.70	421.20	0	0
09 Apr 425.00 (SZU DE-E)	0.0	0.0	392.10	396.60	0	0
09 Apr 450.00 (SZU DJ-E)	0.0	0.0	367.30	371.80	0	0
09 Apr 480.00 (SZU DP-E)	0.0	0.0	337.90	342.40	0	0
09 Apr 490.00 (SZU DR-E)	0.0	0.0	328.10	332.60	0	0
09 Apr 500.00 (SYU DT-E)	334.15	0.0	318.30	322.80	0	7
09 Apr 510.00 (SYU DB-E)	0.0	0.0	308.60	313.10	0	0
09 Apr 515.00 (SYU DU-E)	0.0	0.0	303.70	308.20	0	0
09 Apr 520.00 (SYU DD-E)	0.0	0.0	298.90	303.40	0	0
09 Apr 525.00 (SYU DE-E)	0.0	0.0	294.10	298.60	0	0
09 Apr 530.00 (SYU DF-E)	0.0	0.0	289.20	293.70	0	0
09 Apr 540.00 (SYU DH-E)	0.0	0.0	279.60	284.10	0	0
09 Apr 550.00 (SYU DJ-E)	0.0	0.0	270.00	274.50	0	0
09 Apr 560.00 (SYU DL-E)	0.0	0.0	260.40	264.90	0	0
09 Apr 570.00 (SYU DN-E)	0.0	0.0	250.90	255.40	0	0
09 Apr 575.00 (SYU DP-E)	0.0	0.0	246.20	250.70	0	0
09 Apr 580.00 (SYU DY-E)	0.0	0.0	241.50	245.90	0	0
09 Apr 585.00 (SYU DQ-E)	0.0	0.0	236.80	241.20	0	0
09 Apr 590.00 (SYU DR-E)	0.0	0.0	232.10	236.50	0	0
09 Apr 600.00 (SYG DT-E)	0.0	0.0	222.70	227.10	0	0
09 Apr 610.00 (SYG DB-E)	0.0	0.0	213.40	217.80	0	0
09 Apr 620.00 (SYG DD-E)	0.0	0.0	204.20	208.60	0	0
09 Apr 625.00 (SYG DE-E)	0.0	0.0	199.60	204.00	0	0
09 Apr 630.00 (SYG DF-E)	0.0	0.0	195.00	199.40	0	0
09 Apr 640.00 (SYG DH-E)	0.0	0.0	186.00	190.40	0	0
09 Apr 650.00 (SYG DJ-E)	264.50	0.0	177.00	181.40	0	50
09 Apr 660.00 (SYG DL-E)	0.0	0.0	168.10	172.50	0	0
09 Apr 670.00 (SYG DN-E)	0.0	0.0	159.30	163.70	0	0
09 Apr 675.00 (SYG DO-E)	242.00	0.0	155.00	159.40	0	50

Figure 7.3: **CBOE Quote Table** for S&P 500 Index Options with only call option columns from ***page 10*** of **Delayed Quote Download** page from February 15, 2009.

10 Dec 2500.00 (SYZ LU-E)	0.05	0.0	0.05	0.95	0	9617
11 Dec 250.00 (SZJ LE-E)	543.70	0.0	544.00	549.80	0	13
11 Dec 280.00 (SZJ LP-E)	519.00	0.0	519.20	524.80	0	102
11 Dec 300.00 (SZJ LF-E)	497.80	0.0	503.50	509.00	0	82
11 Dec 350.00 (SZJ LG-E)	0.0	0.0	463.80	469.70	0	0
11 Dec 400.00 (SZJ LB-E)	423.40	0.0	426.20	432.00	0	32
11 Dec 450.00 (SZJ LI-E)	388.80	0.0	390.60	395.90	0	5
11 Dec 500.00 (SZJ LC-E)	360.00	0.0	355.90	361.50	0	3
11 Dec 550.00 (SZJ LK-E)	331.60	0.0	323.20	328.70	0	5
11 Dec 600.00 (SZJ LR-E)	352.50	0.0	291.90	297.90	0	5
11 Dec 650.00 (SZJ LM-E)	324.50	0.0	263.10	268.80	0	5
11 Dec 700.00 (SZJ LA-E)	245.00	0.0	234.90	240.90	0	50
11 Dec 800.00 (SZJ LL-E)	197.50	0.0	185.00	191.00	0	865
11 Dec 850.00 (SZJ LJ-E)	172.55	-2.45	162.20	168.00	3	4210
11 Dec 900.00 (SZJ LT-E)	148.00	+8.50	141.40	147.20	10	2647
11 Dec 950.00 (SZJ LS-E)	128.00	0.0	122.30	128.20	0	456
11 Dec 1000.00 (SZT LR-E)	131.35	0.0	105.00	111.00	0	353
11 Dec 1100.00 (SZT LT-E)	81.00	-7.00	76.00	81.90	300	1088
11 Dec 1200.00 (SZT LU-E)	56.60	0.0	53.40	59.10	0	1970
11 Dec 1300.00 (SZT LW-E)	40.00	0.0	36.20	41.90	0	811
11 Dec 1400.00 (SZT LA-E)	32.00	0.0	23.60	29.30	0	95
11 Dec 1500.00 (SZV LT-E)	20.70	0.0	15.40	19.60	0	1151
11 Dec 1600.00 (SZV LO-E)	15.00	0.0	9.90	12.90	0	5750
11 Dec 1700.00 (SZV LA-E)	17.70	0.0	5.70	8.70	0	25
11 Dec 1800.00 (SZV LD-E)	0.0	0.0	3.00	5.80	0	0
11 Dec 1900.00 (SZV LI-E)	0.0	0.0	1.40	4.00	0	0
11 Dec 2000.00 (SZV LE-E)	2.50	0.0	0.55	2.60	0	4260

Figure 7.4: **CBOE Quote Table** for S&P 500 Index Options with only call option columns from ***page 23*** of **Delayed Quote Download** page from February 15, 2009 (***Long term LEAP options, up to 3 years***).

○ ***7.2.8. Implied Volatility Algorithm with Kernel Regression and Numerical Inversion:***

Returning to the implied volatility computations, here is our **pseudo-algorithm**:

1. **Select an option** to study and download the quote table from the CBOE, or other exchange that allow public domain downloads, from the delayed quote page.
2. **Select a few exercise times T_i**, some in weeks and others in months (do not forget to convert to years, since the FRB risk-free rates are in years and that dominates the units. Also, **short exercise times are more likely to produce implied volatility smiles** (like a minimum curve), while **long times produce smirks** (like a maximum curve). If you have a **volatility surface** in mind and you should, then you will need more than a few exercise times.

3. **Next select a number of strike values K_i**, enough data to produce a respectable implied volatility curve of the σ_i (do not forget to account for quote table scaling) versus moneyness $M_i = S_0/K_i$, where S_0 is the current price of the underlying asset which should be on the top of your quote table (CBOE rules).

4. Compute the model call (or put) option price data using a grid of volatility values, collected in a single index for simplicity, σ_i for $i = 1:n$ that produce a realistic range of option prices $C_i \equiv C(K_i, T_i; \sigma_i)$ with your set of contract parameters (K_i, T_i) or equivalently (M_i, T_i) given S_0 using the **Black-Scholes model** (7.53) or (7.51). Else, using the Black-Scholes model as a test case, the **compound-jump-diffusion (CJD) model**, replacing the match difference $(C_i^{(\mathrm{mkt})} - C_i^{(\mathrm{bs})}(\sigma))$ by $(C_i^{(\mathrm{mkt})} - C_i^{(\mathrm{cjd})}(\sigma))$, where $C_i^{(\mathrm{cjd})}(\sigma) = \widehat{C}_n$ in (7.73), using a **full Monte Carlo simulation**[a], or the simpler **partial Monte Carlo** with simulated jump-part only.

---

[a] Some estimates of jump-parameters, such as an maximum likelihood estimates (MLE) on the CJD zero-one jump daily log-return model would be needed.

5. Now form the function forming the estimated market option price curve for each value of the moneyness M_i with fixed T_i and other parameters using the kernel smoothing technique, using either **regular kernel smoothing regression estimation** as with $\widehat{\sigma}_{ks}^{(*\text{iv})}(M,T)$ (7.89) with $* = \textbf{bs or cjd}$, or the more robust **least squares kernel smoothing regression estimation** with $\widehat{\sigma}_{ls}^{(*\text{iv})}(M,T)$ (7.90). Gaussian kernel and one of several bandwidth formulas can be used, where $X_i = M_i$ **or** T_i and $Y_i = \widehat{\sigma}_i^{(*\text{iv})}$ **or** $(C_i^{(\text{mkt})} - C_i^{(\text{bs})}(\sigma))^2$ **or** $(C_i^{(\text{mkt})} - C_i^{(\text{cjd})}(\sigma))^2$, with other parameters suppressed. However, in particular, the user needs to keep track of the contract set (M_i, T_i) each i-kernel smoothing operation.[a]

[a] See the **MATLAB** public domain **kernel smoothing regression (KSR)** code **ksr**, potentially a vector-argument kernel so could use $\vec{x}_i = [M_i, T_i]'$ with output response Y_i, described later. Also, other regression methods such as maximum likelihood could be used or other smoothing methods such as spline interpolation.

6. Then, for the implied volatility step is basically a nonlinear zero-finding problem: find an σ^* such that $g(\sigma^*, \cdot) = C_i^{(\mathrm{mkt})}$, for each fixed i, where $C_i^{(\mathrm{mkt})} \equiv C^{(\mathrm{mkt})}(K_i, T_i)$ or $C^{(\mathrm{mkt})}(M_i, T_i)$, which in principle could determine a volatility surface. There are many basic methods that could be used here, such as the classic univariate zero finder **`fzero`** for scalar function of a scalar variable that is in MATLAB or Newton's methods or any of its quasi-variants.

7. When the roots for each i data pair for $i = 1 : n$ are assembled, then implied volatility curves versus moneyness and parameterized by exercise time can be plotted. **Also**, the implied or local volatility surface **should** be plotted against both moneyness and exercise or maturity time is three-dimensional graphs with **`surf or mesh`** using a 2D-grid in (M, T).

◦ *7.2.9. Kernel Smoothing Regression (KSR):*[a]

1. Syntax: `function r=ksr(x,y,b,n)`, computes the Gaussian kernel regression of `y` versus `x` and outputs the structure `r`. Part of a KSR-series with `ksrlin`, `ksrmv ...`.
2. Input: The `x` is the explanatory data `n`-vector, `y` is the response data `n`-vector, `b or h` is a specified bandwidth of the kernel (if the user wants `ksr` to compute an optimal bandwidth then use `r=ksr(x,y)` form, and `n` is specified data length but should not be needed.
3. Output: The `r` is a structure, such that `r.h` is the computed bandwidth `b or h` and `r.n` is the **number of samples** and `r.f(r.x)=y(x)+e` is the form of the regression computed, all when the short form is used. The regression is plotted for the forms `r=ksr(x,y)` and `r=ksr(x,y,b)`.

See the MATLAB Central Exchange for more documentation and code.

---

[a] Kernel Smoothing Regression by Yi Cao, 2008, *http://www.mathworks.com/matlabcentral/fileexchange/19195*.

○ *7.2.10. Univariate Root or Zero Finder (`fzero`):*

1. Syntax:

   `[x,fval,exitflag,output]=fzero(@f,x0,options);`

   solves the zero or root problem for a scalar valued function **`f`** of a single scalar argument **`x`**, for an x^* such that $f(x^*)=0$ given a start $x0$ and objective function **`f`** appearing as the first argument as the pointer or handle **`@f`** usually pointing to a subfunction within the main function m-file.

2. Additional parameter can be passed to the (sub-)function **`f`** using a global statement in called and calling functions, as with **`fminsearch`** of Lecture 5, in fact, the syntax is much like that of **`fminsearch`**, except of the multivariate properties.

3. The output arguments have essentially the same descriptions as those in **`fminsearch`** and all but **`x`** are optional.

See D. Higham (2004) Chapters 14 and 20 for other methods for implied volatility, including Monte Carlo. ***For instance,*** $h(x)=g(x)-C_i^{(\mathrm{mkt})}$ ***and*** $x0=\sigma_i$.

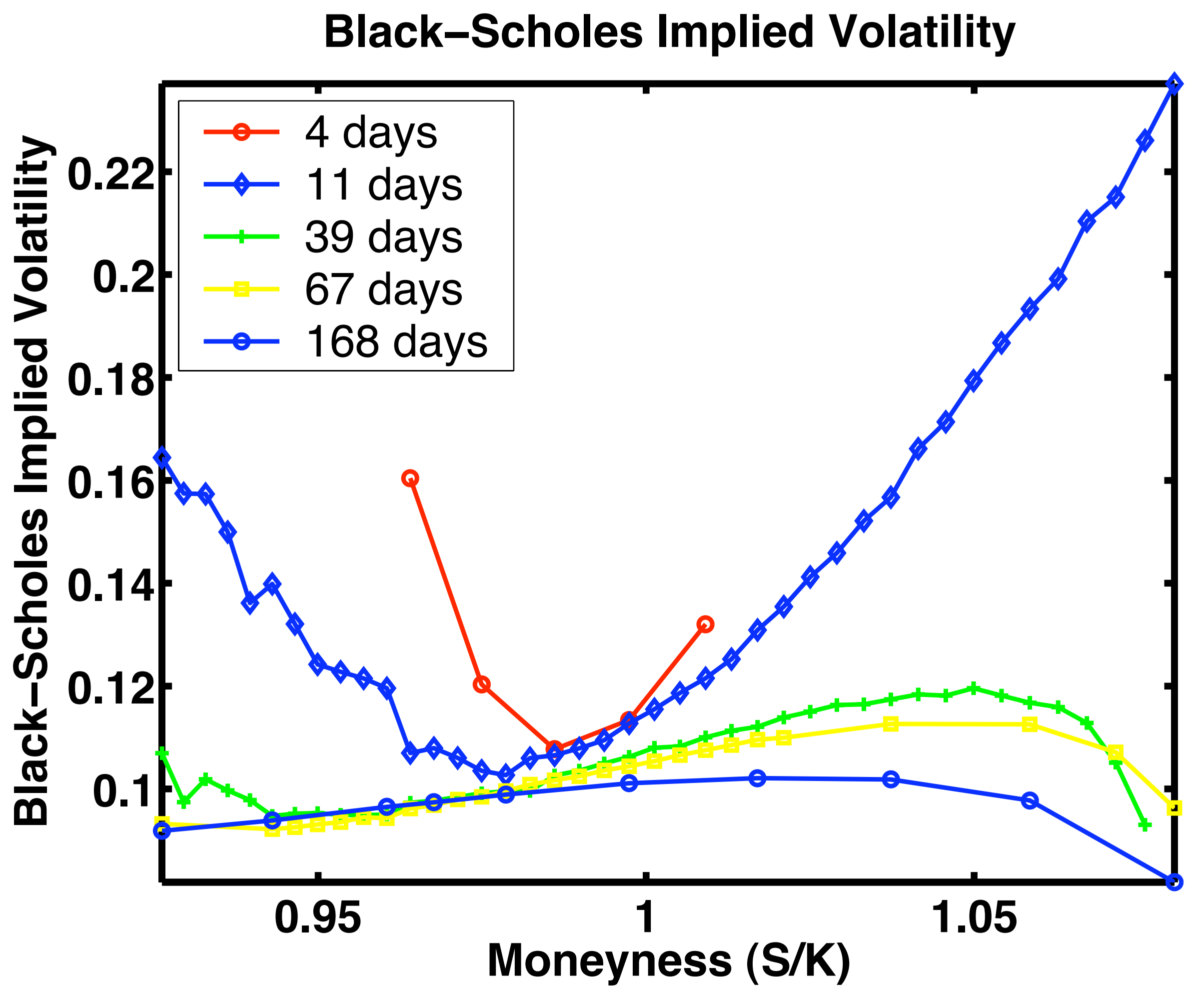
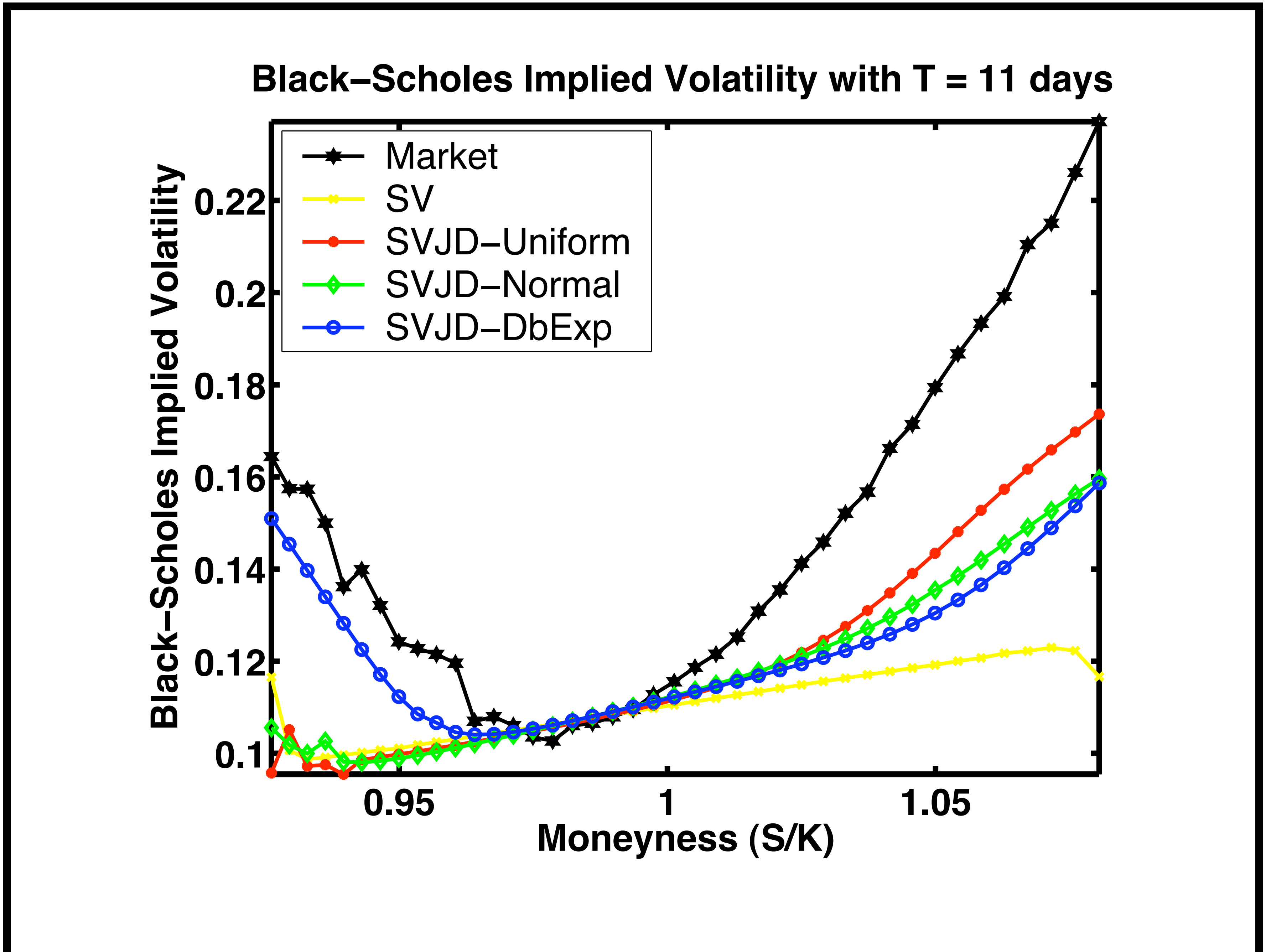
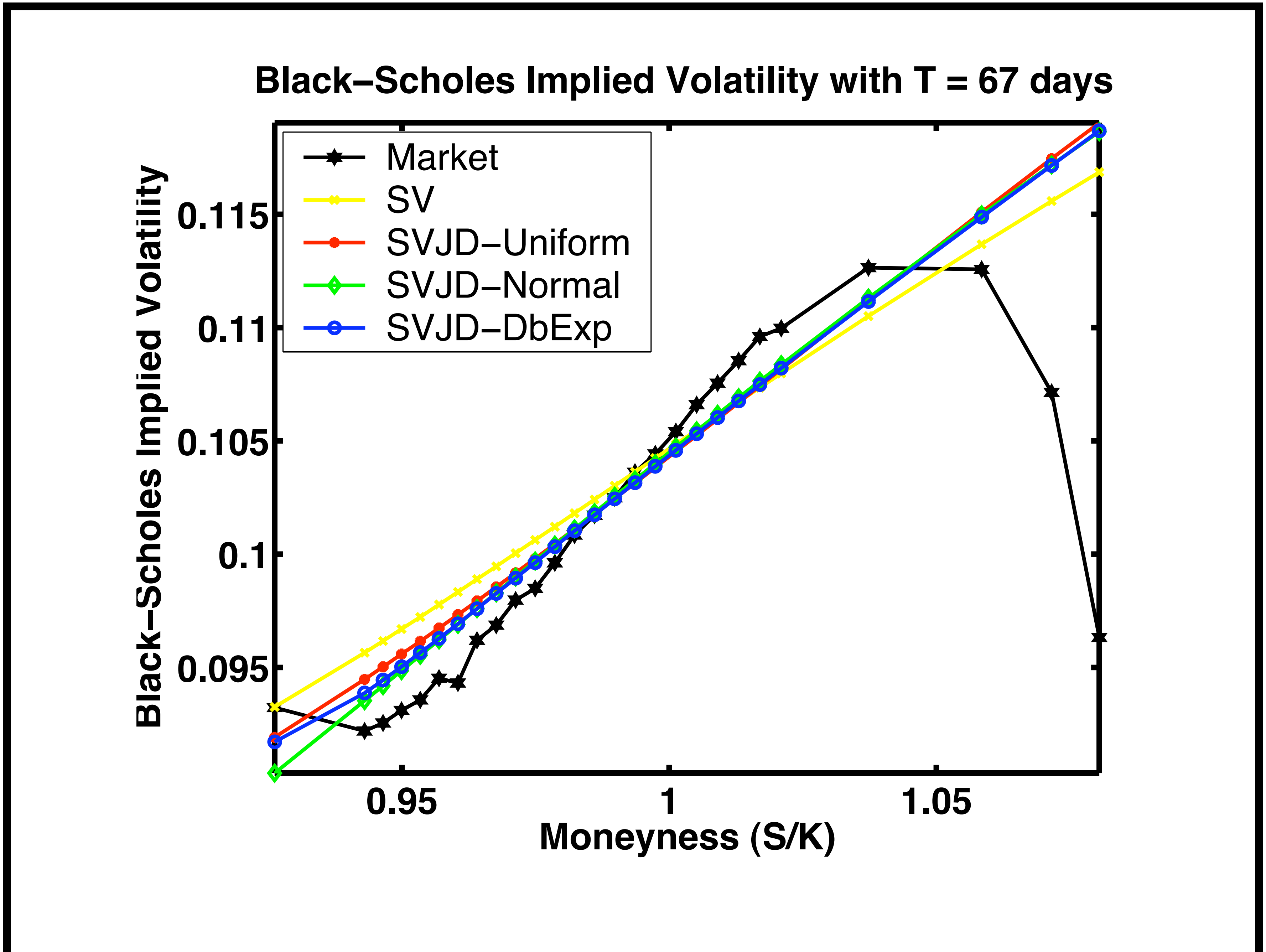


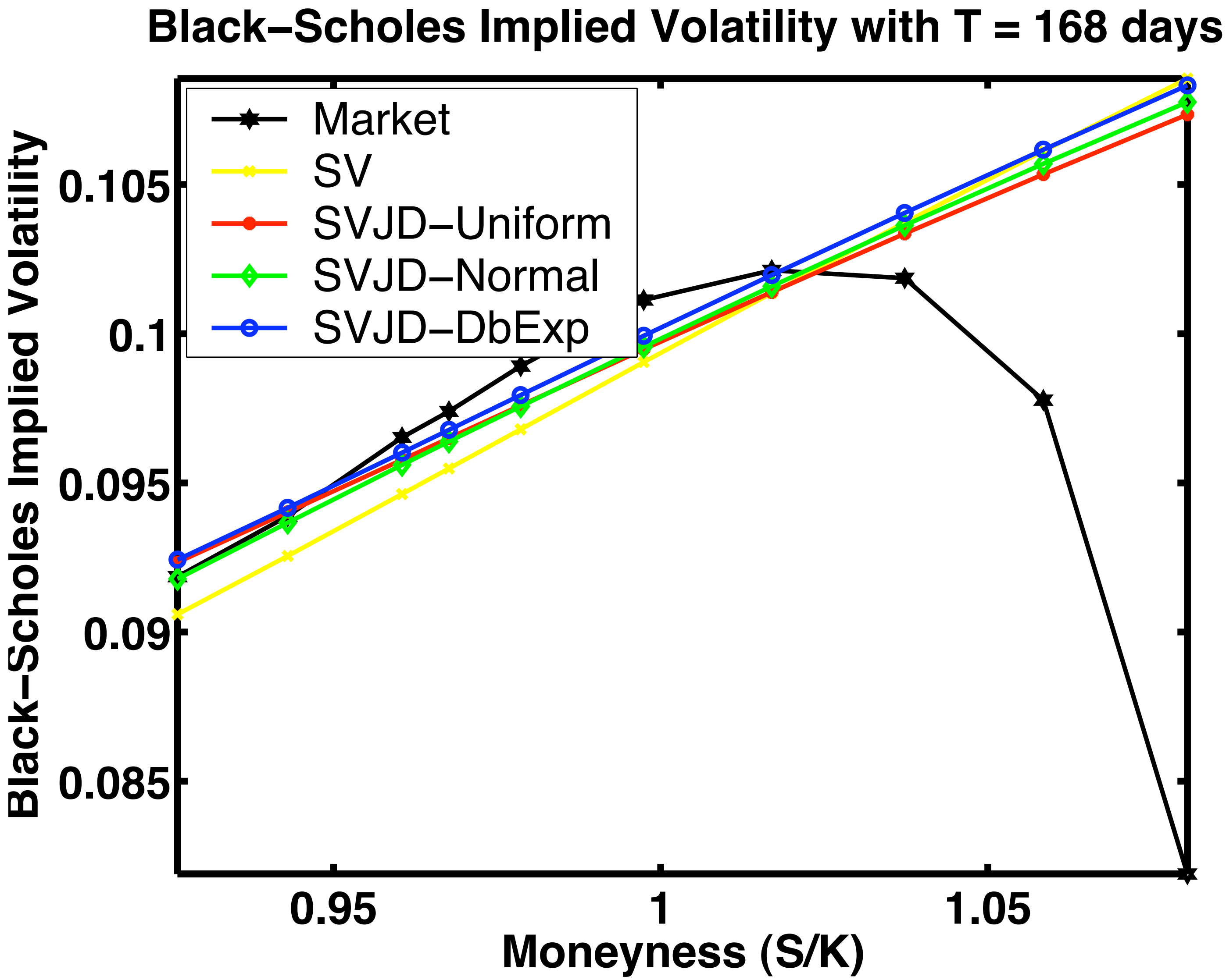
Figure 7.5: Black-Scholes implied volatility by SPX European call options of 5 different maturities, using a maximum likelihood to minimize the mean square error (MSE) between market observations and BS predictions. Option prices were quoted on April 10, 2006 (G. Yan, PhD Thesis, 2006).



Black–Scholes Implied Volatility with T = 11 days
Market
SV
SVJD–Uniform
SVJD–Normal
SVJD–DbExp
Black–Scholes Implied Volatility
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.95
1
1.05
Moneyness (S/K)



Black–Scholes Implied Volatility with T = 67 days
Market
SV
SVJD–Uniform
SVJD–Normal
SVJD–DbExp
Black–Scholes Implied Volatility
0.115
0.11
0.105
0.1
0.095
0.95
1
1.05
Moneyness (S/K)



Black–Scholes Implied Volatility with T = 168 days
Market
SV
SVJD–Uniform
SVJD–Normal
SVJD–DbExp
Black–Scholes Implied Volatility
0.105
0.1
0.095
0.09
0.085
0.95
1
1.05
Moneyness (S/K)

** Reminder: Lecture 7 Homework/Project Posted in Chalk Assignments, due in PDF by Lecture 9 in Chalk Assignments!*

*** Summary of Lecture 7:**

1. **Method of Moments**
2. **Vanilla, European Options**
3. **Market Calibration and Implied Volatility**
4. **Risk-Neutral Pricing and Implied Volatility**
5. **Compound-Jump-Diffusions and Option Pricing**
6. **Nonparametric, Multivariate Kernel Regression**
7. **Kernel Regression for Implied Volatility (IV) Computations**
8. **CBOE Market Quotes**
9. **Implied Volatility (pseudo) Algorithm**
10. **Kernel Smoothing Regression in MATLAB**
11. **Hybrid Root Finder `fzero`**