Examples

Eugene Morozov

(Eugene@HiEugene.com)

Contents

1 Kalman Filter

1.1 Least Mean-Square Error approach

Let's consider a linear, discrete-time dynamical system with its process equation:

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + \mathbf{w}_k, \tag{1}$$

where $\mathbf{F}_{k+1,k}$ is the transition matrix taking the (often hidden or latent) state $\mathbf{x}_k$ from time k to time $k+1$. The process noise $\mathbf{w}_k$ is assumed to be additive, white, and Gaussian, with zero mean and with covariance matrix defined by

$$E[\mathbf{w}_n\mathbf{w}_k^T] = \begin{cases} \mathbf{Q}_k & for\ n = k \\ \mathbf{0} & for\ n \neq k, \end{cases} \tag{2}$$

(therefore it's a Markov sequence.)
Measurement equation:

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{3}$$

where $\mathbf{y}_k$ is the observable variable at time k and $\mathbf{H}_k$ is the measurement matrix. The measurement noise $\mathbf{v}_k$ is assumed to be additive, white, and Gaussian, with zero mean and with covariance matrix defined by

$$E[\mathbf{v}_n\mathbf{v}_k^T] = \begin{cases} \mathbf{R}_k & for\ n = k \\ \mathbf{0} & for\ n \neq k. \end{cases} \tag{4}$$

Moreover, the measurement noise $\mathbf{v}_k$ is uncorrelated with the process noise $\mathbf{w}_k$, i.e. $cov(\mathbf{w}_i, \mathbf{v}_j) = 0$. Also $cov(\mathbf{x}_0, \mathbf{x}_0) \equiv var(\mathbf{x}_0) = \mathbf{P}_0$, $cov(\mathbf{x}_0, \mathbf{w}_k) = cov(\mathbf{x}_0, \mathbf{v}_k) = 0\ for\ all\ k$.

The goal is to find the minimum mean-square error estimate of $\mathbf{x}_k$ given $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$. Let $\hat{\mathbf{x}}_k$ denote the a posteriori estimate of the signal $\mathbf{x}_k$, given the observations $\mathbf{y}_i$, $i = 1, \ldots, k$. The state-error vector is defined by

$$\tilde{\mathbf{x}}_k \equiv \mathbf{x}_k - \hat{\mathbf{x}}_k. \tag{5}$$

In addition to having zero mean, it has covariance $\mathbf{P}_k$.
Let's define a cost (loss) function for incorrect estimates:

- The cost function is non-negative.
- The cost function is a non-decreasing function of the estimation error $\tilde{\mathbf{x}}_k$.

These two requirements are satisfied by the mean-square error defined by

$$\mathbf{J}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^2] = E[\tilde{\mathbf{x}}_k^2] \tag{6}$$

To derive an optimal value for the estimate $\hat{\mathbf{x}}_k$, we may invoke two theorems taken from stochastic process theory:

Theorem 1.1 Conditional mean estimator. *If the stochastic processes* $\mathbf{x}_k$ *and* $\mathbf{y}_k$ *are jointly Gaussian, then the optimum estimate* $\hat{\mathbf{x}}_k$ *that minimizes the mean-square error* $\mathbf{J}_k$ *is the conditional mean estimator:*

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k|\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k].$$

Theorem 1.2 Principle of orthogonality. *Let the stochastic processes* $\mathbf{x}_k$ *and* $\mathbf{y}_k$ *be of zero means; that is,* $E[\mathbf{x}_k] = E[\mathbf{y}_k] = 0$ *for all* k.

Then:

(i) the stochastic processes $\mathbf{x}_k$ *and* $\mathbf{y}_k$ *are jointly Gaussian; or*

(ii) if the optimal estimate $\hat{\mathbf{x}}_k$ *is restricted to be a linear function of the observables and the cost function is the mean-square error,*

(iii) then the optimum estimate $\hat{\mathbf{x}}_k$*, given the observables* $\mathbf{y}_k$*, is the orthogonal projection of* $\mathbf{x}_k$ *on the space spanned by these observables.*

Suppose that a measurement on a linear dynamical system, described by Eqs. (1) and (3), has been made at time k. The requirement is to use the information contained in the new measurement $\mathbf{y}_k$ to update the estimate of the unknown state $\mathbf{x}_k$. Let $\hat{\mathbf{x}}_k^-$ denote a priori estimate of the state, which is already available at time k. With a linear estimator as the objective, we may express the a posteriori estimate $\hat{\mathbf{x}}_k$ as a linear combination of the a priori estimate and the new measurement, as shown by

$$\hat{\mathbf{x}}_k = \mathbf{G}_k^* \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{y}_k, \tag{7}$$

where the multiplying matrix factors $\mathbf{G}_k^*$ and $\mathbf{G}_k$ are to be determined. To find these two matrices, we invoke the principle of orthogonality stated under Theorem 1.2:

$$E[\tilde{\mathbf{x}}_k \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \ldots, k-1 \tag{8}$$

Using (3), (7), and (5), (8), we get

$$E[(\mathbf{x}_k - \mathbf{G}_k^* \hat{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{G}_k \mathbf{v}_k)\mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \ldots, k-1 \tag{9}$$

The noise is uncorrelated: $E[\mathbf{v}_k \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \ldots, k-1$
since $E[\mathbf{v}_k(\mathbf{H}_i \mathbf{x}_i + \mathbf{v}_i)^T] = \mathbf{H}_i^T E[\mathbf{v}_k \mathbf{x}_i^T] = \mathbf{H}_i^T E[\mathbf{v}_k(\mathbf{F}_{i-1}\mathbf{x}_{i-1} + \mathbf{w}_{i-1})^T] = 0$
Therefore

$$E[(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k - \mathbf{G}_k^*)\mathbf{x}_k \mathbf{y}_i^T + \mathbf{G}_k^*(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)\mathbf{y}_i^T] = 0 \tag{10}$$

From the principle of orthogonality (both $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k$ are unbiased estimations) we now note that

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)\mathbf{y}_i^T] = 0 \tag{11}$$

Accordingly, (10) simplifies to

$$(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k - \mathbf{G}_k^*) E[\mathbf{x}_k \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \ldots, k-1 \tag{12}$$

For arbitrary values of the state $\mathbf{x}_k$ and the observable $\mathbf{y}_i$,(12) can only be satisfied if the scaling factors $\mathbf{G}_k$ and $\mathbf{G}_k^*$ are related as follows:

$$\mathbf{G}_k^* = \mathbf{I} - \mathbf{G}_k\mathbf{H}_k \tag{13}$$

Substituting (13) into (7), we may express the a posteriori estimate of the state at time k as

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\hat{\mathbf{x}}_k^- + \mathbf{G}_k\mathbf{y}_k \tag{14}$$

$$= \hat{\mathbf{x}}_k^- + \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-), \tag{15}$$

in light of which, the matrix $\mathbf{G}_k$ is called the Kalman gain.

Now let's derive an explicit formula for $\mathbf{G}_k$. Again, from the principle of orthogonality, similar to (11), we have

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)\mathbf{y}_k^T] = 0 \tag{16}$$

Let the error (innovation, i.e. a measure of the"new" information contained in $\mathbf{y}_k$)

$$\tilde{\mathbf{y}}_k = \hat{\mathbf{y}}_k^- - \mathbf{y}_k \tag{17}$$

$$= \mathbf{H}_k\hat{\mathbf{x}}_k^- - \mathbf{y}_k \tag{18}$$

The parameter $\hat{\mathbf{x}}_k$ depends linearly on $\mathbf{x}_k$, which depends linearly on $\mathbf{y}_k$. Therefore, from (16)

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)\hat{\mathbf{y}}_k^{T-}] = 0 \tag{19}$$

and also (by subtracting (16) from (19))

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)\tilde{\mathbf{y}}_k^T] = 0 \tag{20}$$

Using (3) and (15), we may express the state-error vector $\mathbf{x}_k - \hat{\mathbf{x}}_k$ as

$$\mathbf{x}_k - \hat{\mathbf{x}}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$$

$$= \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{G}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$$

$$= \mathbf{x}_k - \mathbf{G}_k\mathbf{H}_k\mathbf{x}_k - \hat{\mathbf{x}}_k^- + \mathbf{G}_k\mathbf{H}_k\hat{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{x}_k - (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\hat{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k \tag{21}$$

Hence, substituting (18) and (21) into (20), we get

$$E[((\mathbf{I}-\mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k)(\mathbf{H}_k\hat{\mathbf{x}}_k^- - \mathbf{y}_k)^T] = 0$$

$$= E[((\mathbf{I}-\mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k)(\mathbf{H}_k\hat{\mathbf{x}}_k^- - \mathbf{H}_k\mathbf{x}_k - \mathbf{v}_k)^T]$$

$$= E[((\mathbf{I}-\mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k)(\mathbf{H}_k\tilde{\mathbf{x}}_k^- + \mathbf{v}_k)^T] = 0 \tag{22}$$

Since the measurement noise $\mathbf{v}_k$ is independent of the state $\mathbf{x}_k$ and therefore the error $\tilde{\mathbf{x}}_k^-$, the expectation of (22) reduces to

$$(\mathbf{I}-\mathbf{G}_k\mathbf{H}_k)E[\tilde{\mathbf{x}}_k^- - \tilde{\mathbf{x}}_k^{T-}]\mathbf{H}_k^T - \mathbf{G}_k E[\mathbf{v}_k\mathbf{v}_k^T] = 0 \tag{23}$$

Define the a priori covariance matrix

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T]$$

$$= E[\tilde{\mathbf{x}}_k^-\tilde{\mathbf{x}}_k^{T-}] \tag{24}$$

Then we rewrite (23) as

$$(\mathbf{I}-\mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{R}_k = 0 \tag{25}$$

Solving this equation for $\mathbf{G}_k$, we get the desired formula

$$\mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{R}_k = 0$$

$$\mathbf{G}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k) = \mathbf{P}_k^-\mathbf{H}_k^T$$

$$\mathbf{G}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{26}$$

Similar for the a posteriori covariance

$$\mathbf{P}_k = E[\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^T] \tag{27}$$

By substituting (13) into (7), we obtain

$$\hat{\mathbf{x}}_k = (\mathbf{I}-\mathbf{G}_k\mathbf{H}_k)\hat{\mathbf{x}}_k^- + \mathbf{G}_k\mathbf{y}_k$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{28}$$

Subtract $\mathbf{x}_k$ from both sides of the latter equation to obtain

$$\hat{\mathbf{x}}_k - \mathbf{x}_k = \hat{\mathbf{x}}_k^- + \mathbf{G}_k\mathbf{H}_k\mathbf{x}_k + \mathbf{G}_k\mathbf{v}_k - \mathbf{G}_k\mathbf{H}_k\hat{\mathbf{x}}_k^- - \mathbf{x}_k$$

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{H}_k\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k$$

$$\tilde{\mathbf{x}}_k = (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k \tag{29}$$

By substituting (29) into (27) and noting that $E[\tilde{\mathbf{x}}_k^-\mathbf{v}_k^T] = 0$, we obtain

$$\mathbf{P}_k = E[((\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k)((\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k\mathbf{v}_k)^T]$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)E[\tilde{\mathbf{x}}_k^-\tilde{\mathbf{x}}_k^{T-}](\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)^T + \mathbf{G}_k E[\mathbf{v}_k\mathbf{v}_k^T]\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)^T + \mathbf{G}_k\mathbf{R}_k\mathbf{G}_k^T \tag{30}$$

This is so-called "Joseph form" of the covariance update equation. By substituting for $\mathbf{G}_k$ from (26), it can be put into the following form

$$\mathbf{P}_k = (\mathbf{P}_k^- - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-)(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)^T + \mathbf{G}_k\mathbf{R}_k\mathbf{G}_k^T$$

$$= \mathbf{P}_k^- - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{G}_k\mathbf{R}_k\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{G}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{G}_k^T \tag{31}$$

From (25)

$$\mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T = \mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{R}_k$$

Substituting this into (31)

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + (\mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{R}_k + \mathbf{G}_k\mathbf{R}_k)\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{32}$$

This is the one most often used in computation. It implements the effect that *conditioning on the measurement* has on the covariance matrix of estimation uncertainty.

Note: for better numerical stability (to preserve the positive definiteness) use the representation via the square root matrix.

Let's see how the covariance changes in time.

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}$$

For notational simplicity let $\hat{\mathbf{x}}_{k-1} \equiv \hat{\mathbf{x}}_{k-1}^+$ and $\mathbf{P}_{k-1} \equiv \mathbf{P}_{k-1}^+$.

Subtracting $\mathbf{x}_k$ from both sides to obtain

$$\hat{\mathbf{x}}_k^- - \mathbf{x}_k = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k$$

$$\tilde{\mathbf{x}}_k^- = \mathbf{F}_{k-1}(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) - \mathbf{w}_{k-1}$$

$$= \mathbf{F}_{k-1}\tilde{\mathbf{x}}_{k-1} - \mathbf{w}_{k-1} \tag{33}$$

for the propagation of the estimation error $\tilde{\mathbf{x}}$. Post-multiply it by $\tilde{\mathbf{x}}_k^{T-}$ and take the expected values. Use the fact that $E[\tilde{\mathbf{x}}_{k-1}\mathbf{w}_{k-1}^T] = 0$ to obtain the results

$$E[\tilde{\mathbf{x}}_k^-\tilde{\mathbf{x}}_k^{T-}] = \mathbf{P}_k^-$$

$$= \mathbf{F}_{k-1}E[\tilde{\mathbf{x}}_{k-1}\tilde{\mathbf{x}}_k^{T-}]$$

$$= \mathbf{F}_{k-1}E[\tilde{\mathbf{x}}_{k-1}\tilde{\mathbf{x}}_{k-1}^T]\mathbf{F}_{k-1}^T + E[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T]$$

$$= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{34}$$

which gives the a priori value of the covariance matrix of estimation uncertainty as a function of the previous a posteriori value.

These results obtained with the least-mean-squared estimation error do not depend on what probability distribution is used, as long as it has the required first and second moments.

1.2 Maximum Likelihood approach

Now let's look from the linear Gaussian maximum likelihood estimator prospective. We'll use the mean μ_x of X and the information matrix $Y_{xx} \equiv P_{xx}^{-1}$ as parameters for a Gaussian distribution.

$$p(x, \mu_x, P_{xx}) = \frac{1}{\sqrt{2\pi|P_{xx}|}} e^{-\frac{1}{2}(x-\mu_x)^T P_{xx}^{-1}(x-\mu_x)} \tag{35}$$

where P_{xx} is the covariance (second central moment).

The Gaussian likelihood function equivalent to (35) would be of the same form

$$\mathcal{L}(x, \mu_x, P_{xx}) = c\, e^{-\frac{1}{2}(x-\mu_x)^T P_{xx}^{-1}(x-\mu_x)} \tag{36}$$

and the log-likelihood

$$\ln \mathcal{L} = \ln\, c - \frac{1}{2}(x-\mu_x)^T Y_{xx}(x-\mu_x) \tag{37}$$

All covariance matrices in Kalman filtering are symmetric and positive definite, because the variances of estimated quantities are never absolutely zero.

Consequently, all covariance matrices P_{xx} in Kalman filtering will have a matrix inverse $Y_{xx} = P_{xx}^{-1}$, the corresponding information matrix. Also Y_{xx} will also be symmetric and positive definite. In fact, they have the same eigenvectors, and the corresponding eigenvalues of Y_{xx} will be the reciprocals of those of P_{xx}.

In Maximum Likelihood estimation, however, the information matrices Y_{xx} are only symmetric and non-negative definite (i.e. with zero eigenvalues possible) and, therefore, not necessarily invertible.

Using the information matrix in place of the covariance matrix in Gaussian likelihood functions allows us to model what estimation theorists would call"flat priors", a condition under which prior assumptions have no influence on the ultimate estimate. This cannot be done using covariance matrices, because it would require that some eigenvalues be infinite. It can be done using information matrices by allowing them to have zero eigenvalues whose eigenvectors represent linear combinations of the state space in which there is zero information. For example, information matrices can be used to represent the information in a measurement, and the dimension of which may be less than the dimension of the state vector.

Using the singular value decomposition we can write

$$Y_{xx} = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^T = V \; diag(\lambda) \; V^T$$

The Moore-Penrose generalized inverse of Y_{xx} can be defined in terms of its *svd* as

$$Y_{xx}^{\dagger} = \sum_{\lambda_i \neq 0} \lambda_i^{-1} \mathbf{v}_i \mathbf{v}_i^T \tag{38}$$

which is always symmetric and non-negative definite and of the same rank as Y_{xx}.

Two probability distributions are called statistically independent if and only if their joint probability is the product of the individual probabilities. The same is true for likelihoods. Let's denote the joint likelihood function $\mathcal{L}_c(x, \mu_c, Y_c)$ of two independent Gaussian likelihoods $\mathcal{L}_a(x, \mu_a, Y_a)$ and $\mathcal{L}_b(x, \mu_b, Y_b)$.

$$\mathcal{L}_c = c_c e^{-\frac{1}{2}(x-\mu_c)^T Y_c (x-\mu_c)}$$

$$= \mathcal{L}_a \mathcal{L}_b = c_a e^{-\frac{1}{2}(x-\mu_a)^T Y_a (x-\mu_a)} c_b e^{-\frac{1}{2}(x-\mu_b)^T Y_b (x-\mu_b)}$$

$$= c_a c_b e^{-\frac{1}{2}(x-\mu_a)^T Y_a (x-\mu_a) - \frac{1}{2}(x-\mu_b)^T Y_b (x-\mu_b)}$$

Taking the logarithm of both sides and differentiating once and twice with respect to x will yield the following sequence of equations:

$$\ln c_c - \frac{1}{2}(x-\mu_c)^T Y_c (x-\mu_c) = \ln c_a + \ln c_b - \frac{1}{2}(x-\mu_a)^T Y_a (x-\mu_a) - \frac{1}{2}(x-\mu_b)^T Y_b (x-\mu_b) \tag{39}$$

$$Y_c(x-\mu_c) = Y_a(x-\mu_a) + Y_b(x-\mu_b) \tag{40}$$

$$Y_c = Y_a + Y_b \tag{41}$$

the last line of which says that information is additive. Setting $x = 0$ in the next-to-last line yields the equation

$$Y_c\mu_c = Y_a\mu_a + Y_b\mu_b \tag{42}$$

The following substitutions will be made in (41) and (42):

$$\left.\begin{aligned} \mu_a &= \hat{\mathbf{x}}_k^- a\ priori\ estimate \\ Y_a &= \mathbf{P}_k^{-1(-)} a\ priori\ information \\ \mu_b &= \mathbf{H}_k^\dagger \mathbf{y}_k measurement\ mean \\ Y_b &= \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k measurement\ information \\ \mu_c &= \hat{\mathbf{x}}_k a\ posteriori\ estimate \\ Y_c &= \mathbf{P}_k^{-1} a\ posteriori\ information \end{aligned}\right\} \tag{43}$$

$$\mathbf{P}_k^{-1} = \mathbf{P}_k^{-1(-)} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \tag{44}$$

where $\mathbf{P}_k^{-1(-)}$ is the a priori state information and $\mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k$ is the information in the kth measurement $\mathbf{y}_k$. The measurement estimations come from (3) by taking the expectation

$$\mathbf{H}^{-1}\mathbf{y} = \mathbf{x} + \mathbf{H}^{-1}\mathbf{v}$$

$$E[\mathbf{H}^{-1}\mathbf{y}] = E[\mathbf{x}] + E[\mathbf{H}^{-1}\mathbf{v}]$$

the last term expands to $\mathbf{H}^{-1}\mathbf{R}\mathbf{H}^{-1(T)}$; reverse it to obtain the information $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$.

Using an inverse matrix modification formula

$$(A^{-1} + BC^{-1}D)^{-1} = A - AB(C + DAB)^{-1}DA \tag{45}$$

and substituting here

$$\left.\begin{aligned} A^{-1} &= \mathbf{P}_k^{-1(-)} a\ priori\ information\ matrix\ for\ \hat{\mathbf{x}}_k \\ A &= \mathbf{P}_k^- a\ priori\ covariance\ matrix\ for\ \hat{\mathbf{x}}_k \\ B &= \mathbf{H}_k^T transpose\ of\ measurement\ sensitivity\ matrix \\ C &= \mathbf{R}_k covariance\ of\ measurement\ noise\ \mathbf{v}_k \\ D &= \mathbf{H}_k measurement\ sensitivity\ matrix \end{aligned}\right\} \tag{46}$$

The equation (45) becomes

$$(\mathbf{P}_k^{-1(-)} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k)^{-1} = \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^- \tag{47}$$

using (44)

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{G}_k \mathbf{H}_k \mathbf{P}_k^- \tag{48}$$

where $\mathbf{G}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ (cf. with (32) and (26)).
Solving (42)

$$\mathbf{P}_k^{-1} \hat{\mathbf{x}}_k = \mathbf{P}_k^{-1(-)} \hat{\mathbf{x}}_k^- + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{H}_k^\dagger \mathbf{y}_k \tag{49}$$

$$\hat{\mathbf{x}}_k = \mathbf{P}_k (\mathbf{P}_k^{-1(-)} \hat{\mathbf{x}}_k^- + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{H}_k^\dagger \mathbf{y}_k) \tag{50}$$

Substituting (48) and expanding $\mathbf{G}_k$

$$\hat{\mathbf{x}}_k = (\mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^-)(\mathbf{P}_k^{-1(-)} \hat{\mathbf{x}}_k^- + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{H}_k^\dagger \mathbf{y}_k) \tag{51}$$

Opening the parenthesis and rearranging terms yields

$$\hat{\mathbf{x}}_k = [\mathbf{I} - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k](\hat{\mathbf{x}}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{H}_k^\dagger \mathbf{y}_k) \tag{52}$$

and again

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \{[(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{R}_k^{-1} - \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1}] \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-\} \tag{53}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \{[\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1} + \mathbf{I} - \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1}] \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-\} \tag{54}$$

and finally

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \tag{55}$$

Let's define here

$$\mathbf{G}_k \equiv \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{56}$$

Compare with (26) and (15) (or (28)).

1.3 Maximum A Posteriori Probability approach

There is an alternative class of estimators called maximum a posteriori probability (MAP) estimators which use Bayes' rule to compute the argmax of the a posteriori probability density function to select the value of the variable to be estimated at which its probability density is greatest (maximum mode) (i.e. maximize $\hat{\mathbf{x}}$ in (57)). These estimators are applicable to a more general class of problems (including non-Gaussian and nonlinear) than the Kalman filter, but they tend to have computational complexities that would eliminate them from consideration for real-time practical implementations as filters. They are used for some nonlinear and non-real-time applications, however.

Bayesian estimation combines a priori information with the measurements through a conditional density function of $\mathbf{x}$ given the measurements $\mathbf{y}$. This conditional probability density function is known as the a posteriori distribution of $\mathbf{x}$. Therefore, Bayesian estimation requires the probability density functions of both the measurement noise and unknown parameters. The posterior density function $p(\mathbf{x}|\mathbf{y})$ for $\mathbf{x}$ (taking the measurement sample $\mathbf{y}$ into account) is given by Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \tag{57}$$

Note since $\mathbf{y}$ is treated as a set of known quantities, then $p(\mathbf{y})$ provides the proper normalization factor to ensure that $p(\mathbf{x}|\mathbf{y})$ is a probability density function. Alternatively,

$$p(\mathbf{y}) = \int_{-\infty}^{\infty} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

If this integral exists then the posterior function $p(\mathbf{x}|\mathbf{y})$ is said to be *proper*; if it does not exist then $p(\mathbf{x}|\mathbf{y})$ is *improper*, in which case we let $p(\mathbf{y}) = 1$.

Since $p(\mathbf{y})$ does not depend on $\mathbf{x}$ we seek to maximize $p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, or its natural logarithm

$$J_{MAP}(\hat{\mathbf{x}}) = \ln p(\mathbf{y}|\hat{\mathbf{x}}) + \ln p(\hat{\mathbf{x}}) \tag{58}$$

The first term in the sum is actually the log-likelihood function, and the second term gives the a priori information on the to-be-determined parameters.

Therefore, the MAP estimator maximizes

$$J_{MAP}(\hat{\mathbf{x}}) = \ln \mathcal{L}(\mathbf{y}|\hat{\mathbf{x}}) + \ln p(\hat{\mathbf{x}}) \tag{59}$$

Maximum a posteriori estimation has the following properties:

1. if the a priori distribution $p(\hat{\mathbf{x}})$ is uniform, then MAP estimation is equivalent to maximum likelihood estimation;

2. MAP estimation shares the asymptotic consistency and efficiency properties of maximum likelihood estimation;

3. the MAP estimator converges to the maximum likelihood estimator for large samples;

4. the MAP estimator also obeys the invariance principle.

Let's consider a process following a Gaussian distribution. The assumed probability density functions for this case are given by

$$\mathcal{L}(\mathbf{y}|\hat{\mathbf{x}}) = p(\mathbf{y}|\hat{\mathbf{x}}) = \frac{1}{(2\pi)^{m/2}|\mathbf{R}|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{H}\hat{\mathbf{x}})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\hat{\mathbf{x}})} \tag{60}$$

$$p(\hat{\mathbf{x}}) = \frac{1}{(2\pi)^{n/2}|\mathbf{Q}|^{1/2}} e^{-\frac{1}{2}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}})^T\mathbf{Q}^{-1}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}})} \tag{61}$$

where $\mathbf{H}$ has the dimensions of $m \times n$.
Maximizing (59) w.r.t. $\hat{\mathbf{x}}$ leads to the following estimator:

$$\frac{d}{d\hat{\mathbf{x}}}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) + \frac{d}{d\hat{\mathbf{x}}}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}})^T\mathbf{Q}^{-1}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}}) = 0$$

$$\frac{d}{d\hat{\mathbf{x}}}(\mathbf{y}^T\mathbf{R}^{-1}\mathbf{y} - \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} - \mathbf{y}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}}) +$$

$$+\frac{d}{d\hat{\mathbf{x}}}(\hat{\mathbf{x}}_-^T\mathbf{Q}^{-1}\hat{\mathbf{x}}_- - \hat{\mathbf{x}}^T\mathbf{Q}^{-1}\hat{\mathbf{x}}_- - \hat{\mathbf{x}}_-^T\mathbf{Q}^{-1}\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\mathbf{Q}^{-1}\hat{\mathbf{x}}) = 0$$

$$-\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} - \mathbf{y}^T\mathbf{R}^{-1}\mathbf{H} + 2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}} - \mathbf{Q}^{-1}\hat{\mathbf{x}}_- - \hat{\mathbf{x}}_-^T\mathbf{Q}^{-1} + 2\hat{\mathbf{x}}^T\mathbf{Q}^{-1} = 0$$

$$-2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} + 2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}} - 2\mathbf{Q}^{-1}\hat{\mathbf{x}}_- + 2\mathbf{Q}^{-1}\hat{\mathbf{x}} = 0$$

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})^{-1}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} + \mathbf{Q}^{-1}\hat{\mathbf{x}}^-) \tag{62}$$

Let's compare this with our previous results. Substitute (26) into (15) to obtain

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^-) \tag{63}$$

For our Gaussian (61) $\hat{\mathbf{x}}^- = \mathbf{x} + \mathbf{w}$

$$\mathbf{P}^- = E[(\mathbf{x} - \mathbf{x} - \mathbf{w})(\mathbf{x} - \mathbf{x} - \mathbf{w})^T] = \mathbf{Q}$$

Then (63) becomes

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^-) \tag{64}$$

1. Let's consider the coefficient at $\mathbf{y}$:

$$(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \stackrel{?}{=} \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{R})^{-1}$$

$$\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{R}) \stackrel{?}{=} (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{H}^T = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{H}^T$$

2. and for $\hat{\mathbf{x}}^-$:

$$(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1})^{-1}\mathbf{Q}^{-1} \stackrel{?}{=} \mathbf{I}-\mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{H}$$

$$\mathbf{Q}^{-1} \stackrel{?}{=} \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1}-(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{H}$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \stackrel{?}{=} (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{H}$$

$$\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{R}) \stackrel{?}{=} (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{H}^T = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T+\mathbf{H}^T$$

Q.E.D.

1.4 Example

In this 2-D example I'll simulate a hound chasing a hare (or an air-to-air missile chasing a MiG). The hare is running along an ellipse (dashed red line). The hound is old and his eyes and nose are no so sharp as they used to be. He detects the hare with an error which is normally distributed around the hare position at time k (solid red dots). But his mind is still sharp. He keeps track of his own position (x_1, x_2), speed (v_1, v_2) and acceleration (a_1, a_2), and calculates his projected position at time $k+1$ (green dots):

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}t + \frac{1}{2}\mathbf{a}t^2 + \mathbf{w}(t) = \mathbf{x}_0 + \dot{\mathbf{x}}t + \frac{1}{2}\ddot{\mathbf{x}}t^2 + \mathbf{w}(t)$$

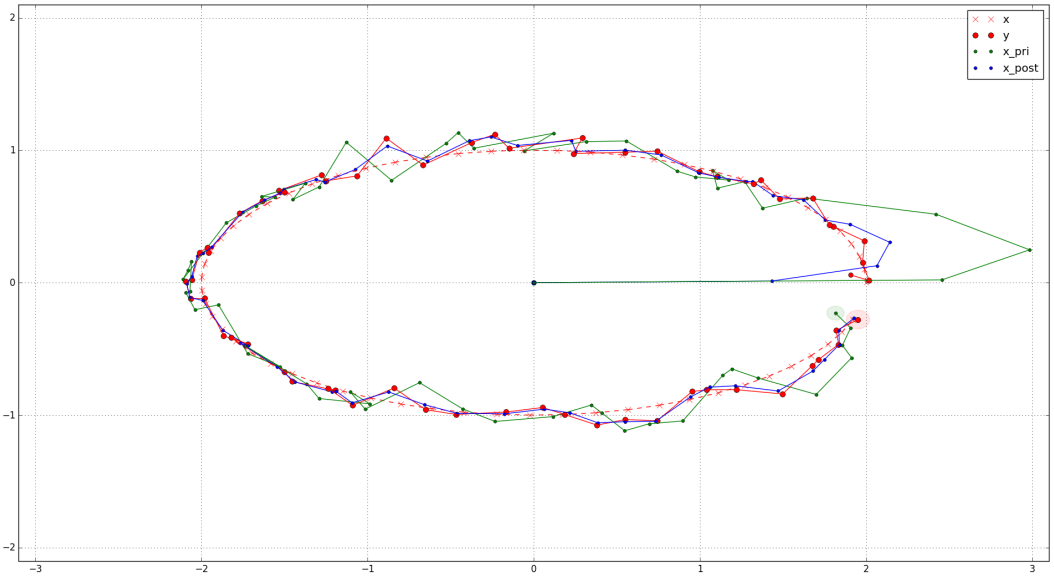
or in discrete form:

$$\mathbf{x}_{k+1} = \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \\ a_1 \\ a_2 \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & 0 & \triangle t & 0 & \frac{1}{2}\triangle t^2 & 0 \\ 0 & 1 & 0 & \triangle t & 0 & \frac{1}{2}\triangle t^2 \\ 0 & 0 & 1 & 0 & \triangle t & 0 \\ 0 & 0 & 0 & 1 & 0 & \triangle t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \\ a_1 \\ a_2 \end{pmatrix}_k + \mathcal{N}(0, q)$$

The process noise $\mathbf{w}$ is due to the paws slippage, wind and other factors not included in our model, and assumed to be Gaussian. As an example, the hound observes only his position $\mathbf{x}$ and not speed or acceleration, i.e.

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As mentioned his observation error is also assumed Gaussian: $\mathcal{N}(0, r)$ (solid red dots). The updated positions (the a posteriori approximations) are displayed as blue dots. The last dots also plot circles in light hue with the radius proportional to its respective variance. The initial approximation, arbitrary chosen as $(0, 0)$, is way off but it converges quite fast. As we can see, in most cases the a posteriori position is closer to the truth than both the prediction and observation.



2 Unscented Kalman filter

Let's consider a nonlinear, discrete-time dynamical system with its process equation:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{q}_k) \tag{65}$$

where $\mathbf{u}_k$ is the input vector. The process noise $\mathbf{q}_k$ caused by disturbances and modeling errors is assumed to be Gaussian (unlike in the particle filter), with zero mean and with covariance matrix defined by

$$E[\mathbf{q}_n \mathbf{q}_k^T] = \begin{cases} \mathbf{Q}_k & for\ n = k \\ \mathbf{0} & for\ n \neq k \end{cases} \tag{66}$$

In this case it is not additive as it transforms through f. The observation equation:

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{67}$$

where the measurement noise is Gaussian with zero mean and covariance:

$$E[\mathbf{v}_n \mathbf{v}_k^T] = \begin{cases} \mathbf{R}_k & for\ n = k \\ \mathbf{0} & for\ n \neq k \end{cases} \tag{68}$$

The noises are uncorrelated: $E[\mathbf{q}_n \mathbf{v}_k^T] = 0$.

We seek the minimum-mean-squared error (MMSE) estimate. The MMSE estimate of $\mathbf{x}(k)$ is the conditional mean. Let $\hat{\mathbf{x}}(i|j)$ be the mean of $\mathbf{x}(i)$ conditioned on all of the observations up to and including time j

$$\hat{\mathbf{x}}(i|j) = E[\mathbf{x}(i)|\mathbf{Y}^{(j)}] \tag{69}$$

where $\mathbf{Y}^{(j)} = [y(1), \ldots, y(j)]$. The covariance of this estimate is denoted $\mathbf{P}(i|j)$.

The Kalman filter propagates the first two moments of the distribution of $\mathbf{x}(k)$ recursively. Given an estimate $\hat{\mathbf{x}}(k|k)$, the filter first predicts what the future state of the system will be, using the process model.

$$\hat{\mathbf{x}}(k+1|k) = E[f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{q}(k))|\mathbf{Y}^{(k)}]$$

$$\mathbf{P}(k+1|k) = E[\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}^T|\mathbf{Y}^{(k)}]$$

The expectations can be calculated only if the distribution of $\mathbf{x}(k)$ conditioned on $\mathbf{Y}^{(k)}$ is known. In general, the distribution cannot be described by a finite number of parameters and most practical systems employ an approximation of some kind. Often the distribution of $\mathbf{x}(k)$ is assumed Gaussian at any time k. Two justifications are made. First, only the mean and covariance

need to be maintained. Second, given just the first two moments the Gaussian distribution is the entropy maximizing or least informative distribution.

The estimate at time $k+1$ is given through updating the prediction by the linear update rule. See (15), (30) and (26). The EKF exploits the fact that the error in the prediction, $\tilde{\mathbf{x}}(i|j) = \mathbf{x}(i) - \hat{\mathbf{x}}(i|j)$, can be attained by expanding (65), (67) as a Taylor series about the estimate $\hat{\mathbf{x}}(k|k)$. Truncating this series at the first order yields the approximate linear expression for the propagation of state error as

$$\tilde{\mathbf{x}}(k+1|k) \approx \nabla f_x \tilde{\mathbf{x}}(k|k) + \nabla f_q \mathbf{q}(k)$$

Using this approximation, the state prediction equations are

$$\hat{\mathbf{x}}(k+1|k) = f(\hat{\mathbf{x}}(k|k), \mathbf{u}(k), 0) \tag{70}$$

$$\mathbf{P}(k+1|k) = \nabla f_x \mathbf{P}(k|k) \nabla^T f_x + \nabla f_q \mathbf{Q}(k+1) \nabla^T f_q \tag{71}$$

The problems with EKF are that in many practical applications this linearization introduces significant biases or errors. Also calculating Jacobians at every prediction step can be cumbersome and time consuming.

We use the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation. Following this intuition, we generate a set of points whose sample mean and sample covariance are $\hat{\mathbf{x}}(k|k)$ and $\mathbf{P}(k|k)$, respectively. The nonlinear function is applied to each of these points in turn to yield a transformed sample, and the predicted mean and covariance are calculated from the transformed sample. Unlike in a Monte Carlo method, the sample are not drawn at random but rather carefully chosen so they capture specific information about the distribution. In general, this intuition can be applied to capture many kind of information about many types of distribution. Here we consider the special case of (i) capturing the mean and covariance of an (ii) assumed Gaussian distribution.

The n-dimensional random variable $\mathbf{x}(k)$ with mean $\hat{\mathbf{x}}(k|k)$ and covariance $\mathbf{P}(k|k)$ is approximated by $2n+1$ weighted samples or σ points selected by the algorithm

$$\begin{aligned}
\mathcal{X}_0(k|k) &= \hat{\mathbf{x}}(k|k) \\
W_0 &= \tfrac{\varkappa}{n+\varkappa} \\
\mathcal{X}_i(k|k) &= \hat{\mathbf{x}}(k|k) + \left(\sqrt{(n+\varkappa)\mathbf{P}(k|k)}\right)_i \\
W_i &= \tfrac{1}{2(n+\varkappa)} \\
\mathcal{X}_{i+n}(k|k) &= \hat{\mathbf{x}}(k|k) - \left(\sqrt{(n+\varkappa)\mathbf{P}(k|k)}\right)_i \\
W_{i+n} &= \tfrac{1}{2(n+\varkappa)}
\end{aligned} \tag{72}$$

where $\varkappa \in \mathbb{R}$, $\left(\sqrt{(n+\varkappa)\mathbf{P}(k|k)}\right)_i$ is the ith row (for $\mathbf{P} = \mathbf{A}^T\mathbf{A}$) or column (for $\mathbf{P} = \mathbf{A}\mathbf{A}^T$) of the matrix square root of $(n+\varkappa)\mathbf{P}(k|k)$, and W_i is the weight that is associated with the$i$th point.

Valuable insight into the Unscented Transformation can be gained by relating it to a numerical technique called the Gauss-Hermite quadrature rule in the context of state estimation. A close similarity also exists between the UT and the central difference interpolation filtering (CDF) techniques.

Theorem 1: The set of samples chosen by (72) have the same sample mean, covariance, and all higher odd-ordered central moments as the distribution of $\mathbf{x}(k)$. The matrix square root and $\varkappa$ affect the 4th and higher order sample moments of the sigma points.

Proof: The matching of the mean, covariance, and all odd-ordered moments can be directly demonstrated. Because the points are symmetrically distributed and chosen with equal weights about $\bar{\mathbf{x}}$, the sample mean is obviously $\bar{\mathbf{x}}$ and all odd-ordered moments are zero. The sample covariance $\mathbf{P}$ is

$$\begin{aligned}
\mathbf{P} &= \sum_{i=0}^{2n} W_i \left[\mathcal{X}_i(k|k) - \hat{\mathbf{x}}(k|k)\right] \left[\mathcal{X}_i(k|k) - \hat{\mathbf{x}}(k|k)\right]^T \\
&= \sum_{i=1}^{n} 2W_i(n+\varkappa) \left(\sqrt{\mathbf{P}(k|k)}\right)_i \left(\sqrt{\mathbf{P}(k|k)}\right)_i^T \\
&= \sum_{i=1}^{n} \left(\sqrt{\mathbf{P}(k|k)}\right)_i \left(\sqrt{\mathbf{P}(k|k)}\right)_i^T \\
&= \mathbf{P}(k|k)
\end{aligned}$$

■

Remark 1: The above properties hold for any choice of the matrix square root. Efficient and stable methods, such as Cholesky decomposition, should be used.

Given the set of samples generated by (72), the prediction procedure is as follows.

1. The state is augmented with the noises:

$$\mathbf{x}_a(k) = \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{q}(k) \\ \mathbf{v}(k) \end{pmatrix}$$

$$\hat{\mathbf{x}}_a(k) = \begin{pmatrix} \hat{\mathbf{x}}(k) \\ E[\mathbf{q}(k)] \\ E[\mathbf{v}(k)] \end{pmatrix}$$

or in our case

$$\hat{\mathbf{x}}_a(k) = \begin{pmatrix} \hat{\mathbf{x}}(k) \\ 0 \\ 0 \end{pmatrix}$$

n will now be the size of the vector $\mathbf{x}_a(k)$. The covariance matrix is augmented

$$\mathbf{P}_a(k) = \begin{pmatrix} \mathbf{P}(k) & \mathbf{P}_{xq}(k) & \mathbf{P}_{xv}(k) \\ \mathbf{P}_{xq}^T(k) & \mathbf{Q}(k) & \mathbf{P}_{qv}(k) \\ \mathbf{P}_{xv}^T(k) & \mathbf{P}_{qv}^T(k) & \mathbf{R}(k) \end{pmatrix}$$

or in our case

$$\mathbf{P}_a(k) = \begin{pmatrix} \mathbf{P}(k) & 0 & 0 \\ 0 & \mathbf{Q}(k) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{pmatrix}$$

2. The initial values

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)]$$

$$\mathbf{P}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T]$$

$$\hat{\mathbf{x}}_a(0) = E[\mathbf{x}_a(0)] = \begin{pmatrix} \hat{\mathbf{x}}(0) \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{P}_a(0) = E[(\mathbf{x}_a(0) - \hat{\mathbf{x}}_a(0))(\mathbf{x}_a(0) - \hat{\mathbf{x}}_a(0))^T] = \begin{pmatrix} \mathbf{P}(0) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{pmatrix}$$

3. Calculate the σ points

$$\mathcal{X}_i^a(k) = \begin{cases} \hat{\mathbf{x}}_a(k) & i = 0 \\ \hat{\mathbf{x}}_a(k) + \left(\sqrt{(n+\varkappa)\mathbf{P}_a(k)}\right)_i & i = 1, \ldots, n \\ \hat{\mathbf{x}}_a(k) - \left(\sqrt{(n+\varkappa)\mathbf{P}_a(k)}\right)_{i-n} & i = n+1, \ldots, 2n \end{cases}$$

for ith column.

4. Each σ point is instantiated through the process model to yield a set of transformed samples

$$\mathcal{X}_i^a(k+1|k) = f(\mathcal{X}_i^a(k|k), \mathbf{u}(k), k) \tag{73}$$

5. The predicted mean is computed as

$$\hat{\mathbf{x}}^{(-)}(k+1|k) = \sum_{i=0}^{2n} W_i \mathcal{X}_i^a(k+1|k) \tag{74}$$

6. The predicted covariance is computed as

$$\mathbf{P}^{(-)}(k+1|k) = \sum_{i=0}^{2n} W_i [\mathcal{X}_i^a(k+1|k) - \hat{\mathbf{x}}^{(-)}(k+1|k)][\mathcal{X}_i^a(k+1|k) - \hat{\mathbf{x}}^{(-)}(k+1|k)]^T \tag{75}$$

7. The predicted observation

$$\mathcal{Y}(k+1|k) = h(\mathcal{X}^a(k+1|k)) \tag{76}$$

8. The mean observation is given by

$$\hat{\mathbf{y}}^{(-)}(k+1|k) = \sum_{i=0}^{2n} W_i \mathcal{Y}_i(k+1|k) \tag{77}$$

9. The update step

$$\mathbf{P}_{yy}(k+1|k+1) = \sum_{i=0}^{2n} W_i \left[\mathcal{Y}_i(k+1|k) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right] \left[\mathcal{Y}_i(k+1|k) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right]^T \tag{78}$$

$$\mathbf{P}_{xy}(k+1|k+1) = \sum_{i=0}^{2n} W_i \left[\mathcal{X}_i^a(k+1|k) - \hat{\mathbf{x}}^{(-)}(k+1|k)\right] \left[\mathcal{Y}_i(k+1|k) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right]^T \tag{79}$$

$$\mathbf{G}(k+1|k+1) = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1} \tag{80}$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}^{(-)}(k+1|k) + \mathbf{G}(k+1|k+1)\left[\mathbf{y}(k+1|k+1) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right] \tag{81}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}^{(-)}(k+1|k) - \mathbf{G}(k+1|k+1)\mathbf{P}_{yy}(k+1|k+1)\mathbf{G}^T(k+1|k+1) \tag{82}$$

The innovation vector (or measurement prediction error or residual) points from our predicted measurement to the actual measurement: $\mathbf{y}(k+1|k+1) - \hat{\mathbf{y}}^{(-)}(k+1|k)$, and $\mathbf{G}$ determines how a vector in measurement space maps to a correction in state space.

Theorem 2: The prediction algorithm introduces errors in estimating the mean and covariance at the 4th and higher orders in the Taylor series. These higher order terms are a function of $\varkappa$ and the matrix square root used.

Proof: Let's consider a Gaussian-distributed random variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_x$. We wish to calculate the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_y$ of the random variable $\mathbf{y}$, which is related to $\mathbf{x}$ through the nonlinear analytic function $\mathbf{y} = f(\mathbf{x})$. Note that $\mathbf{y}$ here is not the observable variable in the KF but rather corresponds to the prior $\mathbf{x}^{(-)}$.

Noting that $\mathbf{x}$ can be written as $\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}$, where $\delta\mathbf{x}$ is a zero-mean Gaussian random variable with covariance $\mathbf{P}_x$, the nonlinear transformation can be expanded as a Taylor series about $\bar{\mathbf{x}}$

$$\mathbf{y} = f(\bar{\mathbf{x}} + \delta\mathbf{x}) = \sum_{i=0}^{\infty} \left[\frac{(\delta\mathbf{x} \cdot \nabla_x)^i f(\mathbf{x})}{i!}\right]_{\mathbf{x}=\bar{\mathbf{x}}} \tag{83}$$

If we define the operator $\mathbf{D}^i_{\delta\mathbf{x}} f$ as

$$\mathbf{D}^i_{\delta\mathbf{x}} f \equiv \left[(\delta\mathbf{x} \cdot \nabla_x)^i f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}}$$

then the Taylor series expansion of the nonlinear transformation $\mathbf{y} = f(\mathbf{x})$ can be written as

$$\mathbf{y} = f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} f + \frac{1}{2}\mathbf{D}^2_{\delta\mathbf{x}} f + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}} f + \frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}} f + \dots \tag{84}$$

The true mean of $\mathbf{y}$ is given by

$$\bar{\mathbf{y}} = E[\mathbf{y}] = E[f(\mathbf{x})]$$

$$= E\left[f(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} f + \frac{1}{2}\mathbf{D}^2_{\delta\mathbf{x}} f + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}} f + \frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}} f + \dots \right] \tag{85}$$

If we assume that $\mathbf{x}$ is a symmetrically distributed random variable, then all odd moments will be zero (expectation which is an integral of $\delta\mathbf{x}$ which is $\mathbf{x} - \bar{\mathbf{x}}$ is 0). Also note that

$$E\left[\delta\mathbf{x}\delta\mathbf{x}^T \right] = \mathbf{P}_x \tag{86}$$

Given this, the mean can be reduced further to

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \frac{1}{2}\left[\left(\nabla^T \mathbf{P}_x \nabla \right) f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}} + E\left[\frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}} f + \frac{1}{6!}\mathbf{D}^6_{\delta\mathbf{x}} f + \dots \right] \tag{87}$$

The Unscented Transformation calculates the posterior mean from the propagated σ points using (74). The sigma points are given by (72), i.e.

$$\mathcal{X}_i = \bar{\mathbf{x}} \pm \left(\sqrt{n + \varkappa} \right) \sigma_i = \bar{\mathbf{x}} \pm \tilde{\sigma}_i$$

where σ_i denotes the ith column of the matrix square root of $\mathbf{P}_x$. This implies that

$$\sum_{i=1}^{n} (\sigma_i \sigma_i^T) = \mathbf{P}_x \tag{88}$$

Given this formulation of the sigma points, we can again write the propagation of each point through the nonlinear function as a Taylor series expansion about $\bar{\mathbf{x}}$:

$$f(\mathcal{X}_i) = f(\bar{\mathbf{x}}) + \mathbf{D}_{\tilde{\sigma}_i} f + \frac{1}{2}\mathbf{D}^2_{\tilde{\sigma}_i} f + \frac{1}{3!}\mathbf{D}^3_{\tilde{\sigma}_i} f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i} f + \dots \tag{89}$$

Using (74) and (72), the UT predicted mean is

$$\bar{\mathbf{y}}_{UT} = \frac{\varkappa}{n+\varkappa} f(\bar{\mathbf{x}}) + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(f(\bar{\mathbf{x}}) + \mathbf{D}_{\tilde{\sigma}_i} f + \frac{1}{2}\mathbf{D}^2_{\tilde{\sigma}_i} f + \frac{1}{3!}\mathbf{D}^3_{\tilde{\sigma}_i} f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i} f + \dots \right) =$$

$$= f(\bar{\mathbf{x}}) + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(\mathbf{D}_{\tilde{\sigma}_i} f + \frac{1}{2} \mathbf{D}^2_{\tilde{\sigma}_i} f + \frac{1}{3!} \mathbf{D}^3_{\tilde{\sigma}_i} f + \frac{1}{4!} \mathbf{D}^4_{\tilde{\sigma}_i} f + \dots \right) \tag{90}$$

Since the sigma points are symmetrically distributed around $\bar{\mathbf{x}}$, all the odd moments are zero. This results in the simplification

$$\bar{\mathbf{y}}_{UT} = f(\bar{\mathbf{x}}) + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(\frac{1}{2} \mathbf{D}^2_{\tilde{\sigma}_i} f + \frac{1}{4!} \mathbf{D}^4_{\tilde{\sigma}_i} f + \dots \right) \tag{91}$$

and since

$$\frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \frac{1}{2} \mathbf{D}^2_{\tilde{\sigma}_i} f = \frac{1}{2(n+\varkappa)} (\nabla f)^T \left(\frac{1}{2} \sum_{i=1}^{2n} \sqrt{n+\varkappa} \sigma_i \sigma_i^T \sqrt{n+\varkappa} \right) \nabla f =$$

$$= \frac{n+\varkappa}{2(n+\varkappa)} (\nabla f)^T \frac{1}{2} \left(\sum_{i=1}^{2n} \sigma_i \sigma_i^T \right) \nabla f =$$

$$= \frac{1}{2} \left[\left(\nabla^T \mathbf{P}_x \nabla \right) f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}}$$

the UT predicted mean can be further simplified to

$$\bar{\mathbf{y}}_{UT} = f(\bar{\mathbf{x}}) + \frac{1}{2} \left[\left(\nabla^T \mathbf{P}_x \nabla \right) f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(\frac{1}{4!} \mathbf{D}^4_{\tilde{\sigma}_i} f + \frac{1}{6!} \mathbf{D}^6_{\tilde{\sigma}_i} f + \dots \right) \tag{92}$$

When we compare (92) and (87), we can clearly see that the true posterior mean and the mean calculated by the UT agrees exactly to the third order and that errors are only introduced in the 4th and higher order terms. The magnitudes of these errors depends on the choice of the composite scaling parameter $\varkappa$ as well as the higher-order derivatives of f. The parameter $\varkappa$ provides an extra degree of freedom to"fine tune" the higher order moments of the approximation, and can be used to reduce the overall prediction errors. When $\mathbf{x}(k)$ is assumed Gaussian, a useful heuristic is to select $n + \varkappa = 3$. If a different distribution is assumed for $\mathbf{x}(k)$, then a different choice of $\varkappa$ might be more appropriate.

Now let's look at the *accuracy of the covariance*. The true posterior covariance is given by

$$\mathbf{P}_y = E \left[(\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})^T \right] \tag{93}$$

where the expectation is taken over the distribution of $\mathbf{y}$. Substituting (84) and (85) into (93) we get

$$\mathbf{y} - \bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} f + \frac{1}{2!} \mathbf{D}^2_{\delta\mathbf{x}} f + \frac{1}{3!} \mathbf{D}^3_{\delta\mathbf{x}} f + \dots - f(\bar{\mathbf{x}}) - E \left[\frac{1}{2!} \mathbf{D}^2_{\delta\mathbf{x}} f + \frac{1}{4!} \mathbf{D}^4_{\delta\mathbf{x}} f + \dots \right]$$

post multiplying the state error by the transpose of itself, taking the expectation, and recalling that all odd moments of $\delta\mathbf{x}$ are zero owing to symmetry (e.g. $E[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\cdot(\mathbf{D}_{\delta\mathbf{x}}f)^T]=0$)

$$\mathbf{P}_y = E\left[\mathbf{D}_{\delta\mathbf{x}}f\cdot(\mathbf{D}_{\delta\mathbf{x}}f)^T+\frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}}f\cdot(\mathbf{D}_{\delta\mathbf{x}}f)^T+\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\cdot(\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f)^T+\mathbf{D}_{\delta\mathbf{x}}f\cdot(\frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}}f)^T+\ldots\right]-$$

$$-E\left[E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right]\cdot(\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f)^T+\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\cdot\left(E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right]\right)^T-E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right]E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right]^T+\ldots\right]$$

Limiting to the 3rd order and using (86)

$$\mathbf{P}_y = \left[\left(\nabla\mathbf{P}_x\nabla^T\right)f(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}}+O(\delta\mathbf{x}^4) \tag{94}$$

The unscented KF predicts the covariance using (see (72))

$$\mathcal{X}_0=\bar{\mathbf{x}}$$

$$\mathcal{X}_i=\bar{\mathbf{x}}\pm\sigma_i$$

which assures that

$$\mathbf{P}_x=\frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}(\mathcal{X}_i-\bar{\mathbf{x}})(\mathcal{X}_i-\bar{\mathbf{x}})^T$$

The transformed set of sigma points are evaluated by

$$\mathcal{Y}_i=f(\mathcal{X}_i)$$

The predicted mean is computed as

$$\bar{\mathbf{y}}=\frac{1}{n+\varkappa}\left(\varkappa\mathcal{Y}_0+\frac{1}{2}\sum_{i=1}^{2n}\mathcal{X}_i\right)$$

And the predicted covariance is computed as

$$\mathbf{P}_y=\sum_{i=0}^{2n}W_i(\mathcal{X}_i-\bar{\mathbf{x}})(\mathcal{X}_i-\bar{\mathbf{x}})^T=\frac{1}{n+\varkappa}\left[\varkappa(\mathcal{Y}_0-\bar{\mathbf{y}})(\mathcal{Y}_0-\bar{\mathbf{y}})^T+\frac{1}{2}\sum_{i=1}^{2n}(\mathcal{Y}_i-\bar{\mathbf{y}})(\mathcal{Y}_i-\bar{\mathbf{y}})^T\right] \tag{95}$$

Here

$$\mathcal{Y}_0-\bar{\mathbf{y}}=f(\bar{\mathbf{x}})-f(\bar{\mathbf{x}})-\frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\bar{\sigma}_i}f+\frac{1}{4!}\mathbf{D}^4_{\bar{\sigma}_i}f+\ldots\right)$$

$$(\mathcal{Y}_0-\bar{\mathbf{y}})(\mathcal{Y}_0-\bar{\mathbf{y}})^T = \frac{1}{4(n+\varkappa)^2}\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f+\frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i}f+\ldots\right)\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f+\frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i}f+\ldots\right)^T = O(\tilde{\sigma}^4)$$

Using (89) and (91)

$$\mathcal{Y}_i-\bar{\mathbf{y}} = \mathbf{D}_{\tilde{\sigma}_i}f+\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f+\frac{1}{3!}\mathbf{D}^3_{\tilde{\sigma}_i}f+\ldots-\frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f+\frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i}f+\ldots\right)$$

$$(\mathcal{Y}_i-\bar{\mathbf{y}})(\mathcal{Y}_i-\bar{\mathbf{y}})^T = \mathbf{D}_{\tilde{\sigma}_i}f\cdot(\mathbf{D}_{\tilde{\sigma}_i}f)^T+O(\tilde{\sigma}^4)$$

Plugging these back to (95) yields

$$\mathbf{P}_y = \frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}\mathbf{D}_{\tilde{\sigma}_i}f\cdot(\mathbf{D}_{\tilde{\sigma}_i}f)^T+O(\tilde{\sigma}^4) =$$

$$= \frac{1}{2(n+\varkappa)}\nabla f\left(\sum_{i=1}^{2n}\sqrt{n+\varkappa}\sigma_i\sigma_i^T\sqrt{n+\varkappa}\right)(\nabla f)^T+O(\tilde{\sigma}^4)$$

and substituting (88)

$$\mathbf{P}_y = \nabla f\mathbf{P}_x(\nabla f)^T+O(\tilde{\sigma}^4)$$

Compare it with (94)

■

To reiterate, when a set of carefully chosen sample points (σ points) propagated through the nonlinear system, for non-Gaussian inputs, they capture the posterior mean and covariance accurately to at least the 2nd order (Taylor series expansion) for any nonlinearity, with the accuracy of 3rd and higher order moments being determined by the choice of $\varkappa$. For a symmetrical (e.g. Gaussian) pdf the accuracy is at least the 3rd order.

2.1 Mackey-Glass Example

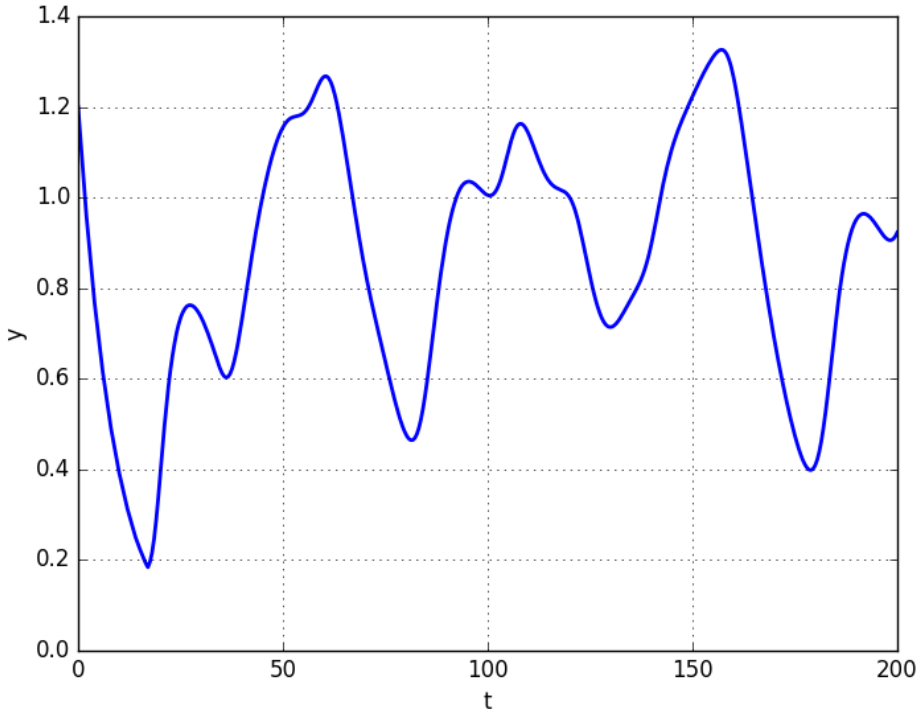
For this example I'll try to predict the Mackey-Glass chaotic series which are defined by

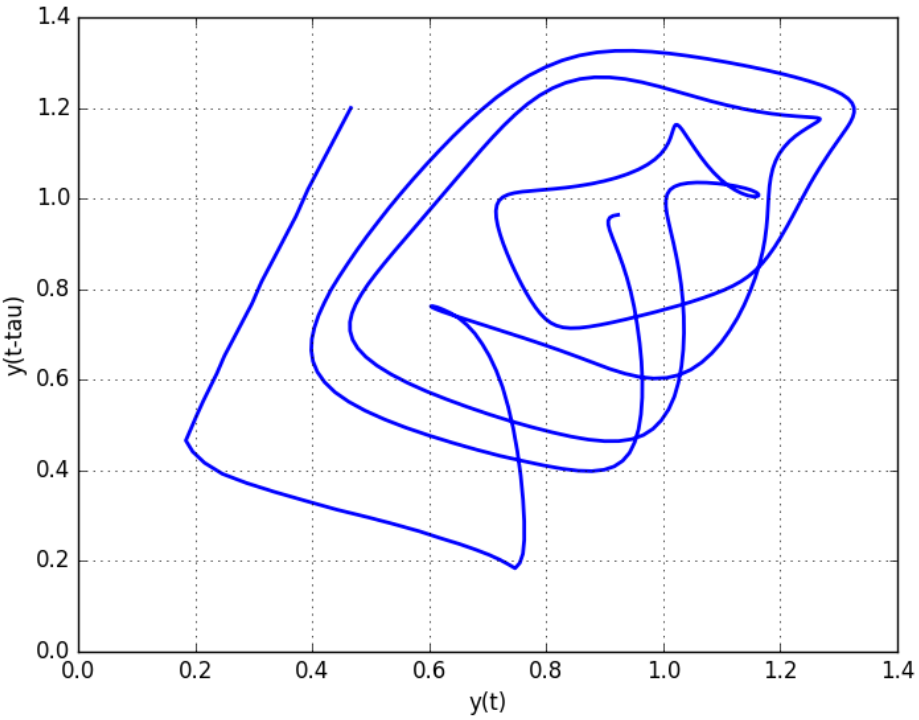
$$\dot{\mathbf{x}} = \beta\frac{\mathbf{x}(t-\tau)}{1+\mathbf{x}(t-\tau)^n}-\gamma\mathbf{x}(t) \tag{96}$$

or for a discrete time t

$$\mathbf{x}(t) = \mathbf{x}(t-1)+\left[\beta\frac{\mathbf{x}(t-\tau)}{1+\mathbf{x}(t-\tau)^n}-\gamma\mathbf{x}(t-1)\right]\triangle t$$

For the delay $\tau = 17$, $\beta = 0.2$, $\gamma = 0.1$, $n = 10$, $\triangle t = 0.1 and x = 1.2 for t \leq 0$ the series look like



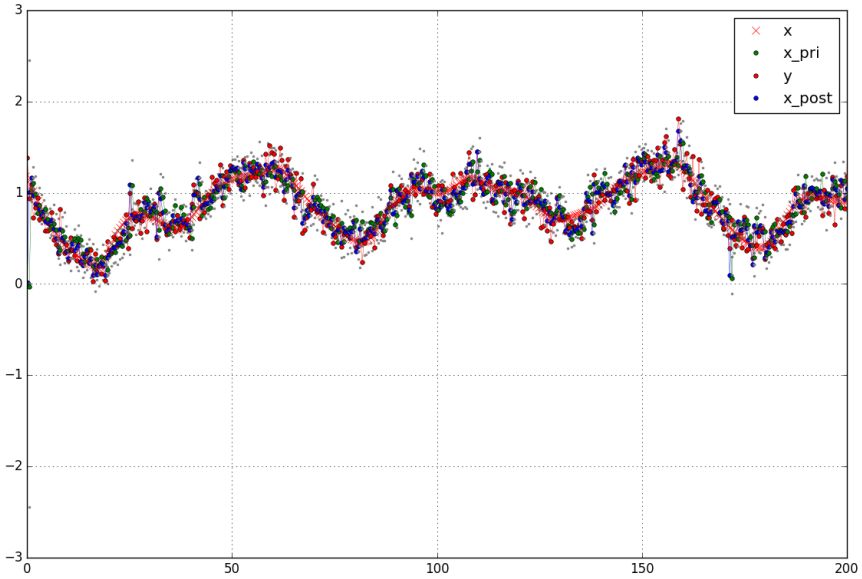


Our augmented state will consist of a scalar function value x and scalar noises:

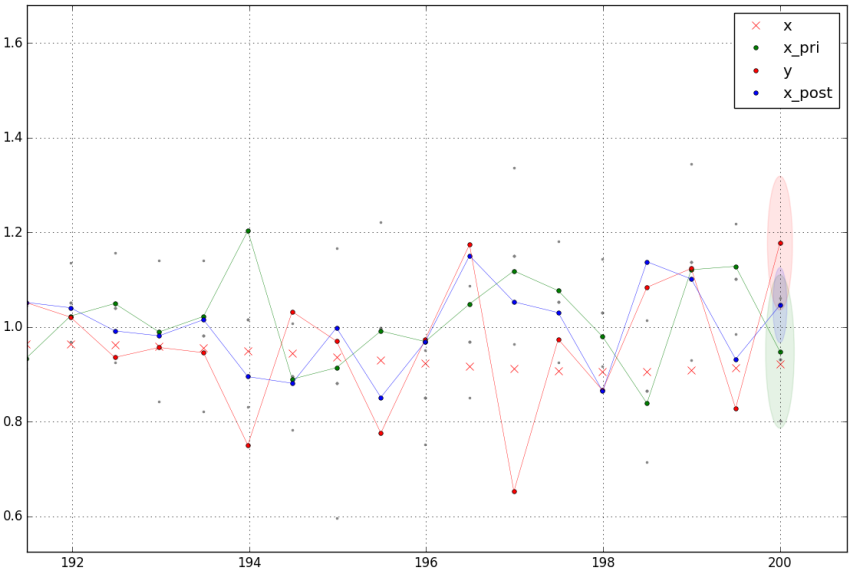
$$\mathbf{x}_a(k) = \begin{pmatrix} x(k) \\ q(k) \\ v(k) \end{pmatrix}_{3\times 1}$$

The initial position is arbitrary chosen at $x = 0$. The initial covariance matrix is equal to

$$\mathbf{P}_a(0) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.02 \end{pmatrix}$$



and zoom in at the end



The gray dots indicate the sigma points. The mean a priori values $\hat{\mathbf{x}}^{(-)}$

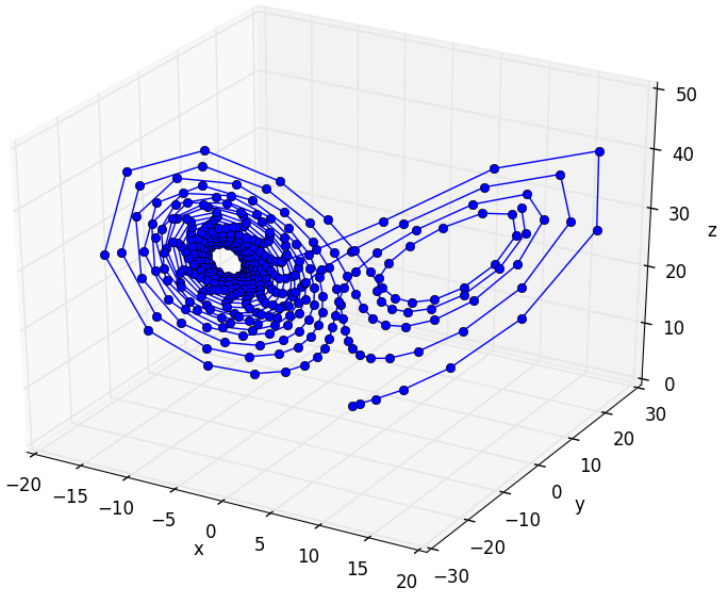
(predictions) are shown in green. The greenish circle of uncertainty has the radius of $\left(\sqrt{\mathbf{P}^{(-)}}\right)_{0,0}$. The observations $\mathbf{y}$ are depicted with red dots. We observe only the first component with some additive Gaussian noise with its standard deviation equal to the square root of r (shown as a reddish circle). The updated a posteriori values $\hat{\mathbf{x}}$ are blue dots with the circle of uncertainty shown in bluish. Its radius is $\left(\sqrt{\mathbf{P}}\right)_{0,0}$.

2.2 Lorenz Attractor Example

The Lorenz equations are

$$\begin{cases} \dot{\mathbf{x}} &= \sigma(\mathbf{y} - \mathbf{x}) \\ \dot{\mathbf{y}} &= \mathbf{x}(\rho - \mathbf{z}) - \mathbf{y} \\ \dot{\mathbf{z}} &= \mathbf{x}\mathbf{y} - \beta\mathbf{z} \end{cases} \tag{97}$$

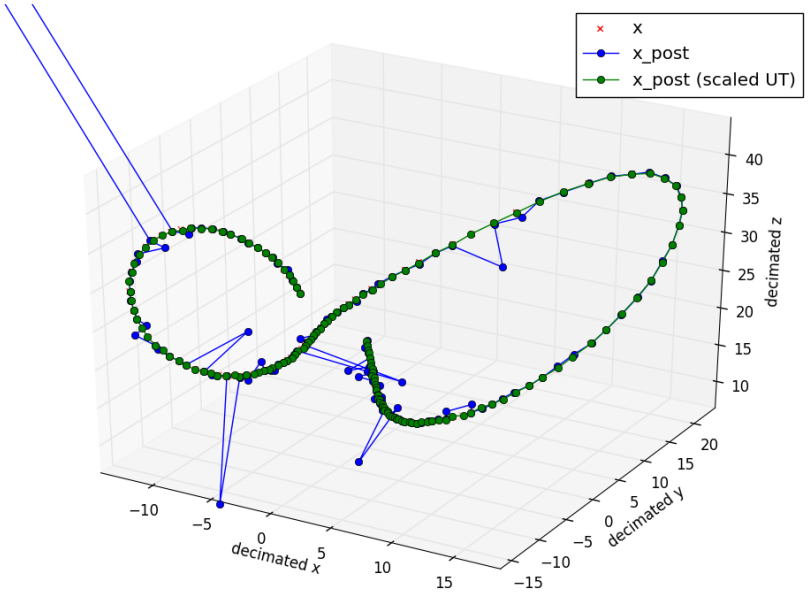
where $\rho = 28,\ \sigma = 10,\ \beta = 8/3$.



The sigma point selection scheme used in the unscented transformation has the property that as the dimension of the state-space increases, the radius of the sphere that bounds all the sigma points increases as well. Even though the mean and covariance of the prior distribution are still captured correctly, it does so at

the cost of sampling non-local effects. If the nonlinearities in question are very severe, this can lead to significant difficulties. In order to address this problem, the sigma points can be scaled towards or away from the mean of the prior distribution by a proper choice of $\varkappa$. The distance of the ith sigma point from $\bar{x}$, $|\mathcal{X}_i - \bar{\mathbf{x}}|$, is proportional to $\sqrt{n+\varkappa}$. When $\varkappa = 0$, the distance is proportional to $\sqrt{n}$. When $\varkappa > 0$ the points are scaled further from $\bar{\mathbf{x}}$ and when $\varkappa < 0$ the points are scaled towards $\bar{\mathbf{x}}$. For the special case of $\varkappa = 3 - n$, the desired dimensional scaling invariance is achieved by canceling the effect of n. However, when $\varkappa = 3 - n < 0$ the weight $W_0 < 0$ and the calculated covariance can be non-positive semidefinite. The scaled unscented transformation was developed to address this problem.

Jumping ahead, here is the comparison of Unscented Kalman filter with the scaled one.



We can see that the scaled UKF (green) is much more stable than the unscaled UKF (blue). Red is the truth.

Our augmented state is:

$$\mathbf{x}_a(k) = \begin{pmatrix} x_1 & x_2 & x_3 & \rho & \sigma & \beta & q_1 & q_2 & q_3 & v_1 & v_2 & v_3 \end{pmatrix}^T$$

The initial position is arbitrary chosen as

$$\mathbf{x}_a(0) = \begin{pmatrix} 0 & 0 & 0 & 28 & 10 & \frac{8}{3} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

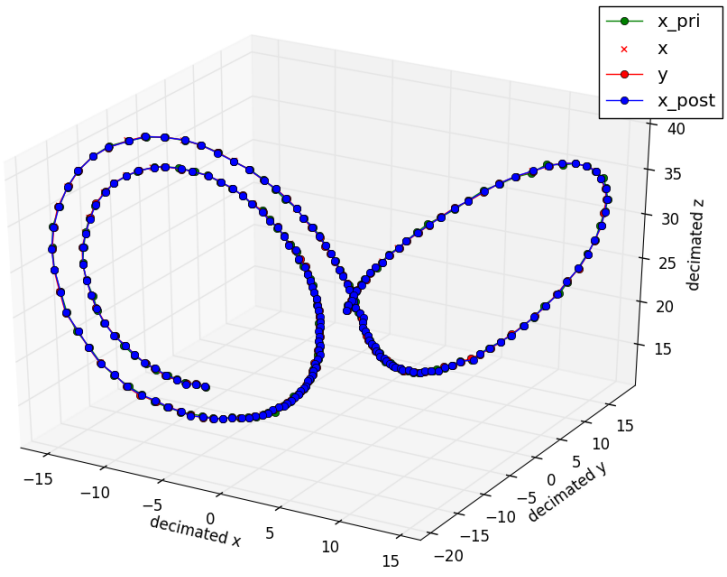
The initial covariance matrix is equal to

$$\mathbf{P}_a(0) = diag\begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.005 & 0.005 & 0.005 \end{pmatrix}$$

We add a Gaussian noise to the entire state and pass it through f (97) which makes it nonlinear.

We only observe the first 3 components.

For the scaled transformation $\alpha = 0.001,\ \beta = 2$.



As we can see it approximates the function pretty good.

3 Unscented Particle Filter

Particle filters or Sequential Monte Carlo methods are a set of genetic, Monte Carlo algorithms used to solve filtering problems arising in signal processing and Bayesian statistical inference. The particle filter methodology is used to solve Hidden Markov Chain (Model) (HMM) and nonlinear filtering problems.

Our state-space model is the same as in (65) and (67) with the exception that the noise does not have the Gaussian distribution and may even be unknown. The states follow a first order Markov process and the observations are assumed to be independent given the states. The posterior density $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, where $\mathbf{x}_{0:t} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\}$ and $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$, constitutes the complete solution to the sequential estimation problem. For such distribution the expectation is calculated as

$$E[g(\mathbf{X})] = I = \int_{\Omega} p(x)g(x)dx \tag{98}$$

One can use the Law of Large Numbers, which states that for a collection of independent identically distributed random variables $\{X_i\}_{i=1}^{\infty}$:

$$E[g(\mathbf{X})] = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{X}_i)$$

Therefore (98) can be approximated by N random variates $\{X_i\}_{i=1}^{N}$ (particles) with distribution $p(x)$ on Ω. The Monte-Carlo method has an accuracy which can be estimated as:

$$\begin{aligned} error &= \left| \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{X}_i) - I \right| \\ &= \left| \frac{\sigma_g}{\sqrt{N}} \left(\frac{\sum_{i=1}^{N} g(\mathbf{X}_i) - NI}{\sigma_g\sqrt{N}} \right) \right| \\ &\approx \left| \frac{\sigma_g}{\sqrt{N}}\eta(0,1) \right| \end{aligned} \tag{99}$$

where

$$\sigma_g^2 = \int_{\Omega} p(x)(g(x) - I)^2 dx \tag{100}$$

and $\eta(0,1)$ denotes a Gaussian random variable with mean 0 and variance 1. The last approximation was obtained by using the Central Limit Theorem, which states that for a sum of i.i.d. random variables Y_i with mean μ and finite variance σ^2:

$$\frac{\sum_{i=1}^{N} Y_i - N\mu}{\sigma\sqrt{N}} \to \eta(0,1) \; as \; N \to \infty$$

This shows that asymptotically the error converges at a rate $\mathcal{O}(1/\sqrt{N})$, independent of the dimensionality of the problem considered. Furthermore, the convergence rate in the Monte-Carlo method is strongly influenced by the prefactor σ_g which depends on the function $g(x)$ and the sampling distribution with density $p(x)$ that is used. The prefactor σ_g presents the primary avenue by which the convergence rate can be improved.

Transformations of random variables

For a random variables X and $Y = g(X)$ we have $P\{g(X) \in A\} = P\{X \in g^{-1}(A)\}$. For increasing functions the cumulative distribution of X

$$F_Y(y) = P\{Y \leq y\} = P\{g(X) \leq y\} = P\{X \leq g^{-1}(y)\} = F_X(g^{-1}(y))$$

Now use the chain rule to compute the density of Y

$$f_Y(y) = F_Y'(y) = \frac{d}{dy} F_X(g^{-1}(y)) = f_X(g^{-1}(y)) \frac{d}{dy} g^{-1}(y)$$

For g decreasing on the range of X

$$F_Y(y) = P\{Y \leq y\} = P\{g(X) \leq y\} = P\{X \geq g^{-1}(y)\} = 1 - F_X(g^{-1}(y))$$

and the density

$$f_Y(y) = F_Y'(y) = -\frac{d}{dy} F_X(g^{-1}(y)) = -f_X(g^{-1}(y)) \frac{d}{dy} g^{-1}(y)$$

For g decreasing, we also have g^{-1} decreasing and consequently the density of Y is indeed positive. We can combine these 2 cases to obtain the transformation formula for a monotonic function

$$f_Y(y) = f_X(g^{-1}(y)) \left| \frac{d}{dy} g^{-1}(y) \right| \tag{101}$$

Let X be a continuous random variable whose distribution function F_X is strictly increasing on the possible values of X. Then F_X has an inverse function. Let $U = F_X(X)$, then for $u \in [0,1]$

$$P\{U \leq u\} = P\{F_X(X) \leq u\} = P\{X \leq F_X^{-1}(u)\} = F_X\left(F_X^{-1}(u)\right) = u$$

In other words, U is a uniform random variable on $[0,1]$. Most random number generators simulate independent copies of this random variable. Consequently, we can simulate independent random variables having distribution

function F_X by simulating U, a uniform random variable on $[0,1]$, and then taking

$$X = F_X^{-1}(U)$$

And for the probability density function we've got

Theorem: Probability Density Transformation. Let $p_X(x)$ be the probability density function for a general n-dimensional random variable $X \in \mathbb{R}^n$. Then the random variable $Z = h(X)$ obtained from an invertible transformation $h : \mathbb{R}^n \to \mathbb{R}^n$ has the probability density

$$p_Z(z) = p_X\left(h^{-1}(z)\right)\left|\frac{dh^{-1}(z)}{dz}\right|$$

where the Jacobian of h^{-1} is defined as

$$\left|\frac{dh^{-1}(z)}{dz}\right| = det\begin{pmatrix} \frac{\partial h_1^{-1}}{\partial z_1} & \frac{\partial h_1^{-1}}{\partial z_2} & \cdots & \frac{\partial h_1^{-1}}{\partial z_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial h_n^{-1}}{\partial z_1} & \frac{\partial h_n^{-1}}{\partial z_2} & \cdots & \frac{\partial h_n^{-1}}{\partial z_n} \end{pmatrix}$$

Proof:

By definition of the random variable Z and invertibility of h we have:

$$P\{Z \in h(A)\} = P\{X \in A\}$$

From the definition of the probability density we have:

$$P\{X \in A\} = \int_A p_X(x)dx$$

$$P\{Z \in h(A)\} = \int_{h(A)} p_Z(z)dz$$

By the change of variable $z = h(x)$ we have:

$$P\{X \in A\} = \int_{h(A)} p_X(h^{-1}(z))\left|\frac{dh^{-1}(z)}{dz}\right|dz$$

This implies that for any set A we have:

$$\int_{h(A)} p_Z(z)dz = \int_{h(A)} p_X(h^{-1}(z))\left|\frac{dh^{-1}(z)}{dz}\right|dz$$

This requires that

$$p_Z(z) = p_X(h^{-1}(z))\left|\frac{dh^{-1}(z)}{dz}\right|$$

almost everywhere.

■

Sampling

For some probability densities $g(x)$ it may be difficult to determine analytically an appropriate transformation. In such cases the desired random variates can be obtained by generating candidate samples which are either accepted or rejected to obtain the desired distribution.

Importance sampling (to"cover" the function under the integral) is concerned with the choosing $p(x)$ for the random variates X_i so that regions which contribute significantly to the expectation of $g(X)$ are sampled with greater frequency. Thus regions where $f(x)$ is large should be sampled more frequently than those regions where $f(x)$ is comparatively very small. This is done in order to reduce the error (see (100) and (99)).

The most common strategy is to sample from the probabilistic model of the states evolution (transition prior). This strategy can, however, fail if the new measurements appear in the tail of the prior or if the likelihood is too peaked in comparison to the prior. This situation does indeed arise in several areas of engineering and finance where one can encounter sensors that are very accurate (peaked likelihoods) or data that undergoes sudden changes (non-stationarities). To overcome this problem, several techniques based on linearization have been proposed in the literature. For example, the EKF Gaussian approximation is used as the proposal distribution for a particle filter. Here we'll use the scaled Unscented Kalman filter to generate the importance proposal distribution. The UKF allows the particle filter to incorporate the latest observations into a prior updating routine. In addition, the UKF generates proposal distributions that match the true posterior more closely and also has the capability of generating heavier tailed distributions than the well known extended Kalman filter.

Since it is often impossible to sample directly from the posterior density function we can circumvent this difficulty by sampling from a known, easy-to-sample, proposal distribution $q(x_{0:t}|y_{1:t})$ and making use of Bayes' theorem and the following substitution

$$\begin{aligned} E[g_t(\mathbf{x}_{0:t})] &= \int g_t(\mathbf{x}_{0:t}) \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \int g_t(\mathbf{x}_{0:t}) \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \int g_t(\mathbf{x}_{0:t}) \frac{w_t(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \end{aligned}$$

where the variables $w_t(\mathbf{x}_{0:t})$ are known as the unnormalized importance weights (likelihood that the particle best represents the true state)

$$w_t(\mathbf{x}_{0:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \tag{102}$$

We can get rid of the unknown normalizing density $p(\mathbf{y}_{1:t})$ as follows

$$\begin{aligned} E[g_t(\mathbf{x}_{0:t})] &= \frac{1}{p(\mathbf{y}_{1:t})} \int g_t(\mathbf{x}_{0:t}) w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \frac{\int g_t(\mathbf{x}_{0:t}) w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t}) \frac{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} d\mathbf{x}_{0:t}} \\ &= \frac{\int g_t(\mathbf{x}_{0:t}) w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}}{\int w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}} \\ &= \frac{E_{q(\cdot|\mathbf{y}_{1:t})}[w_t(\mathbf{x}_{0:t}) g_t(\mathbf{x}_{0:t})]}{E_{q(\cdot|\mathbf{y}_{1:t})}[w_t(\mathbf{x}_{0:t})]} \end{aligned}$$

where the notation $E_{q(\cdot|\mathbf{y}_{1:t})}$ has been used to emphasize that the expectations are taken over the proposal distribution $q(\cdot|\mathbf{y}_{1:t})$. Hence, by drawing samples from the proposal function $q(\cdot|\mathbf{y}_{1:t})$, we can approximate the expectations of interest by the following estimate

$$\begin{aligned} \overline{E[g_t(\mathbf{x}_{0:t})]} &= \frac{1/N \sum_{i=1}^N g_t(\mathbf{x}_{0:t}^{(i)}) w_t(\mathbf{x}_{0:t}^{(i)})}{1/N \sum_{i=1}^N w_t(\mathbf{x}_{0:t}^{(i)})} \\ &= \sum_{i=1}^N g_t(\mathbf{x}_{0:t}^{(i)}) \tilde{w}_t(\mathbf{x}_{0:t}^{(i)}) \end{aligned} \tag{103}$$

where the normalized importance weights $\tilde{w}_t^{(i)}$ are given by

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^N w_t^{(j)}}$$

The estimate of equation (103) is biased as it involves a ratio of estimates. However, it is possible to obtain asymptotic convergence and a central limit theorem for $\overline{E[g_t(\mathbf{x}_{0:t})]}$ under the following assumptions:

1. $\mathbf{x}_{0:t}^{(i)}$ corresponds to a set of i.i.d. samples drawn from the proposal distribution, the support of the proposal distribution includes the support of the posterior distribution and $E[g_t(\mathbf{x}_{0:t})]$ exists and is finite.

2. The expectations of w_t and $w_t g_t^2(\mathbf{x}_{0:t})$ over the posterior distribution exist and are finite.

A sufficient condition to verify the second assumption is to have bounds on the variance of $g_t(\mathbf{x}_{0:t})$ and on the importance weights. Thus, as N tends to infinity, the posterior density function can be approximated arbitrarily well by the point-mass estimate

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^N \tilde{w}_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$$

Sequential Importance Sampling

If we assume that the states correspond to a Markov process and that the observations are conditionally independent given the states then

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) \tag{104}$$

$$p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0)\prod_{j=1}^{t} p(\mathbf{x}_j|\mathbf{x}_{j-1}) \; and \; p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j) \tag{105}$$

By substituting (104) and (105) into (102), a recursive estimate for the importance weights can be derived as follows

$$\begin{aligned} w_t &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &= w_{t-1}\frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})}\frac{1}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &= w_{t-1}\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \end{aligned} \tag{106}$$

Equation (106) provides a mechanism to sequentially update the importance weights, given an appropriate choice of proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$. The exact form of this distribution is a critical design issue and is usually approximated in order to facilitate easy sampling. Since we can sample from the proposal distribution and evaluate the likelihood and transition probabilities, all we need to do is generate a prior set of samples and iteratively compute the importance weights. This procedure, known as sequential importance sampling (SIS), allows us to obtain the type of estimates described by equation (103).

It was proven that the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$ minimizes the variance of the importance weights conditional on $\mathbf{x}_{0:t-1}$ and $\mathbf{y}_{1:t}$. Nonetheless, the distribution

$$q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) \cong p(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(the transition prior) is the most popular choice of proposal function. Although it results in higher Monte Carlo variation than the optimal proposal $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$, as a result of it not incorporating the most recent observations, it is usually easier to implement. The transition prior is defined in terms of the probabilistic model governing the states' evolution (65) and the process noise statistics. For example, if an additive Gaussian process noise model is used, the transition prior is simply,

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(f(\mathbf{x}_{t-1}, 0), Q_{t-1}\right)$$

But because this fails to use the latest available information to propose new values for the states, only a few particles will have significant importance weights when their likelihood are evaluated, and therefore we will not use it.

Resampling (Selection)

It was shown that the variance of importance weights (102) or (106) increases stochastically over time. This can be observed as one of the normalized importance weights tends to 1, while the remaining weights tend to 0. A large number of samples are thus effectively removed from the sample set because their importance weights become numerically insignificant. To avoid this degeneration or depletion of samples a selection (re-sampling) stage may be used to eliminate samples with low importance weights and multiply samples with high importance weights. It is possible to see an analogy to the steps in *genetic algorithms*.

A selection scheme associates to each particle $\mathbf{x}_{0:t}^{(i)}$ a number of"children", say $N_i \in \mathbb{N}$, such that $\sum_{i=1}^{N} N_i = N$. Several selection schemes have been proposed in the literature. These schemes satisfy $E[N_i] = N\tilde{w}_t^{(i)}$ but their performance varies in terms of the variance of the particles $var(N_i)$.

As we have already mentioned, resampling has the effect of removing particles with low weights and multiplying particles with high weights. However, this is at the cost of immediately introducing some additional variance. If particles have unnormalized weights with a small variance then the resampling step might be unnecessary. Consequently, in practice, it is more sensible to resample only when the variance of the unnormalized weights is superior to a pre-specified threshold. This is often assessed by looking at the variability of the weights using the so-called Effective Sample Size (ESS) criterion which is given by

$$ESS = \left(\sum_{i=1}^{N} \left(\tilde{w}_t^{(i)} \right)^2 \right)^{-1}$$

Its interpretation is that in a simple IS setting, inference based on the N weighted samples is approximately equivalent (in terms of estimator variance) to inference based on ESS perfect samples from the target distribution. The ESS takes values between 1 and N and we resample only when it is below a threshold N_T; typically $N_T = N/2$. Alternative criteria can be used such as the entropy of the weights $\tilde{w}_t^{(i)}$ which achieves its maximum value when $\tilde{w}_t^{(i)} = 1/N$. In this case, we resample when the entropy is below a given threshold.

Because of this sample degeneracy any Sequential MC algorithm which relies upon the distribution of full paths $x_{1:n}$ will fail for large enough n for any finite sample size N, in spite of the asymptotic justification. It is intuitive that one should endeavor to employ algorithms which do not depend upon the full path of the samples, but only upon the distribution of some finite component $x_{n-L:n}$ for some fixed L which is independent of n. Furthermore, ergodicity (a tendency

for the future to be essentially independent of the distant past) of the underlying system will prevent the accumulation of errors over time.

Although sample degeneracy emerges as a consequence of resampling, it is really a manifestation of a deeper problem – one which resampling actually mitigates. It is inherently impossible to accurately represent a distribution on a space of arbitrary high dimension with a sample of fixed, finite size. Sample impoverishment is a term which is often used to describe the situation in which very few different particles have significant weight. This problem has much the same effect as sample degeneracy and occurs, in the absence of resampling, as the inevitable consequence of multiplying together incremental importance weights from a large number of time steps. It is, of course, not possible to circumvent either problem by increasing the number of samples at every iteration to maintain a constant effective sample size as this would lead to an exponential growth in the number of samples required. This sheds some light on the resampling mechanism: it"resets the system" in such a way that its representation of final time marginals remains well behaved at the expense of further diminishing the quality of the path-samples. By focusing on the fixed-dimensional final time marginals in this way, it allows us to circumvent the problem of increasing dimensionality.

We will now present a number of selection or resampling schemes, namely: *sampling-importance resampling* (SIR), *residual resampling* and *minimum variance sampling*. We found that the specific choice of resampling scheme does not significantly affect the performance of the particle filter, so we used residual resampling in all of the experiments.

Sampling-importance resampling (SIR) and multinomial sampling

Resampling involves mapping the Dirac random measure $\left\{\mathbf{x}_{0:t}^{(i)}, \tilde{w}_t^{(i)}\right\}$ into an equally weighted random measure $\left\{\mathbf{x}_{0:t}^{(i)}, N^{-1}\right\}$. This can be accomplished by sampling uniformly from the discrete set $\left\{\mathbf{x}_{0:t}^{(i)}; i = 1, \ldots, N\right\}$ with probabilities $\left\{\tilde{w}_t^{(i)}; i = 1, \ldots, N\right\}$. We do this by constructing the cumulative distribution of the discrete set. Then a uniformly drawn sampling index i is projected onto the distribution range and then onto the distribution domain. The intersection with the domain constitutes the new sample index j. That is, the vector $\mathbf{x}_{0:t}^{(j)}$ is accepted as the new sample. Clearly, the vectors with the larger sampling weights will end up with more copies after the resampling process.

Sampling N times from the cumulative discrete distribution $\sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$ is equivalent to drawing $(N_i; i = 1, \ldots, N)$ from a multinomial distribution with parameters N and $\tilde{w}_t^{(i)}$. This procedure can be implemented in $\mathcal{O}(N)$ operations. As we are sampling from a multinomial distribution, the variance is $var(N_i) = N\tilde{w}_t^{(i)}(1 - \tilde{w}_t^{(i)})$.

Residual resampling

This procedure involves the following steps. Firstly, set $\tilde{N}_i = \left\lfloor N\tilde{w}_t^{(i)} \right\rfloor$. Secondly, perform an SIR procedure to select the remaining $\bar{N}_t = N - \sum_{i=1}^{N} \tilde{N}_i$ samples with new weights $w_t'^{(i)} = \bar{N}_t^{-1}(\tilde{w}_t^{(i)} N - \tilde{N}_i)$. Finally, add the results to the current $\tilde{N}_i$. For this scheme, the variance $var(N_i) = \bar{N}_t w_t'^{(i)}(1 - w_t'^{(i)})$ is smaller than the one given by the SIR scheme. Moreover, this procedure is computationally cheaper.

Minimum variance sampling

This strategy includes the stratified/systematic sampling procedures and the Tree Based Branching Algorithm. One samples a set of N points U in the interval $[0, 1]$, each of the points a distance N^{-1} apart. The number of children N_i is taken to be the number of points that lie between $\sum_{j=1}^{i-1} \tilde{w}_t^{(j)}$ and $\sum_{j=1}^{i} \tilde{w}_t^{(j)}$. This strategy introduces a variance on N_i even smaller than the residual resampling scheme, namely $var(N_i) = \bar{N}_t w_t'^{(i)}(1 - \bar{N}_t w_t'^{(i)})$. Its computational complexity is $\mathcal{O}(N)$.

MCMC Move Step

In the resampling stage, any particular sample with a high importance weight will be duplicated many times. As a result, the cloud of samples may eventually collapse to a single sample. This degeneracy will limit the ability of the algorithm to search for lower minima in other regions of the error surface. In other words, the number of samples used to describe the posterior density function will become too small and inadequate. A brute force strategy to overcome this problem is to increase the number of particles. A more refined strategy is to implement a Markov chain Monte Carlo (MCMC) step after the selection step.

After the selection scheme at time t, we obtain N particles distributed marginally approximately according to $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Since the selection step favors the creation of multiple copies of the"fittest" particles, it enables us to track time varying filtering distributions. However, many particles might end up having no children ($N_i = 0$), whereas others might end up having a large number of children, the extreme case being $N_i = N$ for a particular value i. In this case, there is a severe depletion of samples. We, therefore, require a procedure to introduce sample variety after the selection step without affecting the validity of the approximation.

A transition kernel for a Markov chain when X is discrete is simply a transition matrix $\mathcal{K}$ with elements

$$\mathcal{K} = \begin{pmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{pmatrix}$$

where $p_{i,j}$ is the probability of moving from the current state i to a new state j; $i, j \in X$. Now, rather than start with a specific state we consider a probability distribution over these states, $\mathbf{v}^{(0)} = \left(v_1^{(0)}, v_2^{(0)}, \ldots, v_n^{(0)}\right)^T$. If we randomly select the initial state from this distribution, then the probability distribution of the next state in the chain is given by

$$\mathbf{v}^{(1)} = \mathcal{K}\mathbf{v}^{(0)}$$

This idea can be extended, the probability distribution over the states after the second move is simply $\mathbf{v}^{(2)} = \mathcal{K}\mathbf{v}^{(1)} = \mathcal{K}^2\mathbf{v}^{(0)}$. This idea can be generalized; specifically, $\mathbf{v}^{(t)} = \mathcal{K}^t\mathbf{v}^{(0)}$. Of particular interest, is the distribution as the chain becomes long. As the chain's length increases then the distribution over the states becomes less and less determined by the starting distribution and more and more determined by the transition probabilities. Indeed, providing the chain satisfies certain regularity conditions, i.e. it does not get stuck in one state, there exists a unique invariant distribution associated with every transition matrix. Let π represent this invariant distribution. So for any starting distribution, $\pi^{(0)}$, as the chain becomes long then the $\pi^{(t)}$ tends to π ($\lim_{t\to\infty} \pi^{(t)} = \pi$).

There are two ways to calculate this invariant distribution. The first is analytical. This method exploits the fact that $\pi = \mathcal{K}\pi$, and solves this system of equations. The second is to simulate π by actually running the Markov chain. This involves choosing a starting value and simply running the Markov chain. The initial values in the chain depend strongly upon the starting values, hence they are usuallydiscarded. However, as the chain becomes longer then the elements of the chain represent random draws from the (invariant) probability distribution π. Another practical problem is the high auto-correlation between elements in the chain. This reduces the rate at which convergence is achieved. A practical solution is to sub-sample.

These ideas are readily extendable to continuous state space models, where the transition matrix is replaced by a transition kernel (a conditional probability density over the next state that depends only upon the current state) (notation: commonly written $P(x, y)$ instead of $p(y|x)$)

$$\mathcal{K}(x, x')P(X \in A|x) = \int_A \mathcal{K}(x, x')dx' = \int_A f(x'|x)dx'$$

where $f(x'|x)$ is a density function.

Most of Markov theory revolves around finding the invariant distribution of Markov chains. MCMC turns the problem around. Rather than finding the invariant distribution of a specific Markov chain, it starts with a specific invariant distributions and says, can I find a Markov chain that has this invariant distribution. (Each Markov process has a unique invariant distribution. Yet, many Markov chains could have the same invariant distribution. Thus, we are free to use any of these process to simulate the invariant distribution.) Typically, we already know the distribution of interest: the posterior distribution of

the parameters. The key is to find a transition kernel that has this invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$.

A strategy for solving the sample depletion problem involves introducing MCMC steps of invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ on each particle. The basic idea is that if the particles are distributed according to the posterior $p(\tilde{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t})$, then applying a Markov chain transition kernel $\mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t})$, with invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ such that $\int \mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t})p(\tilde{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, still results in a set of particles distributed according to the posterior of interest. However, the new particles might have been moved to more interesting areas of the state-space. In fact, by applying a Markov transition kernel, the total variation of the current distribution with respect to the invariant distribution can only decrease. Note that we can incorporate any of the standard MCMC methods, such as the Gibbs sampler and Metropolis Hastings algorithms, into the filtering framework, but we no longer require the kernel to be ergodic. The MCMC move step can also be interpreted as sampling from the finite mixture distribution $N^{-1}\sum_{i=1}^{N}\mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t}^{(i)})$.

One can generalize this idea by introducing MCMC steps on the product space with invariant distribution $\prod_{i=1}^{N} p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})$, that is to apply MCMC steps on the entire population of particles. It should be noted that independent MCMC steps spread out the particles in a particular mode more evenly, but do not explore modes devoid of particles, unless"clever" proposal distributions are available. By adopting MCMC steps on the whole population, we can draw upon many of the ideas developed in parallel MCMC computation. In this work, however, we limit ourselves to the simpler case of using independent MCMC transitions steps on each particle. In the case of standard particle filters, we propose to sample from the transition prior and accept according to a Metropolis-Hastings (MH) step as follows.

It is easy to construct a Markov kernel with a specified invariant distribution. For example, we could consider the following kernel, based upon the Gibbs sampler: set $x'_{1:n-L} = x_{1:n-L}$ then sample x'_{n-L+1} from $p(x_{n-L+1}|y_{1:n}, x'_{1:n-L}, x_{n-L+2:n})$, sample x'_{n-L+2} from $p(x_{n-L+2}|y_{1:n}, x'_{1:n-L+1}, x_{n-L+3:n})$ and so on until we sample x'_n from $p(x_n|y_{1:n}, x'_{1:n-1})$; that is

$$\mathcal{K}_n(x'_{1:n}|x_{1:n}) = \delta_{x_{1:n-L}}(x'_{1:n-L}) \prod_{k=n-L+1}^{n} p(x'_k|y_{1:n}, x'_{1:k-1}, x_{k+1:n})$$

and we write, with a slight abuse of notation, the non-degenerate component of the MCMC kernel $\mathcal{K}_n(x'_{n-L+1:n}|x_{1:n})$. It is straightforward to verify that this kernel is $p(x_{1:n}|y_{1:n})$-invariant.

If it is not possible to sample from $p(x'_k|y_{1:n}, x'_{1:k-1}, x_{k+1:n}) = p(x'_k|y_k, x'_{k-1}, x_{k+1})$, we can instead employ a Metropolis-Hastings (MH) strategy and sample a candidate according to some proposal $q(x'_k|y_k, x'_{k-1}, x_{k:k+1})$ and accept it with the usual MH acceptance probability

$$min\left(1, \frac{p(x'_{1:k}, x_{k+1:n}|y_{1:n})q(x_k|y_k, x'_{k-1}, x'_k, x_{k+1})}{p(x'_{1:k-1}, x_{k+1:n}|y_{1:n})q(x'_k|y_k, x'_{k-1}, x_{k:k+1})}\right)$$

$$= min\left(1, \frac{g(y_k|x'_k)f(x_{k+1}|x'_k)f(x'_k|x'_{k-1})q(x_k|y_k, x'_{k-1}, x'_k, x_{k+1})}{g(y_k|x_k)f(x_{k+1}|x_k)f(x_k|x'_{k-1})q(x'_k|y_k, x'_{k-1}, x_{k:k+1})}\right)$$

It is clear that these kernels can be ergodic only if $L = n$ and *all* of the components of $x_{1:n}$ are updated. However, in our context we will typically not use ergodic kernels as this would require sampling an increasing number of variables at each time step. In order to obtain truly online algorithms, we restrict ourselves to updating the variables $X_{n-L+1:n}$ for some fixed or bounded L.

To expand on MH update, it preserves any distribution π specified by an unnormalized density h with respect to a measure μ. There is no restriction on $h(x)$ other than that it actually be an unnormalized density (its normalizing constant is nonzero and finite) and that it can be evaluated, that is, for each x we can calculate $h(x)$. There is no requirement that we be able to do any integrals or know the value of the normalizing constant. In particular, unlike the Gibbs sampler, we do not need to know anything about any conditional distributions of π.

The Metropolis-Hastings update uses an auxiliary transition probability specified by a density $q(x, y)$ called the proposal distribution. For every point x in the state space, $q(x, \cdot)$ is a (normalized) probability density with respect to μ having 2 properties: for each x we can simulate a random variate y having the density $q(x, \cdot)$ and for each x and y we can evaluate the $q(x, y)$. To summarize, this is what we need

1. For each x we can evaluate $h(x)$.
2. For each x and y we can evaluate $q(x, y)$.
3. For each x we can simulate a random variate with density $q(x, \cdot)$ with respect to μ.

There is no necessary connection between the auxiliary density $q(x, y)$ and the density $h(x)$ of the stationary distribution. We can choose any density that we know how to simulate. For example, if the state space is d-dimensional Euclidean space $\mathbb{R}^d$ we could use a multivariate normal proposal density with mean x and variance a constant times the identity. If ϕ denotes a $Normal(0, \sigma^2 I)$ density, then we have $q(x, y) = \phi(y - x)$. We can easily simulate multivariate normal variates and evaluate the density.

The Metropolis-Hastings update then works as follows. The current position is x, and the update changes x to its value at the next iteration.

1. Simulate a random variate y having the density $q(x, \cdot)$.

2. Calculate the "Hastings ratio"

$$R = \frac{h(y)q(y,x)}{h(x)q(x,y)} \tag{107}$$

3. Do "Metropolis rejection": with probability $min(1, R)$ set $x = y$ (otherwise x remains the same).

Note also that the denominator of the Hastings ratio (107) can never be zero if the chain starts at a point where $h(x)$ is nonzero. A proposal y such that $q(x, y) = 0$ occurs with probability zero, and a proposal y such that $h(y) = 0$ is accepted with probability zero. Thus there is probability zero that denominator of the Hastings ratio is ever zero during an entire run of the Markov chain so long as $h(X_1) > 0$. If we do not start in the support of the stationary distribution we have the problem of defining how the chain should behave when $h(x) = h(y) = 0$, that is, how the chain should move when both the current position and the proposal are outside the support of the stationary distribution. The Metropolis-Hastings algorithm says nothing about this. It is a problem that is best avoided by starting at a point where $h(x)$ is positive.

Also note specifically that there is no problem if the proposal is outside the support of the stationary distribution. If $h(y) = 0$, then $R = 0$ and the proposal is always rejected, but this causes no difficulties.

The special case when we use a proposal density satisfying $q(x, y) = q(y, x)$ is called the Metropolis update. In this case the Hastings ratio (107) reduces to the odds ratio

$$R = \frac{h(y)}{h(x)}$$

and there is no need to be able to evaluate $q(x, y)$ only to be able to simulate it.

For our example that uses the Normal distribution as a proposed one, if we choose σ too small, the chain can't get anywhere in any reasonable number of iterations. If σ is chosen ridiculously large, all of the proposals will be so far out in the tail that none will be accepted in any reasonable number of iterations. A rule of thumb is to choose σ such that about 20% of proposals are accepted but this may fail sometimes.

When the state X is a vector $X = (X_1, \ldots, X_d)$, the Metropolis-Hastings update can be done one variable at a time, just like the Gibbs update.

Smoothing MH step

1. sample $v \sim U_{[0,1]}$

2. sample the proposal candidate $x_t^{*(i)} \sim p(x_t | x_{t-1}^{(i)})$

3. if $v \leq min\left\{1, \frac{p(\mathbf{y}_t|\mathbf{x}_t^{*(i)})}{p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^{(i)})}\right\}$

$$\rightarrow then\ accept\ move: \qquad \mathbf{x}_{0:t}^{(i)} = \left(\tilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{x}_t^{*(i)}\right)$$

$$\rightarrow else\ reject\ move: \qquad \mathbf{x}_{0:t}^{(i)} = \tilde{\mathbf{x}}_{0:t}^{(i)}$$

Algorithm

1. initialization: $t = 0$

for $i = 1, \ldots, N$ draw the states (particles) $\mathbf{x}_0^{(i)}$ from the prior $p(\mathbf{x}_0)$ and set

$$\bar{\mathbf{x}}_0^{(i)} = E[\mathbf{x}_0^{(i)}]$$
$$\mathbf{P}_0^{(i)} = E\left[\left(\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0^{(i)}\right)\left(\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0^{(i)}\right)^T\right]$$
$$\bar{\mathbf{x}}_0^{(i)a} = E[\mathbf{x}_0^{(i)a}] = \left[(\bar{\mathbf{x}}_0^{(i)})^T\ \mathbf{0}\ \mathbf{0}\right]^T$$

$$\mathbf{P}_0^{(i)a} = E\left[\left(\mathbf{x}_0^{(i)a} - \bar{\mathbf{x}}_0^{(i)a}\right)\left(\mathbf{x}_0^{(i)a} - \bar{\mathbf{x}}_0^{(i)a}\right)^T\right] = \begin{pmatrix} \mathbf{P}_0^{(i)} & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{R} \end{pmatrix}$$

2. for $t = 1, 2, \ldots$

2.1. importance sampling step

(a) for $i = 1, \ldots, N$:

— update the particles with the UKF:

* calculate sigma points:

$$\mathcal{X}_{t-1}^{(i)a} = \left(\bar{\mathbf{x}}_{t-1}^{(i)a},\ \bar{\mathbf{x}}_{t-1}^{(i)a} \pm \sqrt{(n_a + \lambda)\mathbf{P}_{t-1}^{(i)a}}\right)$$

* propagate particle into future (time update):

$$\mathcal{X}_{t|t-1}^{(i)x} = f\left(\mathcal{X}_{t-1}^{(i)x},\ \mathcal{X}_{t-1}^{(i)v}\right) \qquad \bar{\mathbf{x}}_{t|t-1}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(m)} \mathcal{X}_{j,t|t-1}^{(i)x}$$

$$\mathbf{P}_{t|t-1}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(c)} \left(\mathcal{X}_{j,t|t-1}^{(i)x} - \bar{\mathbf{x}}_{t|t-1}^{(i)}\right)\left(\mathcal{X}_{j,t|t-1}^{(i)x} - \bar{\mathbf{x}}_{t|t-1}^{(i)}\right)^T$$

$$\mathcal{Y}_{t|t-1}^{(i)} = h\left(\mathcal{X}_{t|t-1}^{(i)x},\ \mathcal{X}_{t-1}^{(i)n}\right) \qquad \bar{\mathbf{y}}_{t|t-1}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(m)} \mathcal{Y}_{j,t|t-1}^{(i)}$$

* incorporate new observation (measurement update):

$$\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} = \sum_{j=0}^{2n_a} W_j^{(c)} \left(\mathcal{Y}_{j,t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right)\left(\mathcal{Y}_{j,t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right)^T$$

$$\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} = \sum_{j=0}^{2n_a} W_j^{(c)} \left(\mathcal{X}_{j,t|t-1}^{(i)x} - \bar{\mathbf{x}}_{t|t-1}^{(i)}\right)\left(\mathcal{Y}_{j,t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right)^T$$

$$\mathbf{K}_t = \mathbf{P}_{\mathbf{x}_t\mathbf{y}_t}\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}^{-1}$$

$$\bar{\mathbf{x}}_t^{(i)} = \bar{\mathbf{x}}_{t|t-1}^{(i)} + \mathbf{K}_t\left(\mathbf{y}_t - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right) \qquad \hat{\mathbf{P}}_t^{(i)} = \mathbf{P}_{t|t-1}^{(i)} - \mathbf{K}_t\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}\mathbf{K}_t^T$$

— sample $\hat{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}) = \mathcal{N}(\bar{\mathbf{x}}_t^{(i)}, \hat{\mathbf{P}}_t^{(i)})$
— set $\hat{\mathbf{x}}_{0:t}^{(i)} \equiv (\mathbf{x}_{0:t-1}^{(i)}, \hat{\mathbf{x}}_t^{(i)})$ and$\hat{\mathbf{P}}_{0:t}^{(i)} \equiv (\mathbf{P}_{0:t-1}^{(i)}, \hat{\mathbf{P}}_t^{(i)})$

(b) for $i = 1, \ldots, N$: evaluate the importance weights up to a normalizing constant

$$w_t^{(i)} \propto \frac{p(\mathbf{y}_t|\hat{\mathbf{x}}_t^{(i)})p(\hat{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\hat{\mathbf{x}}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}$$

(c) for $i = 1, \ldots, N$: normalize the importance weights

$$\tilde{w}_t^{(i)} = w_t^{(i)}\left[\sum_{j=1}^{N} w_t^{(j)}\right]^{-1}$$

2.2. selection step (resampling)

Multiply/suppress particles/samples $\left(\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{\mathbf{P}}_{0:t}^{(i)}\right)$ with high/low importance weights $\tilde{w}_t^{(i)}$, respectively, to obtain N random particles $\left(\tilde{\mathbf{x}}_{0:t}^{(i)}, \tilde{\mathbf{P}}_{0:t}^{(i)}\right)$ (approximately distributed according to $p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})$).

2.3. MCMC step (optional)

Apply a Markov transition kernel with invariant distribution $p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})$ to obtain $\left(\mathbf{x}_{0:t}^{(i)}, \mathbf{P}_{0:t}^{(i)}\right)$.

2.4. output (expectation)

The output of the algorithm is a set of samples that can be used to approximate the posterior distribution as follows

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$$

One obtains straightforwardly the following estimate of $E[g_t(\mathbf{x}_{0:t})]$

$$E[g_t(\mathbf{x}_{0:t})] = \int g_t(\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \approx \frac{1}{N} \sum_{i=1}^{N} g_t(\mathbf{x}_{0:t}^{(i)})$$

for some function of interest $g_t : (\mathbb{R}^{n_x})^{(t+1)} \to \mathbb{R}^{n_{g_t}}$ integrable with respect to $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$.

Examples of appropriate functions include the marginal conditional mean of $\mathbf{x}_{0:t}$, in which case $g_t(\mathbf{x}_{0:t}) = \mathbf{x}_t$,

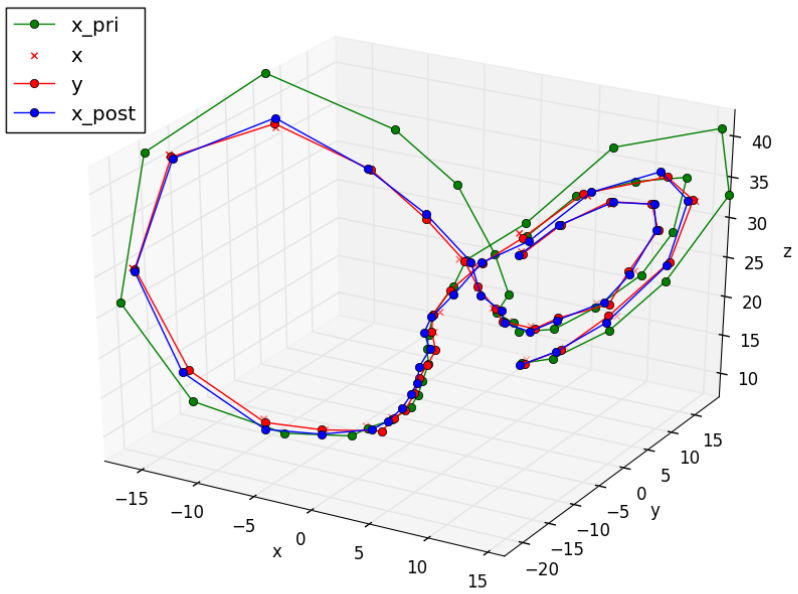
or mode,

or the marginal conditional covariance of $\mathbf{x}_{0:t}$ with $g_t(\mathbf{x}_{0:t}) = \mathbf{x}_t \mathbf{x}_t' - E_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\mathbf{x}_t] E'_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\mathbf{x}_t]$.

The marginal conditional mean is often the quantity of interest, because it is the optimal MMSE estimate of the current state of the system.

3.1 Lorenz Attractor Example

I use the same ρ, σ *and* β as in the section 2.2. The number of particles used is 200. With the time step 50 times bigger than in UKF, and noises are 10 times bigger a piece of the function looks like



3.2 Bond Mid-Price Estimation

The Kalman filter assumes a linear system with a Gaussian noise. When the system is inherently nonlinear, the first solution is to linearize it through the Taylor expansion as in the Extended Kalman filter. The next improvement is the Unscented Kalman filter (a.k.a. sigma points filter) which uses a clever approach of estimating a probability distribution rather than the function. This achieves a better accuracy. The particle filter removes the assumption of a Gaussian noise; it may even be of unknown distribution. It comes at the computational expense, though.

For bond pricing it presents additional advantages:

- It is well known that prices do not exhibit the normal distribution (fat tails).
- It is not easy to incorporate information about "Traded Away" transactions reported in RFQs (Request For Quote) in Kalman-like filters.

In this model the distance to mid (DTM) is assumed symmetrical and does not depend on the side for simplicity. Although this skewness, e.g. due to a holding position, can be added later.

Our state $\mathbf{x}$ will include d bonds. Instead of a price we'll consider the yield to benchmark (YtB) which is customary for investment grade (IG) bonds. We assume that it follows the Weiner process without a drift

$$d\mathbf{x}_t = \sigma d\mathbf{W}_t \tag{108}$$

where $\mathbf{W}_t$ is a d-dimensional Brownian motion with

$$d\left\langle W_t^i, W_t^j \right\rangle = \rho^{i,j} dt$$

We denote by Σ the covariance matrix for process noise:

$$\Sigma^{ij} = \rho^{i,j}\sigma^i\sigma^j$$

Another improvement to this model would be to consider an Ornstein–Uhlenbeck process $d\mathbf{x}_t = \theta(\mu - \mathbf{x}_t)dt + \sigma d\mathbf{W}_t$, where $\theta > 0$, $\sigma > 0$ (this is also mentioned as the Vasicek model).

Let's introduce another process $\mathbf{z}_t$ following a d-dimensional Ornstein–Uhlenbeck process:

$$d\mathbf{z}_t = -A\mathbf{z}_t dt + V d\mathbf{B}_t \tag{109}$$

where $\mathbf{z}_0$ is given, A and V are $d \times d$ matrices, and $\mathbf{B}_t$ a d-dimensional standard Brownian motion assumed to be independent from the process $\mathbf{W}_t$.

Meucci showed in [33] that

$$\mathbb{E}[\mathbf{z}_{t+\tau}|\mathbf{z}_t] = e^{-A\tau}\mathbf{z}_t$$

and

$$\mathbb{V}[\mathbf{z}_{t+\tau}|\mathbf{z}_t] = \Gamma(\tau)$$

$(\tau > 0)$

$$vec\left(\Gamma(\tau)\right) = \left(A \otimes I_d + I_d \otimes A\right)^{-1} \left(I_{d^2} - exp\left[-\left(A \otimes I_d + I_d \otimes A\right)\tau\right]\right) vec(VV^T)$$

where $vec(\cdot)$ refers to the vectorization operator, i.e.

$$vec\left((M_{i,j})_{1\leq i,j\leq d}\right) = (M_{1,1}, \ldots, M_{d,1}, \ldots, M_{1,d}, \ldots, M_{d,d})^T$$

In practice, we consider A to be a diagonal matrix. Then we have

$$\Gamma_{ij}(\tau) = \frac{1}{a^i + a^j}\left(1 - e^{-(a^i+a^j)\tau}\right)(VV^T)_{ij}$$

Let's denote the DTM (half bid-ask spread) of asset i as ψ_t^i and define the stochastic process as

$$\psi_t^i = \psi_0^i exp(z_t^i), \qquad \psi_0^i \, given$$

In other words, $x_t^i + \psi_t^i$ and $x_t^i - \psi_t^i$ are the bid-YtB and the ask-YtB, respectively (the bid-YtB has to be higher than the ask-YtB for the bid price to be lower than the ask price). That is, our state for each particle at time t is $\{\mathbf{x}, \psi\}$, vectors $\mathbf{x}$ and ψ of dimension d.

From the viewpoint of a given dealer D the information available to her corresponds to 5 different situations:

		D's observation at time t
1- D2C	A client buys bond i from D at time t.	$y_t^i = x_t^i - \psi_t^i + \epsilon_t^i$
2- D2C	A client sells bond i to D at time t.	$y_t^i = x_t^i + \psi_t^i + \epsilon_t^i$
3- Traded Away	A client buys bond i from another dealer at time t.We assume D has proposed $\check{y}$ but was not chosen because a better price was found elsewhere.	$y_t^i = x_t^i - \psi_t^i + \epsilon_t^i$ for D observation is $1_{y\geq\check{y}}$
4- Traded Away	A client sells bond i to another dealer at time t.We assume D has proposed $\check{y}$ but was not chosen because a better price was found elsewhere.	$y_t^i = x_t^i + \psi_t^i + \epsilon_t^i$ for D observation is $1_{y\leq\check{y}}$
5-D2D	Another dealer transacted bond i with D on the inter-dealer broker (IDB) market.	$y_t^i \in [x_t^i - a_t^i + \epsilon_t^i,\, x_t^i + a_t^i + \epsilon_t^i]$ where a_t^i can for instance be chosen $\sim$to ψ_t^i or $\sim$to a typical size for the bid-ask spread, e.g. of CBBT

where $\epsilon_t^i \sim \mathcal{N}\left(0,\, \sigma_\epsilon^{i^2}\right)$ assumed to be independent of all other random variables (observation noise).

It should be noted that for scenarios 3-5 when calculating the normal probability for likelihood the cumulative distribution function Φ should be used, namely

3	$\Phi\left(-(\check{y}-x+\psi)/R\right)$
4	$\Phi\left((\check{y}-x-\psi)/R\right)$
5	$\Phi\left((y-x+a)/R\right)-\Phi\left((y-x-a)/R\right)$

,

and when sampling for $\hat{\mathbf{x}}$ in the importance sampling step and for $\mathbf{x}'$ in the MCMC step, the truncated Gaussian distribution should be used (right-sided, left-sided and two-sided, respectively).

Because we receive an observation for 1 bond at a time, e.g. j^{th}, we apply the filter treatment to this component of $\mathbf{x}$. The rest are updated in accordance with their correlations. For each particle

$$x_{t+1}^{i\neq j}=\begin{pmatrix} x_{t+1}^1 \\ \vdots \\ x_{t+1}^{j-1} \\ x_{t+1}^{j+1} \\ \vdots \\ x_{t+1}^d \end{pmatrix}+\left(x_{t+1}^j-x_t^j\right)\begin{pmatrix} \rho^{j,1}\frac{\sigma^1}{\sigma^j} \\ \vdots \\ \rho^{j,j-1}\frac{\sigma^{j-1}}{\sigma^j} \\ \rho^{j,j+1}\frac{\sigma^{j+1}}{\sigma^j} \\ \vdots \\ \rho^{j,d}\frac{\sigma^d}{\sigma^j} \end{pmatrix}$$

and

$$\Sigma^{i\neq j}=\begin{pmatrix} {\sigma^1}^2 & \cdots & \rho^{1,j-1}\sigma^1\sigma^{j-1} & \rho^{1,j+1}\sigma^1\sigma^{j+1} & \cdots & \rho^{1,d}\sigma^1\sigma^d \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho^{j-1,1}\sigma^{j-1}\sigma^1 & \cdots & \rho^{j-1,j-1}\sigma^{j-1}\sigma^{j-1} & \rho^{j-1,j+1}\sigma^{j-1}\sigma^{j+1} & \cdots & \rho^{j-1,d}\sigma^{j-1}\sigma^d \\ \rho^{j+1,1}\sigma^{j+1}\sigma^1 & \cdots & \rho^{j+1,j-1}\sigma^{j+1}\sigma^{j-1} & \rho^{j+1,j+1}\sigma^{j+1}\sigma^{j+1} & \cdots & \rho^{j+1,d}\sigma^{j+1}\sigma^d \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho^{d,1}\sigma^d\sigma^1 & \cdots & \rho^{d,j-1}\sigma^d\sigma^{j-1} & \rho^{d,j+1}\sigma^d\sigma^{j+1} & \cdots & \rho^{d,d}\sigma^d\sigma^d \end{pmatrix}-$$

$$-\begin{pmatrix} \rho^{j,1}\sigma^1 \\ \vdots \\ \rho^{j,j-1}\sigma^{j-1} \\ \rho^{j,j+1}\sigma^{j+1} \\ \vdots \\ \rho^{j,d}\sigma^d \end{pmatrix}\left(\rho^{j,1}\sigma^1,\ldots,\rho^{j,j-1}\sigma^{j-1},\rho^{j,j+1}\sigma^{j+1},\ldots,\rho^{j,d}\sigma^d\right).$$

It is noteworthy that between 2 observations times, one could continue to diffuse particles by using the dynamics given by (108) and (109) to obtain an empirical estimation of the distribution of mid-YtBs and half bid-ask spreads.

The parameters Σ, A, V *and* ψ_0 can be either estimated using a fully Bayesian approach or calibrated off-line on historical data.

For the covariance matrix Σ one can assume that the correlation structure and the volatility levels of the YtBs associated with the CBBT mid-prices are the same as those of our mid-YtBs. Therefore, it is reasonable to estimate Σ on CBBT mid-price data.

Simulation

Synthetic data was used for this example. It has the correlated hyperbolic secant distribution which has heavier tails than Gaussian, and an easy analytical formula for the inverse cdf. To generate correlated distributions the NORTA (Normal-to-anything) algorithm can be used. The goal is to generate X with cdf F and $corr(x) = \Sigma_X$

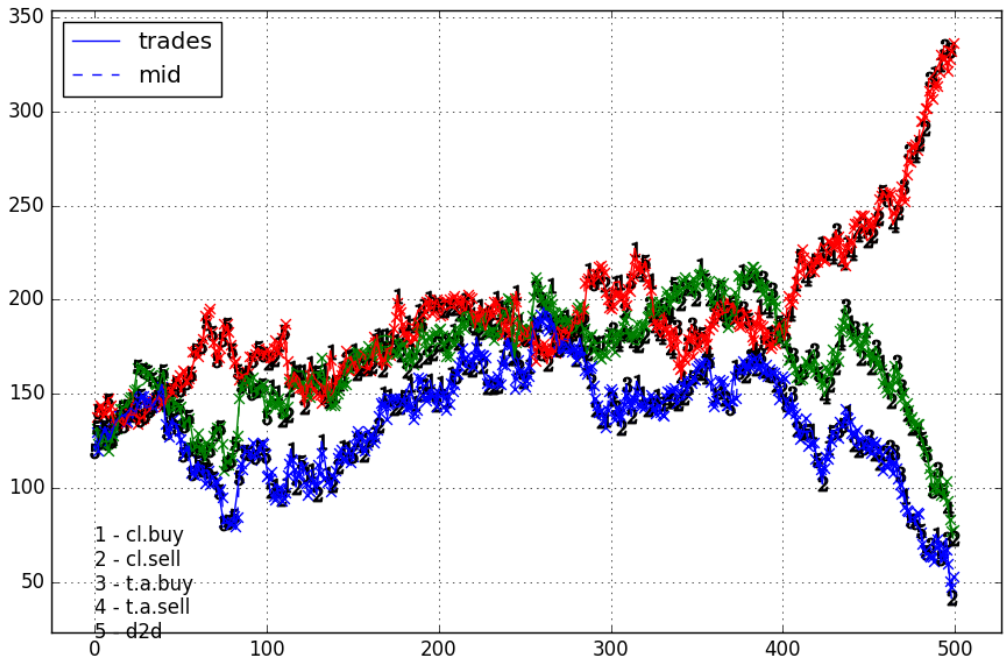
$$X = \left(F_{x_1}^{-1}\left[\Phi(z_1)\right],\, F_{x_2}^{-1}\left[\Phi(z_2)\right], \ldots\right)^T$$

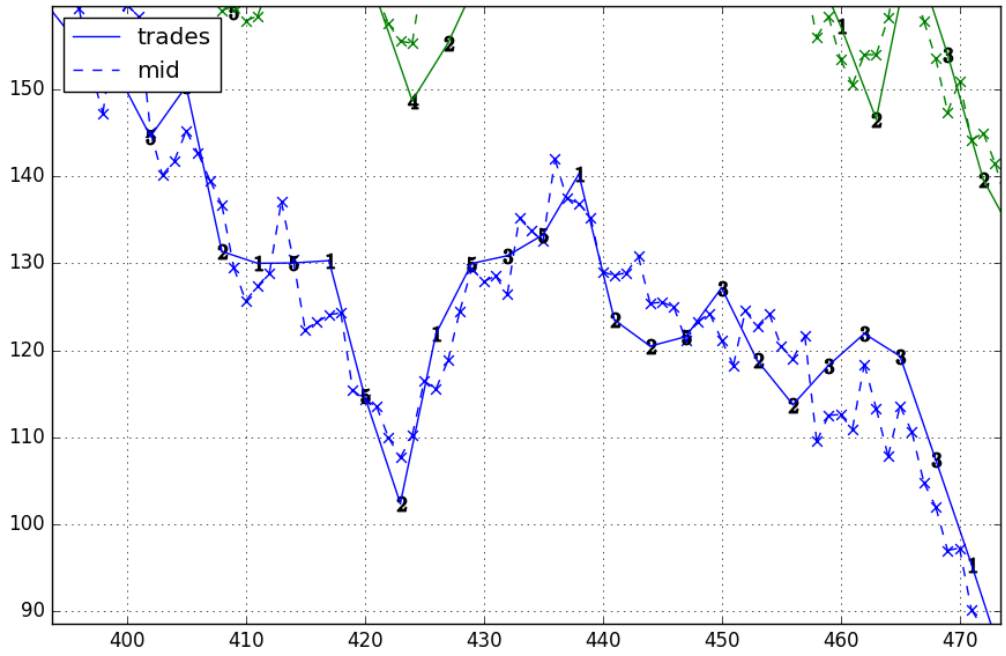
where Φ is the univariate standard normal cdf, $Z = (z_1,\, z_2, \ldots)^T$ is standard multivariate normal vector with correlation matrix Σ_Z.

Therefore, in the NORTA transformation process, a multivariate normal vector **Z** is transformed into a multivariate uniform vector **U** and then the multivariate uniform vector **U** into the desired vector **X**. So, the joint distribution of **U** is known as a **copula** and any joint distribution has a representation as a transformation of a copula.

Here is an example for 3 simulated bonds with the correlation matrix

$$\Sigma = \begin{pmatrix} 1.0 & 0.8 & -0.48 \\ 0.8 & 1.0 & -0.6 \\ -0.48 & -0.6 & 1.0 \end{pmatrix}$$





Using the calibration method described below the following parameters were obtained:

$$\Sigma = \begin{pmatrix} 4.689318 & 5.14435 & -2.589705 \\ 5.14435 & 12.219445 & -2.516943 \\ -2.589705 & 2.516943 & 2.501255 \end{pmatrix}$$

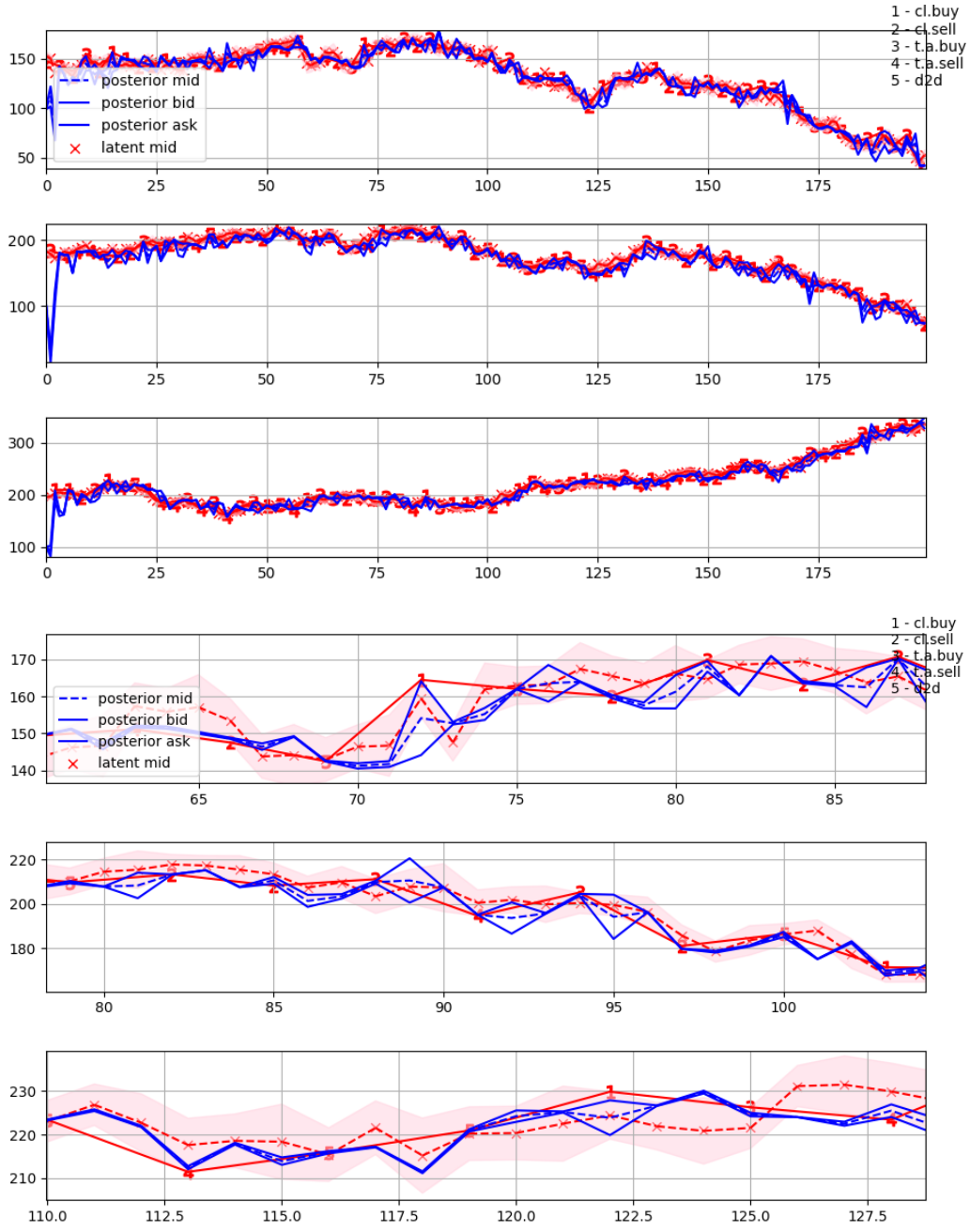
$$A = \begin{pmatrix} 1.11 & 0 & 0 \\ 0 & 1.66 & 0 \\ 0 & 0 & 1.63 \end{pmatrix}$$

$$\psi_0 = \begin{pmatrix} 0.51 & 0.54 & 0.41 \end{pmatrix}^T$$

$$V = \begin{pmatrix} 1.25 & 0 & 0 \\ 0 & 1.80 & 0 \\ 0 & 0 & 2.01 \end{pmatrix}$$

$$\Gamma = \begin{pmatrix} 0.627387 & 0 & 0 \\ 0 & 0.940622 & 0 \\ 0 & 0 & 1.19172 \end{pmatrix}$$

The results:


1 - cl.buy
2 - cl.sell
3 - t.a.buy
4 - t.a.sell
5 - d2d
posterior mid
posterior bid
posterior ask
latent mid

3.3 Analytic solution to Ornstein-Uhlenbeck SDE

The Ornstein-Uhlenbeck process:

$$dS(t) = \lambda(\mu - S(t))dt + \sigma dW(t) \tag{110}$$

where $W(t)$ is a standard Brownian motion, and $\lambda > 0$, μ, $and\, \sigma > 0$ are constants.

Motivated by the observation that μ is supposed to be the long-term mean of the process $S(t)$, we can simplify the SDE (110) by introducing the change of variable

$$y = S - \mu$$

that subtracts off the mean. Then $y(t)$ satisfies the SDE:

$$dy = dS = -\lambda y dt + \sigma dW \tag{111}$$

In SDE (111), the process $y(t)$ is seen to have a drift towards the value zero, at an exponential rate λ ($dy \sim \lambda y$). This motivates the change of variables

$$y = e^{-\lambda t} z \quad \Leftrightarrow \quad z = e^{\lambda t}$$

which should remove the drift. A calculation with the product rule for Itô integrals shows that this is so:

$$\begin{aligned} dz &= \lambda e^{\lambda t} y dt + e^{\lambda t} dy \\ &= \lambda e^{\lambda t} y dt + e^{\lambda t}(-\lambda y dt + \sigma dW) \\ &= \sigma e^{\lambda t} dW \end{aligned}$$

The solution for $z(t)$ is immediately obtained by Itô-integrating both sides from t_0 to t:

$$z(t) = z(t_0) + \sigma \int_{t_0}^{t} e^{\lambda \tau} dW(\tau)$$

Reversing the changes of variables, we have:

$$e^{\lambda t} y = e^{\lambda t_0} y(t_0) + \sigma \int_{t_0}^{t} e^{\lambda \tau} dW(\tau)$$

$$y = e^{-\lambda(t-t_0)} y(t_0) + \sigma e^{-\lambda t} \int_{t_0}^{t} e^{\lambda \tau} dW(\tau)$$

$$S(t) = \mu + e^{-\lambda(t-t_0)}\left(S(t_0) - \mu\right) + \sigma \int_{t_0}^{t} e^{-\lambda(t-\tau)} dW(\tau) \tag{112}$$

For any fixed t_0 and t, the random variable $S(t)$, conditional upon $S(t_0)$, is normally distributed with

$$mean = \mu + e^{-\lambda(t-t_0)}\left(S(t_0) - \mu\right), \quad variance = \frac{\sigma^2}{2\lambda}\left(1 - e^{-2\lambda(t-t_0)}\right).$$

The Ornstein-Uhlenbeck process is a time-homogeneous Itô diffusion.

3.3.1 Calibration of O.U.

The following simulation equation can be used to generate paths:

$$S_{t+1} = \mu\left(1 - e^{-\lambda\triangle t}\right) + e^{-\lambda\triangle t} S_t + \sigma\sqrt{\frac{1 - e^{-2\lambda\triangle t}}{2\lambda}}\mathcal{N}(0,1) \tag{113}$$

Linear regression: $S_{t+1} = aS_t + b + \epsilon$, where

$$\begin{aligned} a &= e^{-\lambda\triangle t} \\ b &= \mu\left(1 - e^{-\lambda\triangle t}\right) \\ sd(\epsilon) &= \sigma\sqrt{\frac{1 - e^{-2\lambda\triangle t}}{2\lambda}} \end{aligned}$$

Rewriting these equations gives

$$\begin{aligned} \lambda &= -\frac{\ln a}{\triangle t} \\ \mu &= \frac{b}{1-a} \\ \sigma &= sd(\epsilon)\sqrt{\frac{-2\ln a}{(1-a^2)\triangle t}} \end{aligned}$$

Note: when $\lambda = 0$ using L'Hôpital's rule the last term in (113) simplifies to $\sigma\sqrt{\triangle t}\mathcal{N}(0,1)$.

4 Optimal Control

Let's consider a system described by

$$\dot{x} = f(t, x, u), \quad x(t_0) = x_0 \tag{114}$$

with the cost functional for each possible behavior

$$J(u(t)) := \int_{t_0}^{t_f} L(t, x(t), u(t))\, dt + K(t_f, x_f) \tag{115}$$

where L and K are running and terminal cost, t_f is the final time.

Note: the variational problem is a particular case of the optimal control with $u = \frac{dx}{dt}$.

Weierstrass Theorem:*If f is a continuous function and D is a compact set, then there exists a global minimum of f over D.*

For subsets of $\mathbb{R}^n$, compactness can be defined in three equivalent ways:

1. D is compact if it is closed and bounded.
2. D is compact if every open cover of D has a finite subcover.
3. D is compact if every sequence in D has a subsequence converging to some point in D (sequential compactness).

If D is a convex set and f is a convex function, then first, a local minimum is automatically a global one; second, the first-order necessary condition (for $f \in \mathcal{C}^1$) is also a sufficient condition.

Now suppose that D is a surface in $\mathbb{R}^n$ defined by the equality constraints

$$h_1(x) = h_2(x) = \ldots = h_m(x) = 0 \tag{116}$$

where h_i are $\mathcal{C}^1$ functions from $\mathbb{R}^n$ to $\mathbb{R}$. Then the gradient of f at x^* is a linear combination of the gradients of the constraint functions $h_1, \ldots, h_m$ at x^*:

$$\nabla f(x^*) \in span\{\nabla h_i(x^*),\, i = 1, \ldots, m\}$$

Indeed, if the claim were not true, then $\nabla f(x^*)$ would have a component orthogonal to $span\{\nabla h_i(x^*)\}$.

This means that there exist real numbers $\lambda_1^*, \ldots, \lambda_m^*$ such that

$$\nabla f(x^*) + \lambda_1^* \nabla h_1(x^*) + \ldots + \lambda_m^* \nabla h_m(x^*) = 0 \tag{117}$$

This is the first-order necessary condition for constraint optimality. The coefficients λ_i^* are called **Lagrange multipliers**.

Let's introduce the augmented cost function

$$\ell(x, \lambda) := f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) \tag{118}$$

Thus, adding Lagrange multipliers, loosely speaking, converts a constrained problem into an unconstrained one

$$\nabla \ell(x^*, \lambda^*) = \begin{pmatrix} \ell_x(x^*, \lambda^*) \\ \ell_\lambda(x^*, \lambda^*) \end{pmatrix} = \begin{pmatrix} \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) \\ h(x^*) \end{pmatrix} = 0 \quad (119)$$

This is a necessary condition for x^* to be a constraint extremum of f but not sufficient.

For a metric we'll use either the 0-norm

$$\|y\|_0 = \max_{a \le x \le b} |y(x)| \quad (120)$$

on the space $\mathcal{C}^0([a,b], \mathbb{R}^n)$, or 1-norm

$$\|y\|_1 = \max_{a \le x \le b} |y(x)| + \max_{a \le x \le b} |y'(x)| \quad (121)$$

on the space $\mathcal{C}^1([a,b], \mathbb{R}^n)$.

A linear functional $\delta J|_y : V \to \mathbb{R}$ is called the first variation of J at y if for all η and all α we have

$$J(y + \alpha\eta) = J(y) + \delta J|_y(\eta)\alpha + o(\alpha) \quad (122)$$

The first variation as defined above corresponds to the so-called Gateaux derivative of J, which is just the usual derivative of $J(y+\alpha\eta)$ with respect to α (for fixed y and η) evaluated at $\alpha = 0$:

$$\delta J|_y(\eta) = \lim_{\alpha \to 0} \frac{J(y + \alpha\eta) - J(y)}{\alpha} \quad (123)$$

Then the first-order necessary condition for optimality: for all admissible perturbations η (or, strictly speaking, $\alpha\eta$), we must have

$$\delta J|_{y^*}(\eta) = 0 \quad (124)$$

Alternatively, the first variation defined as

$$J(y + \eta) = J(y) + \delta J|_y(\eta) + o(\|\eta\|) \quad (125)$$

corresponds to the so-called Frèchet derivative of J, which is a stronger differentiability notion than the Gateaux derivative.

A quadratic form $\delta^2 J|_y : V \to \mathbb{R}$ is called the second variation of J at y if for all $\eta \in V$ and all α we have

$$J(y + \alpha\eta) = J(y) + \delta J|_y(\eta)\alpha + \delta^2 J|_y(\eta)\alpha^2 + o(\alpha^2) \quad (126)$$

The second-order necessary condition for optimality: if y^* is a local minimum of J over $A \subset V$, then for all admissible perturbations η we have

$$\delta^2 J|_{y^*}(\eta) \ge 0 \quad (127)$$

In other words, the second variation of J at y^* must be positive semidefinite on the space of admissible perturbations.

Extrema of J with the respect to the 0-norm are called strong extrema, and those with the respect to the 1-norm are called weak extrema.

Let's consider the function space V to be $\mathcal{C}^1([a,b], \mathbb{R}^n)$, the subset A consists of functions $y \in V$ satisfying the boundary conditions

$$y(a) = y_0, \quad y(b) = y_1 \tag{128}$$

and the functional J to be minimized is

$$J(y) = \int_a^b L\left(x, y(x), y'(x)\right) dx \tag{129}$$

We'll use the 1-norm. L is a running cost which is also called Lagrangian. In $L\left(x, y(x), y'(x)\right)$ even though y and y' are the position and velocity along the curve, L is to be viewed as a function of three independent variables. To emphasize this fact, we will sometimes write $L = L(x, y, z)$.

By using (124) (dropping the asterisk in y^*) and Taylor expanding

$$J(y + \alpha\eta) = \int_a^b L\left(x,\, y(x) + \alpha\eta(x),\, y'(x) + \alpha\eta'(x)\right) dx$$

to the first-order with respect to α we get Euler-Lagrange equation providing the first-order necessary condition for optimality

$$L_y\left(x, y(x), y'(x)\right) = \frac{d}{dx} L_z\left(x, y(x), y'(x)\right) \quad \forall x \in [a,b] \tag{130}$$

often written in the shorter form

$$L_y = \frac{d}{dx} L_{y'} \tag{131}$$

Written out in detail, the right-hand side of (130) is

$$\frac{d}{dx} L_z\left(x, y(x), y'(x)\right) = L_{zx}\left(x, y(x), y'(x)\right) + L_{zy}\left(x, y(x), y'(x)\right) y'(x) + L_{zz}\left(x, y(x), y'(x)\right) y''(x)$$

Trajectories satisfying the Euler-Lagrange equation (130) are called extremals (of the functional J). Since the Euler-Lagrange equation is only a necessary condition for optimality, not every extremal is an extremum.

4.1 Light propagation in homogeneous medium.

△

Let's minimize the time, or equivalently, distance between 2 points:

$$J = \int_a^b ds = \int_a^b \sqrt{dx^2 + dy^2} = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$

The Lagrangian is$L = \sqrt{1 + (y')^2}$. Using (130) and noting that y does not appear explicitly in L, leaves us with

$$\frac{d}{dx}\frac{\partial L}{\partial y'} = 0$$

Substituting for L and taking the derivative,

$$\frac{d}{dx}\frac{2y'}{2\sqrt{1 + (y')^2}} = 0$$

Therefore,

$$\frac{y'}{\sqrt{1 + (y')^2}} = c$$

Then

$$\frac{(y')^2}{1 + (y')^2} = c^2$$

where $0 \leq c^2 < 1$. Hence

$$(y')^2 = \frac{c^2}{1 - c^2}$$

or

$$y' = c_1$$

where c_1 is another constant which can be found from the boundary conditions. Hence

$$y = ax + b$$

which is a straight line.

▲

The momentum is defined as

$$p := L_{y'}(x, y, y') \tag{132}$$

and the Hamiltonian (the total energy in a mechanical system):

$$H(x, y, y', p) := p \cdot y' - L(x, y, y') \tag{133}$$

If y is an extremal, i.e. satisfies the Euler-Lagrange equation (131) then the differential equations describing the evolution of y and p along such a curve, when written in terms of the Hamiltonian H, take a particularly nice form. For y, we have

$$\frac{dy}{dx} = y'(x) = H_p\left(x, y(x), y'(x)\right)$$

and for p we have

$$\frac{dp}{dx} = \frac{d}{dx} L_{y'}\left(x, y(x), y'(x)\right) = L_y\left(x, y(x), y'(x)\right) = -H_y\left(x, y(x), y'(x)\right)$$

In more concise form the result is

$$y' = H_p, \quad p' = -H_y \tag{134}$$

which is known as the system of Hamilton's canonical equations.

An important additional observation is that the partial derivative of H with respect to y' is

$$H_{y'}\left(x, y, y', p\right) = p - L_{y'}\left(x, y, y'\right) = 0 \tag{135}$$

Second-order **sufficient** condition for optimality: An extremal $y(\cdot)$ is a strict minimum if $L_{y'y'}\left(x, y(x), y'(x)\right) > 0$ for all $x \in [a, b]$ and the interval $[a, b]$ contains no points conjugate to a.

Maximum Principle for the Basic Variable-Endpoint Control Problem (Pontryagin school): Let $u^* : [t_0, t_f] \to U$ be an optimal control and let $x^* : [t_0, t_f] \to \mathbb{R}^n$ be the corresponding optimal state trajectory. Then there exist a function $p^* : [t_0, t_f] \to \mathbb{R}^n$ and a constant $p_0^* \leq 0$ satisfying $(p_0^*, p^*(t)) \neq (0, 0)$ for all $t \in [t_0, t_f]$ and having the following properties:

1. x^* and p^* satisfy the canonical equations

$$\begin{cases} \dot{x}^* &= H_p(x^*, u^*, p^*, p_0^*) \\ \dot{p}^* &= -H_x(x^*, u^*, p^*, p_0^*) \end{cases} \tag{136}$$

with respect to the Hamiltonian

$$H(x, u, p, p_0) = \langle p, f(x, u) \rangle + p_0 L(x, u) \tag{137}$$

with the boundary conditions $x^*(t_0) = x_0$ and $x^*(t_f) \in S_1$, where the target set is of the form $S = [t_0, \infty) \times S_1$, where S_1 is a k-dimensional

surface in $\mathbb{R}^n$ for some nonnegative integer $k \leq n$. We define such a surface via equality constraints:

$$S_1 = \{x \in \mathbb{R}^n \,:\, h_1(x) = h_2(x) = \cdots = h_{n-k}(x) = 0\}$$

where $h_i,\ i = 1, \ldots, n-k$ are $\mathcal{C}^1$ functions from $\mathbb{R}^n$ to $\mathbb{R}$. We also assume that every $x \in S_1$ is a regular point.

2. For all $t \in [t_0, t_f]$ and all $u \in U$

$$H\left(x^*(t), u^*(t), p^*(t), p_0^*\right) \geq H\left(x^*(t), u, p^*(t), p_0^*\right) \tag{138}$$

3. :

$$H\left(x^*(t), u^*(t), p^*(t), p_0^*\right) = 0 \tag{139}$$

not only for t_f but at any moment of time, $t_0 \leq t \leq t_f$ (and $p_0(t) \leq 0$).

4. The vector $p^*(t_f)$ is orthogonal to the tangent space to S_1 at $x^*(t_f)$:

$$\langle x^*(t_f), d\rangle = 0 \quad \forall d \in T_{x^*(t_f)} S_1 \tag{140}$$

(the transversality condition). The tangent space can be characterized as

$$T_{x^*(t_f)} S_1 = \left\{\, d \in \mathbb{R}^n \,:\, \langle \nabla h_i\left(x^*(t_f)\right), d\rangle = 0, \quad i = 1, \ldots, n-k\right\} \tag{141}$$

and that (140) is equivalent to saying that $p^*(t_f)$ is a linear combination of the gradient vectors $\nabla h_i\left(x^*(t_f)\right),\ i = 1, \ldots, n-k$. Note that when $k = n$ and hence $S_1 = \mathbb{R}^n$, the transversality condition reduces to $p^*(t_f) = 0$ (because the tangent space is the entire $\mathbb{R}^n$).

Note: we can define an additional state variable, $x^0 \in \mathbb{R}$, to be the solution of

$$\dot{x}^0 = L(x, u), \quad x^0(t_0) = 0$$

and arrive at the augmented system

$$\begin{aligned} \dot{x}^0 &= L(x, u) \\ \dot{x} &= f(x, u) \end{aligned}$$

with the initial condition $\begin{pmatrix} 0 \\ x_0 \end{pmatrix}$. The cost can then be rewritten as

$$J(u) = \int_{t_0}^{t_f} \dot{x}^0(t) dt = x^0(t_f)$$

i.e. there is a terminal cost and no running cost.

It should be noted that the maximum principle provides only necessary conditions for optimality. Also keep in mind that an optimal control may not even exist. To verify we can rely on **Filippov's theorem**: *Given a control system in the standard form (114) with $u \in U$, assume that its solutions exist on a time interval $[t_0, t_f]$ for all controls $u(\cdot)$ and that for every pair (t, x) the set $\{f(t, x, u) \,:\, u \in U\}$ is compact and convex. Then the reachable set $R^t(x_0)$ is compact for each $t \in [t_0, t_f]$.*

4.2 Rocket flying vertically

A rocket flying vertically in the gravitational field is described by the differential equation (the air resistance is neglected):

$$m\frac{dv}{dt} = -mg - c\frac{dm}{dt} \tag{142}$$

where m is the rocket mass, v is its speed, g is the free-fall acceleration, c is the exhaust flow velocity. For the period $[0, T]$ the rocket will lift to the height

$$h = \int_0^T v(t)dt$$

Let's consider a problem of finding the optimal law of changing the engines thrust which will provide the maximum rocket elevation. Let m_0 be the rocket mass at time $t = 0$, m_1 ($m_1 < m_0$) is the mass at time T. Renaming $v(t) = x(t)$, $\frac{m(t)}{m_0} = y(t)$, $\frac{dy}{dt} = -u$, the equation (142) can be rewritten as the following system of differential equations:

$$\begin{cases} \frac{dx}{dt} &= \frac{c}{y}u - g \\ \frac{dy}{dt} &= -u \end{cases} \tag{143}$$

We'll consider the acceptable controls such that

$$0 \le u(t) \le u_0$$

The last inequality means that the fuel consumption $-\frac{dm}{dt}$ is limited above by the constant $m_0 u_0$.

The exact mathematical formulation of the problem is thus read: among all possible controls $u = u(t)$ ($0 \le u(t) \le u_0$) such that the corresponding solution $x = x(t)$, $y = y(t)$ of the system of equations (143) satisfies the conditions

$$x(0) = 0, \quad x(T) = 0 \tag{144}$$

$$y(0) = 1, \quad y(T) = \mu \quad (\mu = \frac{m_1}{m_0},\ 0 < \mu < 1) \tag{145}$$

at some $T > 0$, find such that the functional

$$J = -\int_0^T x(t)dt \tag{146}$$

reaches its minimum. (Here u_0 and μ are given positive constants.)

To solve the problem we'll use (137). In our case

$$H = -p_0 x + p_1(\frac{c}{y}u - g) - p_2 u = -p_0 x - p_1 g + (p_1\frac{c}{y} - p_2)u \tag{147}$$

$$H^* = \sup_{0\le u\le u_0} H = \begin{cases} -p_0 x - p_1 g + (p_1\frac{c}{y} - p_2)u_0, & if\ p_1\frac{c}{y} - p_2 \ge 0 \\ -p_0 x - p_1 g, & if\ p_1\frac{c}{y} - p_2 < 0 \end{cases}$$

therefore from (138)

$$u = \begin{cases} u_0, & if\ p_1\frac{c}{y} - p_2 \ge 0 \\ 0, & if\ p_1\frac{c}{y} - p_2 < 0 \end{cases} \tag{148}$$

Substituting (137) into (136) yields

$$\frac{\partial p_i}{\partial t} = -\frac{\partial H}{\partial x_i} = -p_0\frac{\partial f^0}{\partial x_i} - \sum_{k=1}^{n} p_k\frac{\partial f^k}{\partial x_i} \tag{149}$$

In our case (cf. (146) and (143))

$$\begin{cases} \frac{\partial p_1}{\partial t} = & p_0 - p_1\frac{\partial}{\partial x}(\frac{c}{y}u - g) - p_2\frac{\partial}{\partial x}(-u) = p_0 \\ \frac{\partial p_2}{\partial t} = & -p_0\frac{\partial}{\partial y}(-x) - p_1\frac{\partial}{\partial y}(\frac{c}{y}u - g) - p_2\frac{\partial}{\partial y}(-u) = p_1\frac{cu}{y^2} \end{cases} \tag{150}$$

where $p_0 = const \le 0$.
Note that because (143) and (150)

$$\frac{d}{dt}\left(p_1\frac{c}{y} - p_2\right) = \frac{c}{y}p_1 - p_1\frac{c}{y^2}\frac{dy}{dt} - \frac{cu}{y^2}p_1 = \frac{c}{y}p_1 \le 0 \tag{151}$$

since $p_0 \le 0$ and $y > 0$ (as y is not increasing and changing from 1 to μ). Hence the function $p_1\frac{c}{y} - p_2$ is not increasing and therefore it changes its sign no more than once. Therefore the sought trajectory consists of no more than 2 pieces, one of which has $u = u_0$ and the other $u = 0$.

At the start of the sought trajectory $u \ne 0$ because if it were, from (143) and (144) $x = -gt$, i.e. the rocket will go down. So, $u(0) = u_0$. In this case the first part of the trajectory is determined by

$$\begin{cases} y = -u_0 t + k_1 = 1 - u_0 t \\ x = -c\ln(1 - u_0 t) - gt \end{cases} \tag{152}$$

If along all trajectory $u = u_0$ then the function $p_1\frac{c}{y} - p_2$ does not change its sign and nonnegative because (148). From (151) it follows that

$$p_1\frac{c}{y} - p_2 = \int \frac{p_0 c}{1 - u_0 t}dt = -\frac{p_0 c}{u_0}\ln(1 - u_0 t) + k_1$$

$$\left(0 \le t < \frac{1}{u_0}, \quad p_0 \le 0\right).$$

Hence we can see that the function $p_1\frac{c}{y} - p_2$ either is non-positive (if $k_1 \le 0$) or changes its sign (if $k_1 > 0$). This contradicts out assumption.

So, the sought trajectory must consist from 2 pieces. The first one has $u = u_0$, and at the second $u = 0$.

Let the function $p_1\frac{c}{y} - p_2$ changes its sign at $t = \tau$, at that $p_1\frac{c}{y} - p_2 > 0$ when $0 \le t < \tau$ and $p_1\frac{c}{y} - p_2 < 0$ when $\tau < t \le T$. Then according to (151)

$$p_1\frac{c}{y} - p_2 = \begin{cases} -\frac{p_0 c}{u_0}\ln(1 - u_0 t) + k_1, & if\ 0 \le t < \tau \\ p_0 ct + k_2, & if\ \tau < t \le T \end{cases} \tag{153}$$

From the equality $\left(p_1\frac{c}{y} - p_2\right)(\tau) = 0$ the constants k_1 and k_2 can be found uniquely.

From (139) and (147) we obtain $-p_0 x(T) - p_1(T)g = 0$, i.e.

$$p_1(T) = 0 \tag{154}$$

By virtue of the first equation of (150)$p_1 = p_0 t + k_3$. Hence, using (154)

$$p_1 = p_0(t - T) \tag{155}$$

From the equations (153) and (155) we can uniquely find p_2. The constant p_0 that is in p_1 and p_2, is not a significant parameter. It can be set, for example, to $p_0 = -1$.

So, when $0 \le t < \tau$ the trajectory is determined by (152).

When $\tau < t \le T$ we get (since $u = 0$) $x = -gt + k_1, \quad y = k_2$.

Because (144) and (145) $x(T) = 0, \quad y(T) = \mu$, then

$$\begin{cases} x &= g(T - t) \\ y &= \mu \end{cases} \tag{156}$$

Because the trajectory is continuous when $t = \tau$, from (152), (156) we obtain $1 - u_0\tau = \mu, \quad -c\ln(1 - u_0\tau) - g\tau = g(T - \tau)$, and it follows

$$\begin{aligned} \tau &= \tfrac{1-\mu}{u_0} \\ T &= -\tfrac{c\ln\mu}{g} \end{aligned}$$

Thereby, we found the optimal controls:

$$u = \begin{cases} u_0, & if\ 0 \le t < \frac{1-\mu}{u_0} \\ 0, & if\ \frac{1-\mu}{u_0} < t \le -\frac{c\ln\mu}{g} \end{cases}$$

and together the corresponding trajectory. Obviously, the calculations make sense only when $-\frac{cu_0\ln\mu}{g} \ge 1 - \mu \quad (T \ge \tau)$.

Let's demonstrate that this trajectory is optimal. Indeed, let $x(t),\ y(t)$ be an arbitrary solution of the system (143) satisfying (144) and (145) at some $T > 0$. Integrating the first equation from (143) along $[0, t]$ we obtain

$$x(t) - x(0) = -gt - c\int_0^t d\ln y = -gt - c\ln y(t) + c\ln y(0)$$

Hence, according to (144) and (145) we get

$$x(t) = -gt - c\ln y(t) \tag{157}$$

Setting here $t = T$, we obtain $T = -\frac{c}{g}\ln\mu$. Therefore, this solution is defined on the same period that the above.

Integrating (157) along $[0, T]$, we find the maximum elevation of the rocket

$$h = -\frac{gT^2}{2} - c\int_0^T \ln y(t)dt \tag{158}$$

Integrating the conditions $\frac{dy}{dt} \le 0 \quad (u \ge 0)$ along $[t, T]$ we obtain

$$y(t) \ge \mu \quad (0 \le t \le T) \tag{159}$$

Similarly, integrating $\frac{dy}{dt} \ge -u_0 \quad (u \le u_0)$ along $[0, t]$ we get

$$y(t) \ge 1 - u_0 t \quad (0 \le t \le T) \tag{160}$$

From the inequalities (159) and (160) it follows that

$$y(t) \ge max\{\mu, 1 - u_0 t\} = \begin{cases} 1 - u_0 t, & if\ 0 \le t \le \frac{1-\mu}{u_0} \\ \mu, & if\ \frac{1-\mu}{u_0} < t \le -\frac{c\ln\mu}{g} \end{cases}$$

On the right we got the same function as in our derivation above. From (158) it follows that the functional h gets the maximum value on it. So, the found above trajectory is optimal.

Using (152), (156) and (158) we can obtain the formula for the maximum rocket elevation:

$$\begin{aligned} h &= \int_0^\tau [-c\ln(1 - u_0 t) - gt]\, dt + \int_\tau^T g(T - t)dt \\ &= \frac{c}{u_0}\left[(1 - u_0 t)\ln(1 - u_0 t) - (1 - u_0 t)\right]\Big|_0^\tau - \frac{g\tau^2}{2} + \left[gTt - \frac{gt^2}{2}\right]\Big|_\tau^T \\ &= \frac{c(1 - \mu + \ln\mu)}{u_0} + \frac{(c\ln\mu)^2}{2g} \end{aligned}$$

▲

Let us introduce the *value function*

$$V(t, x) := \inf_{u_{[t,t_1]}} J(t, x, u) \tag{161}$$

where the notation $u_{[t,t_1]}$ indicates that the control u is restricted to the interval $[t, t_1]$, and the family of the cost functionals

$$J(t,x,u) = \int_t^{t_1} L\left(s, x(s), u(s)\right) ds + K\left(x(t_1)\right) \tag{162}$$

where t ranges over $[t_0, t_1)$ and x ranges over $\mathbb{R}^n$. If an optimal control exists, then the infimum turns into a minimum and V coincides with the optimal cost-to-go. It is clear that the value function must satisfy the boundary condition

$$V(t_1, x) = K(x) \quad \forall x \in \mathbb{R}^n \tag{163}$$

The **principle of optimality**: *For every $(t,x) \in [t_0, t_1) \times \mathbb{R}^n$ and every $\Delta t \in (0, t_1 - t]$, the value function V defined in (161) satisfies the relation*

$$V(t,x) = \inf_{u_{[t,t+\triangle t]}} \left\{ \int_t^{t+\triangle t} L\left(s, x(s), u(s)\right) ds + V\left(t + \triangle t, x(t+\triangle t)\right) \right\} \tag{164}$$

where $x(\cdot)$ on the right-hand side is the state trajectory corresponding to the control $u_{[t,t+\triangle t]}$ and satisfying $x(t) = x$.

By Taylor expanding and taking $\triangle t \to 0$ we obtain the **Hamilton-Jacobi-Bellman (HJB)** equation:

$$-V_t(t,x) = \inf_{u \in U} \{L(t,x,u) + \langle V_x(t,x), f(t,x,u)\rangle\} \tag{165}$$

that holds for all $t \in [t_0, t_1)$ and all $x \in \mathbb{R}^n$. The accompanying boundary condition is (163).

For different target sets, the boundary condition changes but the HJB equation remains the same. However, the HJB equation will not hold for $(t,x) \in S$ just like it does not hold at $t = t_1$ in the fixed-time case, because the principle of optimality is not valid there.

We can rewrite (165) in the equivalent form:

$$V_t(t,x) = \sup_{u \in U} \{\langle -V_x(t,x), f(t,x,u)\rangle - L(t,x,u)\} \tag{166}$$

Recalling (137) as

$$H(t,x,u,p) := \langle p, f(t,x,u)\rangle - L(t,x,u)$$

we see that the expression inside the supremum in (166) is nothing but the Hamiltonian, with $-V_x$ playing the role of the costate. This brings us to the Hamiltonian form of the HJB equation:

$$V_t(t,x) = \sup_{u \in U} H\left(t, x, u, -V_x(t,x)\right) \tag{167}$$

Assuming that an optimal control exists we can establish the following **sufficient** condition for optimality: *Suppose that a $\mathcal{C}^1$ function $\hat{V} : [t_0, t_1] \times \mathbb{R}^n \to \mathbb{R}$ satisfies the HJB equation*

$$-\hat{V}_t(t,x) = \inf_{u\in U}\left\{L(t,x,u) + \left\langle \hat{V}_x(t,x), f(t,x,u)\right\rangle\right\} \tag{168}$$

(for all $t \in [t_0, t_1)$ and all $x \in \mathbb{R}^n$) and the boundary condition

$$\hat{V}(t_1,x) = K(x) \tag{169}$$

Suppose that a control $\hat{u} : [t_0,t_1] \to U$ and the corresponding trajectory $\hat{x} : [t_0,t_1] \to \mathbb{R}^n$, with the given initial condition $\hat{x}(t_0) = x_0$, satisfy everywhere the equation

$$L\left(t,\hat{x}(t),\hat{u}(t)\right) + \left\langle \hat{V}_x\left(t,\hat{x}(t)\right), f\left(t,\hat{x}(t),\hat{u}(t)\right)\right\rangle = \min_{u\in U}\left\{L\left(t,\hat{x}(t),u\right) + \left\langle \hat{V}_x\left(t,\hat{x}(t)\right), f\left(t,\hat{x}(t),u\right)\right\rangle\right\} \tag{170}$$

which is equivalent to the Hamiltonian maximization condition

$$H\left(t,\hat{x}(t),\hat{u}(t),-\hat{V}_x\left(t,\hat{x}(t)\right)\right) = \max_{u\in U} H\left(t,\hat{x}(t),u,-\hat{V}_x\left(t,\hat{x}(t)\right)\right) \tag{171}$$

Then $\hat{V}(t_0,x_0)$ is the optimal cost (i.e. $\hat{V}(t_0,x_0) = V(t_0,x_0)$ where V is the value function) and $\hat{u}$ is an optimal control. (Note that this optimal control is not claimed to be unique; there can be multiple controls giving the same cost.)

We can regard the function $\hat{V}$ as providing a tool for verifying optimality of candidate optimal controls (obtained, for example, from the maximum principle). This optimality is automatically global.

Let us assume for clarity that the system and the cost are time-invariant. The maximum principle is formulated in terms of the canonical equations

$$\dot{x}^* = H_p\Big|_*, \quad \dot{p}^* = -H_x\Big|_* \tag{172}$$

and says that at each time t, the value $u^*(t)$ of the optimal control must maximize $H\left(x^*(t),u,p^*(t)\right)$ with respect to u:

$$u^*(t) = \arg\max_{u\in U} H\left(x^*(t),u,p^*(t)\right) \tag{173}$$

This is an *open-loop* specification, because $u^*(t)$ depends not only on the state $x^*(t)$ but also on the costate $p^*(t)$ which has to be computed from the adjoint differential equation. Now, in the context of the HJB equation, the optimal control must satisfy

$$u^*(t) = \arg\max_{u\in U} H\left(x^*(t),u,-V_x(t,x^*(t))\right) \tag{174}$$

This is a *closed-loop* (feedback) specification; indeed, assuming that we know the value function V everywhere, $u^*(t)$ is completely determined by the current state $x^*(t)$. The ability to generate an optimal control policy in the form of

a state feedback law is an important feature of the dynamic programming approach. Clearly, we cannot implement this feedback law unless we can first find the value function by solving the HJB partial differential equation. This is in general can be a very difficult task. Therefore, from the computational point of view the maximum principle has an advantage in that it involves only ordinary and not partial differential equations. In principle, the dynamic programming approach provides more information (including sufficiency), but in reality, the maximum principle is often easier to use and allows one to solve many optimal control problems for which the HJB equation is analytically intractable.

Comparing (173) and (174) it is clearly seen that

$$p^*(t) = -V_x\left(t, x^*(t)\right) \tag{175}$$

In the proof of the maximum principle, the adjoint vector p^* was defined as the normal to a suitable hyperplane. In our earlier discussions, it was also related to the momentum and to the vector of Lagrange multipliers. From (175) we now have another interpretation of the adjoint vector in terms of the gradient of the value function, i.e., the sensitivity of the optimal cost with respect to the state x. In economic terms, this quantity corresponds to the"marginal value", or "shadow price"; it tells us by how much we can increase benefits by increasing resources/spending, or how much we would be willing to pay someone else for resources and still make a profit.

However, the value function must have a well-defined gradient and, moreover, that this gradient can be further differentiated with respect to time. In other words, we need the existence of second-order partial derivatives of V. Unfortunately, we cannot expect this to be true in general. However, this can be mitigated by the viscous solutions.

4.3 Static Linear Quadratic Regulator (LQR) problem

Let's consider a simplified (finite-horizon) LQR system described by the following equation:

$$\dot{x} = ax + bu, \qquad x(t_0) = x_0 \tag{176}$$

with $x \in \mathbb{R}$ and $u \in \mathbb{R}$ (the control is unconstrained); the target set is $S \in \{t_1\} \times \mathbb{R}$, where t_1 is a fixed time (so this is a fixed-time, free-endpoint problem), a, b are constants; and the cost functional is

$$J(u) = \int_{t_0}^{t_1} (qx^2 + ru^2)dt \tag{177}$$

where $q = const \geq 0$, $r = const > 0$; no terminal cost. The quadratic cost is very reasonable: it penalizes both the size of the state and the control effort. Incidentally, the formula $L(t, x, u) = qx^2 + ru^2$ for the running cost is another justification of the acronym LQR.

Specialized to our present LQR problem, the HJB equation (165) becomes

$$-V_t(t,x) = \inf_{u\in\mathbb{R}} \left\{qx^2 + ru^2 + (ax+bu)V_x\right\} \tag{178}$$

the boundary condition (163) reads

$$V(t_1,x) = 0 \tag{179}$$

Since $r > 0$, it is easy to see that the infimum of the quadratic function of u in (178) is a minimum and to calculate that the minimizing control is

$$2ru + bV_x = 0$$

$$u^* = -\frac{b}{2r}V_x \tag{180}$$

We substitute this control into (178) and, after some term cancellations, bring the HJB equation to the following form:

$$\begin{aligned} -V_t(t,x) &= qx^2 + \frac{b^2}{4r}V_x^2 + axV_x - \frac{b^2}{2r}V_x^2 \\ &= -\frac{b^2}{4r}V_x^2 + axV_x + qx^2 \end{aligned} \tag{181}$$

This PDE can be solved:

$$V(t,x) = \left(\frac{b^2}{4r}k_2^2 - ak_2x - qx^2\right)t + k_2x + k_1 \tag{182}$$

Using the boundary conditions in (176) and (179) we can express k_1 *and* k_2 via $t_0, t_1, x_0$and x_1 (yet unknown). Then we can substitute the partial derivative from (182) $\frac{\partial V}{\partial x} = -ak_2t - 2qtx + k_2$ into (180) to obtain the optimal control. Then the optimal trajectory can be found by solving (176).

We can also approach this problem from the maximum principle point of view. The Hamiltonian is given by

$$H(t,x,u,p) = axp + bup - qx^2 - ru^2 \tag{183}$$

Note that, compared to the general formula (137) for the Hamiltonian, we took the abnormal multiplier p_0 to be equal to -1.

The gradient of H with respect to u is$H_u = bp - 2ru$, and along an optimal trajectory it must vanish. Therefore

$$u^* = \frac{b}{2r}p^* \tag{184}$$

Moreover, since $H_{uu} = -2r < 0$, the above control indeed maximizes the Hamiltonian (globally). The costate satisfies the adjoint equation:

$$\dot{p}^* = -H_x\Big|_* = -ap^* + 2qx^* \tag{185}$$

with the boundary condition

$$p^*(t_1) = 0 \tag{186}$$

Let us show that p^* can be represented in the form

$$p^*(t) = -2P(t)x^*(t) \tag{187}$$

Putting together the dynamics (176) of the state, the control law (184), and the dynamics (185) of the costate, we can write the system of canonical equations in the following combined closed-loop form:

$$\begin{pmatrix} \dot{x}^* \\ \dot{p}^* \end{pmatrix} = \begin{pmatrix} a & \frac{b^2}{2r} \\ 2q & -a \end{pmatrix} \begin{pmatrix} x^* \\ p^* \end{pmatrix} =: \mathcal{H}(t) \begin{pmatrix} x^* \\ p^* \end{pmatrix} \tag{188}$$

The matrix $\mathcal{H}(t)$ is sometimes called the Hamiltonian matrix. Let us denote the state transition matrix for the linear time-varying system (188) by $\Phi(\cdot,\cdot)$. Then we have, in particular, $\begin{pmatrix} x^*(t) \\ p^*(t) \end{pmatrix} = \Phi(t,t_1) \begin{pmatrix} x^*(t_1) \\ p^*(t_1) \end{pmatrix}$; here $\Phi(t,t_1) = \Phi^{-1}(t_1,t)$ propagates the solutions backward in time from t_1 to t. Φ can be partitioned to

$$\Phi = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix}$$

Due to our terminal condition (186), the addendum with Φ_{12} and Φ_{22} are 0. So, we have

$$\begin{aligned} x^*(t) &= \Phi_{11}(t,t_1)x^*(t_1) \\ p^*(t) &= \Phi_{21}(t,t_1)p^*(t_1) \end{aligned} \tag{189}$$

Solving the first equation of (189) for $x^*(t_1)$ and plugging the result into the second equation, we obtain

$$p^*(t) = \frac{\Phi_{21}(t,t_1)}{\Phi_{11}(t,t_1)} x^*(t)$$

We have thus established (187) with

$$P(t) = -\frac{1}{2}\frac{\Phi_{21}(t,t_1)}{\Phi_{11}(t,t_1)} \tag{190}$$

Combining (184) and (187), we deduce that the optimal control must take the form

$$u^*(t) = -\frac{b}{r}P(t)x^*(t) = \frac{bq}{ar}x^*(t) \tag{191}$$

Substituting this into (176) we write

$$\dot{x}^* = ax^* + \frac{b^2 q}{ar} x^*$$

Solving this we obtain the optimal trajectory

$$x^* = k_1 e^{(a+\frac{b^2 q}{ar})t}$$

or with the initial condition

$$x^* = x_0 e^{(a+\frac{b^2 q}{ar})(t-t_0)} \tag{192}$$

▲

Stochastic Optimal Control

Let's consider a dynamical system

$$x(t+dt) = x(t) + \mu\left(t, x(t), u(t)\right) dt + \sigma\left(t, x(t)\right) dW(t) \tag{193}$$

where W is a Wiener process. Let $F(t,x)$ be a function that is differentiable at least once in t and twice in x. The"total differential" of $F(t,x(t))$ can be approximated with a Taylor series expansion. Using the notation $F_t \equiv \frac{\partial F}{\partial t}$ and substituting from (193) we find that

$$\begin{aligned} dF &= F_t dt + F_x dx + \frac{1}{2} F_{xx} (dx)^2 + \dots \\ &= F_t dt + F_x(\mu dt + \sigma dW) + \frac{1}{2} F_{xx} \left(\mu^2 (dt)^2 + 2\mu\sigma dt dW + \sigma^2 (dW)^2\right) + \dots \end{aligned}$$

Dropping terms of order higher than dt or $(dW)^2$ we obtain

$$dF = F_t dt + \mu F_x dt + \sigma F_x dW + \frac{1}{2}\sigma^2 F_{xx} (dW)^2 \tag{194}$$

Since $\mathbb{E}[dW] = 0$ and $\mathbb{E}\left[(dW)^2\right] = dt$, taking expectations in (194) gives (Itô's lemma)

$$\mathbb{E}[dF] = \left(F_t + \mu F_x + \frac{1}{2}\sigma^2 F_{xx}\right) dt \tag{195}$$

and

$$\mathbb{V}[dF] = \mathbb{E}\left[dF - \mathbb{E}[dF]\right]^2 = \sigma^2 F_x^2 dt \tag{196}$$

Let's rewrite the principle of optimality ((164))

$$\begin{aligned}\mathbb{E}\left[V\left(t+dt, x(t+dt)\right)\right] &= V(t,x) + V_t dt + V_x \mathbb{E}[dx] + \frac{1}{2} V_{xx} \mathbb{E}[dx^2] + o(dt) \\ &= V(t,x) + V_t dt + \mu V_x dt + \frac{1}{2}\sigma^2 V_{xx} dt + o(dt)\end{aligned}$$

and

$$V(t,x) = \inf_U \left\{ \int_t^{t+dt} L(s,x,u)ds + V(t,x) + \left(V_t + \mu V_x + \frac{1}{2}\sigma^2 V_{xx} \right) dt \right\}$$

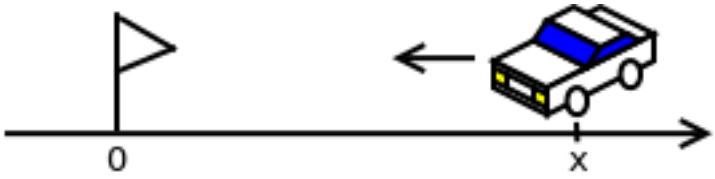
Because V does not depend on u it can be pulled out of infimum and it cancels out. The integral can be approximated as

$$\int_t^{t+dt} L(s,x,u)ds = L(t,x,u)dt + o(dt)$$

therefore, dividing by dt, we obtain the **stochastic HJB**:

$$-V_t(t,x) = \inf_{u \in U} \left\{ L(t,x,u) + \mu(t,x,u)V_x + \frac{1}{2}\sigma(t,x)^2 V_{xx} \right\} \tag{197}$$

4.4 Optimal Parking problem



Suppose you're driving down a one-way road towards your destination, a theater, looking for a parking space. As you drive along, you can observe only one parking place at a time, the one right next to you, noting whether or not it's vacant. If it is vacant, you may either (1) stop and park there or (2) drive on to the next space. If it is occupied then you must drive on to the next space. You cannot return to vacant spaces that you have passed. Assume that vacant spaces occur independently and that the probability that any given space is vacant is p. If you have not parked by the time you reach your destination, you must park in the pay parking lot, at a cost of c. If you park x spaces away from your destination, then you consider that a"walking" cost of x has been incurred. The closest space (before reaching the pay parking lot) to your destination is

one space away from your destination, i.e. $x = 1$. Furthermore, we assume that $c > 1$. The objective is to minimize your total expected cost.

Let the state $s = (x, i)$ be defined by the number x of the space that is being approached, and i, which is the availability of the space: $i = 0$ indicates that the space is vacant and $i = 1$ indicates that the space is occupied. Let $a = 0$ denote the decision to park and let $a = 1$ denote the decision to drive to the next space. For example, if you decide to park at $s = (3, 0)$, you incur the cost 3 and the process ends. Our action space is $A_{(x,0)} = \{0, 1\}$ and $A_{(x,1)} = \{1\}$ for all $x \geq 0$.

Let $f(s) = f(x, i)$ denote the minimum expected cost of parking, starting in state s. As usual we work backward. In particular,

$$f(1, i) = \begin{cases} min(1, c) = 1 & if\, i = 0 \\ c & if\, i = 1 \end{cases}$$

It is convenient to define

$$F(x) := pf(x, 0) + qf(x, 1) \quad x \geq 0 \tag{198}$$

where $q := 1 - p$. $F(x)$ is the minimum expected cost of parking, given that you are approaching space x (still looking to park) but have not yet observed its availability.

The optimality equations for $x \geq 2$ can therefore be written as follows:

$$f(x, i) = \begin{cases} \min(x, F(x-1)) & if\, i = 0 \\ F(x-1) & if\, i = 1 \end{cases} \tag{199}$$

If $x \leq F(x-1)$ then it is optimal to park, otherwise it is optimal to drive on.

By substituting (199) into (198) we can write

$$F(x) = p \min(x, F(x-1)) + qF(x-1) \tag{200}$$

Let

$$g(x) := F(x-1) - x \tag{201}$$

and substitute $F(x-1) = x + g(x)$ into (199) to rewrite the optimality equations as

$$f(x, i) = \begin{cases} x + \min(0, g(x)) & if\, i = 0 \\ x + g(x) & if\, i = 1 \end{cases} \tag{202}$$

Let us show a few facts:

(a) The function $F(x)$ is decreasing in x because $\min(x, F(x-1)) \leq F(x-1)$ and by (200)

$$F(x) \leq pF(x-1) + qF(x-1) = F(x-1)$$

(b) Using $F(0) = c$, we get by (201) that $g(1) = c - 1 > 0$ (by assumption). By (a), F is decreasing, therefore, $F(x) \leq c \; \forall x$. Select an arbitrary $x \geq c$. Then $g(x) = F(x-1) - x \leq c - c = 0$, i.e. $g(x) \leq 0$ for all $x \geq c$.

(c) The function g is the sum of two decreasing functions, one of which is strictly decreasing, and is therefore also strictly decreasing.

Because of (c) $g(x)$ would increase as x, the number of spaces away, gets smaller. By (202) it makes sense to park after the cutoff point S such that $g(x) < 0$ for $x > S$, and $g(x) \geq 0$ for $x \leq S$. So, the optimal parking policy could be characterized as follows: do not park at any space that is strictly farther away than S and do park at any vacant space that is closer than S.

We can rewrite (202) in terms of g. From (201) we have $g(x+1) = F(x) - (x+1)$. Substituting from (198)

$$g(x+1) = pf(x,0) + qf(x,1) - x - 1$$

and from (202)

$$g(x+1) = p\min(0, g(x)) + qg(x) - 1 \tag{203}$$

For example, suppose $p = 0.2$ and $c = 5$. Then $g(1) = c - 1 = 4$. Next $g(2) = 0.2\min(0,4) + 0.8 * 4 - 1 = 2.2$, $g(3) = 0.2\min(0, 2.2) + 0.8 * 2.2 - 1 = 0.76$, *and* $g(4) = 0.2\min(0, 0.76) + 0.8 * 0.76 - 1 = -0.392 < 0$. That is, it is not optimal to park in space 4 if it is vacant, but it is optimal to park in the first vacant space found thereafter. In short, even though space 4 yields a lower cost than the parking lot, it is worth gambling and looking for a closer vacant space. However, if spot 3 is vacant, it is not worth gambling: you should park in it (rather than driving past to see if space 2 is vacant). Using (201), the minimum expected cost of parking is $F(3)$ for all $x \geq 3$, namely, $F(3) = g(4) + 4 = 3.608$, which, as it should be, is strictly less than the cost ($c = 5$) of driving straight to the parking lot.

In this problem it is possible to derive an explicit formula for the optimal policy. Let $v(x, S)$ be the expected parking cost of approaching space x (before seeing if it's vacant or not) while utilizing the cutoff point S. Thinking recursively, we have the following

$$v(x,S) = \begin{cases} c & if\, x = 0 \\ px + qv(x-1,S) & if\, 1 \leq x \leq S \\ v(S,S) & if\, x > S \end{cases}$$

Let's evaluate $v(S,S)$

$$\begin{aligned} v(S,S) &= pS + qv(S-1,S) \\ &= pS + q\left[p(S-1) + qv(S-2,S)\right] \\ &= p\left[S + qS(-1) + q^2 v(S-2,S)\right] \\ &= p\sum_{i=0}^{S} q^i(S-i) + q^S c \end{aligned}$$

$$p\sum_{i=0}^{S} q^i(S-i) = pS\frac{1-q^{S+1}}{1-q} - p\frac{(Sq-S-1)q^{S+1}+q}{(1-q)^2} = p\frac{S(1-q)(1-q^{S+1}) - (Sq-S-1)q^{S+1}-q}{(1-q)^2}$$
$$= p\frac{q^{S+1}-(S+1)q+S}{(1-q)^2} = \frac{q^{S+1}-Sq+S-q}{p} = S - \frac{q(1-q^S)}{p}$$

So,

$$v(S,S) = S - \frac{q(1-q^S)}{p} + q^S c$$

and we seek to minimize it over S. The first forward difference

$$\begin{aligned}\triangle(S) &:= v(S+1,S+1) - v(S,S)\\ &= S+1 - \frac{q(1-q^{S+1})}{p} + q^{S+1}c - S + \frac{q(1-q^S)}{p} - q^S c\\ &= 1 + \frac{q}{p}(q^{S+1}-q^S) + c(q^{S+1}-q^S)\\ &= 1 + (q^{S+1}-q^S)(\frac{q}{p}+c)\\ &= 1 - q^S(q+pc)\end{aligned}$$

Here, $\triangle(S)$ represents the increase in expected cost from increasing the cutoff point from S to $S+1$. Note that $\triangle(0) = 1-(q+pc) = p(1-c)$ is strictly negative by our assumption ($c > 1$). Choosing $S = 1$ is strictly better than $S = 0$. Thus, $\triangle$ starts at a strictly negative value and is increasing on the positive integers. Hence, the optimal cutoff will be the smallest integer S that satisfies $\triangle(S) \geq 0$: S is better than $S+1$ and strictly better than $S-1$. Using the expression for the difference this inequality becomes

$$q^S \leq \frac{1}{pc+q}$$

Taking logarithms and using the facts that $\ln q < 0$ and $\ln(1/x) = -\ln x$ yields

$$S \geq \frac{\ln(pc+q)}{-\ln q} \tag{204}$$

Returning to the example in which $c = 5$ and $p = 0.2$, we see that $\ln(pc+q) = \ln(1.8) \approx 0.5878$ and $-\ln(0.8) \approx 0.2231$, so $-\ln(pc+q)/\ln q \approx 2.6$. Thus the optimal cutoff level is $S = 3$, as we had previously determined.

▲

4.5 Bride problem (Secretary problem)

The bride problem in its simplest form has the following features:

1. You search for one bride.
2. The number n of candidates is known.
3. You meet girls sequentially in random order, each order being equally likely.
4. It is assumed that you can rank all the brides from best to worst without ties. The decision to accept or reject a bride must be based only on the relative ranks of those candidates met so far.
5. A bride once rejected cannot later be recalled.
6. You are very particular and will be satisfied with nothing but the very best. (That is, your payoff is 1 if you choose the best of the n candidates and 0 otherwise.)

If you marry the very first bride then the probability of her being the best is $1/n$. The same is for the last bride. We will improve the odds by following this strategy: for some integer $r \geq 1$ you reject the first r brides, and then marry the next bride who is better than all seen so far. The probability $p(r)$ of selecting the best bride for $r > 1$ is

$$\begin{aligned} p_n(r) &= \sum_{j=1}^{n} p\left(bride\, j\, is\, selected \cap bride\, j\, is\, the\, best\right) \\ &= \sum_{j=1}^{n} p\left(bride\, j\, is\, selected \,|\, bride\, j\, is\, the\, best\right) p\left(bride\, j\, is\, the\, best\right) \\ &= \left[\sum_{j=1}^{r-1} 0 + \sum_{j=r}^{n} p\left(the\, best\, of\, the\, first\, j-1\, brides\, is\, the\, first\, r-1\, brides \,|\, bride\, j\, is\, the\, best\right)\right] \cdot \frac{1}{n} \\ &= \sum_{j=r}^{n} \frac{1}{n} \frac{r-1}{j-1} = \frac{r-1}{n} \sum_{j=r}^{n} \frac{1}{j-1} \end{aligned} \tag{205}$$

The optimal r is the one that maximizes this probability. For small values of n, the optimal r can easily be computed. Of interest are the approximate values of the optimal r for large n. If we let n tend to infinity and write x as the limit of r/n, then using t for j/n and dt for $1/n$, the sum becomes a Riemann approximation to an integral,

$$
\begin{aligned}
p_n(r) &= \frac{r-1}{n}\sum_{j=r}^{n}\frac{n}{j-1}\frac{1}{n} \\
&\to x\int_{x}^{1}\frac{1}{t}dt = -x\ln x
\end{aligned}
\tag{206}
$$

The value of x that maximizes this quantity is easily found by setting the derivative with respect to x equal to zero and then solving for x. When this is done we find that

$$
x^* = \frac{1}{e} \approx 0.37
$$

and optimal probability

$$
p^* = \frac{1}{e}
$$

Thus for large n, it is approximately optimal to wait until about 37% of the brides have been dated and then to select the next relatively best one. The probability of success is also about 37%.

▲

4.6 A Mean-Variance Portfolio Selection problem

Suppose there is a market in which 1 bond and m stocks are traded continuously without transaction costs (and no consumption). The bond price can be described by

$$
\begin{cases}
dP_0(t) &= r(t)P_0(t)dt, \quad t \in [0,T] \\
P_0(0) &= p_0 \ > 0
\end{cases}
\tag{207}
$$

where $r(t) > 0$ is the interest rate of the bond. The stock prices satisfy the stochastic differential equations:

$$
\begin{cases}
dP_i(t) &= P_i(t)b_i(t)dt + P_i(t)\sum_{i=1}^{m}\sigma_{ij}(t)dW^j(t), \quad t \in [0,T] \\
P_i(0) &= p_i \ > 0
\end{cases}
\tag{208}
$$

where the deterministic function $b_i(t) > r(t) > 0$ is the appreciation rate, and $\sigma_i(t) = (\sigma_{i1}(t), \ldots, \sigma_{im}(t))$ is the volatility or the dispersion of the stocks.

The total wealth is

$$
x(t) = \sum_{i=0}^{m} N_i(t)P_i(t) \tag{209}
$$

where $N_i(t)$ is number of shares. Introducing $u_i(t) := N_i(t)P_i(t)$ – the total market value of the ith asset, we write

$$\begin{cases} dx(t) &= \{r(t)x(t) + \sum_{i=1}^{m} [b_i(t) - r(t)]\, u_i(t)\}\, dt + \sum_{j=1}^{m} \sum_{i=1}^{m} \sigma_{ij}(t) u_i(t) dW^j(t) \\ x(0) &= x_0 \end{cases} \tag{210}$$

Note that we allow short-selling, so that $u_i(t) < 0$ would be allowed. We call $u(t) = (u_1(t), \ldots, u_m(t))^T$ a portfolio of the investor. Notice that we exclude the allocation of the bond, $u_0(t)$, from the portfolio, as it will be determined completely by the allocations of the stocks.

The objective of the investor is to maximize the mean terminal wealth, $\mathbb{E}[x(T)]$, and at the same time to minimize the variance of the terminal wealth

$$\mathbb{V}[x(T)] \equiv \mathbb{E}[x(T) - \mathbb{E}[x(T)]]^2 = \mathbb{E}\left[x(T)^2\right] - \mathbb{E}[x(T)]^2 \tag{211}$$

This is a multi-objective optimization problem with two criteria in conflict. We can reformulate the goal as to

$$\min\{-\mathbb{E}[x(T)] + \mu \mathbb{V}[x(T)]\} \tag{212}$$

where $\mu > 0$. It can be shown (see [46]) that with introducing

$$\lambda = 1 + 2\mu \mathbb{E}[x(T)] \tag{213}$$

this optimization is equivalent to

$$\min\left\{\mathbb{E}\left[\mu x^2(T) - \lambda x(T)\right]\right\} \tag{214}$$

Putting $\gamma = \frac{\lambda}{2\mu}$ and $y(t) = x(t) - \gamma$, (214) becomes

$$\min\left\{\mathbb{E}\left[\frac{1}{2}\mu y^2(T)\right]\right\} \tag{215}$$

subject to

$$\begin{cases} dy(t) &= [A(t)y(t) + B(t)u(t) + b(t)]\, dt + \sum_{j=1}^{m} D_j(t) u(t) dW^j(t) \\ y(0) &= x_0 - \gamma \end{cases} \tag{216}$$

where

$$\begin{cases} A(t) = r(t) & B(t) = (b_1(t) - r(t), \ldots, b_m(t) - r(t)) \\ b(t) = \gamma r(t) & D_j(t) = (\sigma_{1j}(t), \ldots, \sigma_{mj}(t)) \end{cases} \tag{217}$$

Thus the problem becomes an SLQ (Stochastic Linear Quadratic) problem. This stochastic Riccati equation with multidimensional Brownian motion has a solution. Let's define

$$\rho(t) := B(t)\left(\sum_{j=1}^{m} D_j(t)^T D_j(t)\right)^{-1} B(t)^T = B(t)\left(\sigma(t)\sigma(t)^T\right)^{-1} B(t)^T$$

Then (216) reduces to (here $P(t)$ is the coefficient in $p^*(t) = -P(t)x^*(t) - \varphi(t)$, where $p(t)$ is the costate)

$$\begin{cases} \dot{P}(t) = (\rho(t) - 2r(t))\, P(t) \\ P(T) = \mu \\ P(T)\left(\sigma(t)\sigma(t)^T\right) > 0 \qquad a.e.\ t \in [0,T] \end{cases} \tag{218}$$

Clearly, the solution of (218) is given by

$$P(t) = \mu e^{-\int_t^T (\rho(s) - 2r(s))ds} \tag{219}$$

and the optimal feedback control is given by

$$\begin{aligned} u^*(t,y) &= (u_1^*(t,y), \ldots, u_m^*(t,y)) \\ &= -\left(\sigma(t)\sigma(t)^T\right)^{-1} B(t)^T \left(y + \frac{\varphi(t)}{P(t)}\right) \\ &= \left(\sigma(t)\sigma(t)^T\right)^{-1} B(t)^T \left(\gamma e^{-\int_t^T r(s)ds} - x\right) \end{aligned} \tag{220}$$

The factor $\left(\sigma(t)\sigma(t)^T\right)^{-1} B(t)^T$ is called the risk premium. Under the above optimal feedback control, the wealth equation (210) evolves as

$$\begin{cases} dx(t) &= \left\{(r(t) - \rho(t))\, x(t) + \gamma e^{-\int_t^T r(s)ds}\rho(t)\right\} dt \\ &\quad + B(t)\left(\sigma(t)\sigma(t)^T\right)^{-1} \sigma(t)\left[\gamma e^{-\int_t^T r(s)ds} - x(t)\right] dW(t) \\ x(0) &= x_0 \end{cases} \tag{221}$$

Moreover, applying Itô's formula to $x^2(t)$, we obtain

$$\begin{cases} dx^2(t) &= \left\{(2r(t) - \rho(t))\, x^2(t) + \gamma^2 e^{-2\int_t^T r(s)ds}\rho(t)\right\} dt \\ &\quad + 2x(t)B(t)\left(\sigma(t)\sigma(t)^T\right)^{-1} \sigma(t)\left[\gamma e^{-\int_t^T r(s)ds} - x(t)\right] dW(t) \\ x^2(0) &= x_0^2 \end{cases} \tag{222}$$

Taking expectations on both sides of (221) and (222), we conclude that $\mathbb{E}[x(t)]$ and $\mathbb{E}\left[x^2(t)\right]$ satisfy the following two ordinary differential equations:

$$\begin{cases} d\mathbb{E}[x(t)] &= \left\{(r(t) - \rho(t))\, \mathbb{E}[x(t)] + \gamma e^{-\int_t^T r(s)ds}\rho(t)\right\} dt \\ \mathbb{E}[x(0)] &= x_0 \end{cases} \tag{223}$$

and

$$\begin{cases} d\mathbb{E}\left[x^2(t)\right] &= \left\{(2r(t)-\rho(t))\,\mathbb{E}\left[x^2(t)\right] + \gamma^2 e^{-2\int_t^T r(s)ds}\rho(t)\right\}dt \\ \mathbb{E}\left[x^2(0)\right] &= x_0^2 \end{cases} \tag{224}$$

Solving (223) and (224) we can express $\mathbb{E}\left[x(t)\right]$ and $\mathbb{E}\left[x^2(t)\right]$ as explicit functions of γ

$$\begin{cases} \mathbb{E}\left[x(T)\right] &= \alpha x_0 + \beta\gamma \\ \mathbb{E}\left[x^2(T)\right] &= \delta x_0^2 + \beta\gamma^2 \end{cases} \tag{225}$$

where

$$\begin{cases} \alpha &= e^{\int_0^T (r(t)-\rho(t))dt} \\ \beta &= 1 - e^{-\int_0^T \rho(t)dt} \\ \delta &= e^{\int_0^T (2r(t)-\rho(t))dt} \end{cases} \tag{226}$$

An optimal solution of the problem $P(\mu)$, if it exists, can be found by selecting λ^* such that (see (213))

$$\lambda^* = 1 + 2\mu\mathbb{E}\left[x^*(T)\right] = 1 + 2\mu\left(\alpha x_0 + \beta\frac{\lambda^*}{2\mu}\right)$$

This yields

$$\lambda^* = \frac{1+2\mu\alpha x_0}{1-\beta} = e^{\int_0^T \rho(t)dt} + 2\mu x_0 e^{\int_0^T r(t)dt} \tag{227}$$

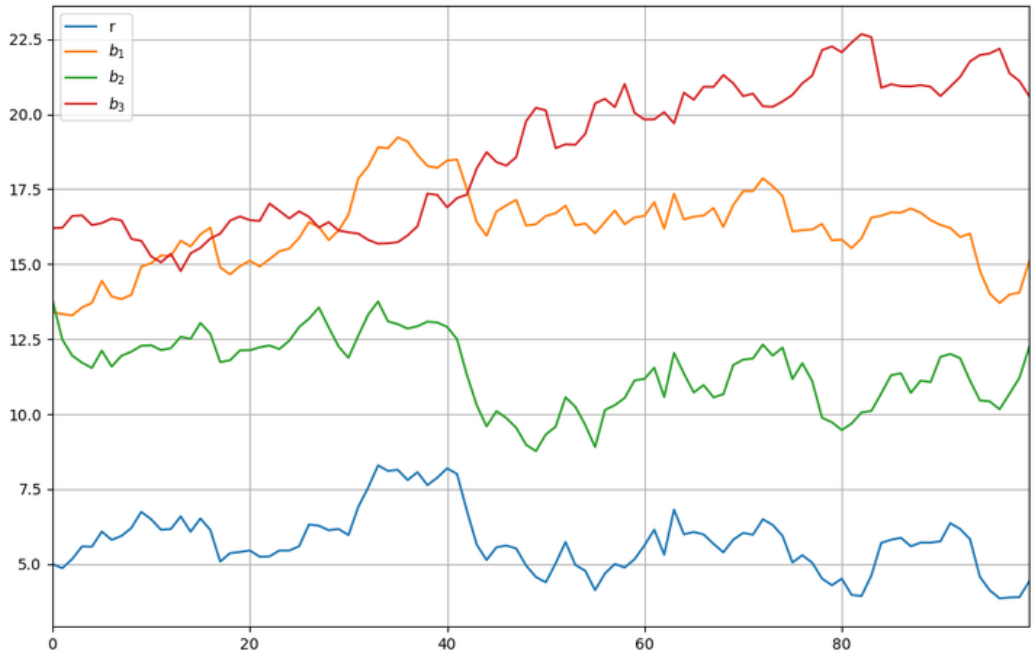
Hence the optimal control for the problem $P(\mu)$ (if it exists) must be given by (220) with $\gamma = \gamma^* = \frac{\lambda^*}{2\mu}$ and λ^* given by (227). In this case the corresponding variance of the terminal wealth is

$$\begin{aligned} \mathbb{V}\left[x^*(T)\right] &= \mathbb{E}\left[x^*(T)^2\right] - \mathbb{E}^2\left[x^*(T)\right] \\ &= \beta(1-\beta)\gamma^{*2} - 2\alpha\beta x_0\gamma^* + (\delta-\alpha^2)x_0^2 \\ &= \frac{1-\beta}{\beta}\left[\beta^2\gamma^{*2} - 2\frac{\alpha\beta^2 x_0\gamma^*}{1-\beta} + \frac{\beta(\delta-\alpha^2)}{1-\beta}x_0^2\right] \\ &= \frac{1-\beta}{\beta}\left[(\beta\gamma^* + \alpha x_0)^2 - 2\frac{\alpha\beta x_0\gamma^*}{1-\beta} + \frac{\beta\delta-\alpha^2}{1-\beta}x_0^2\right] \end{aligned} \tag{228}$$

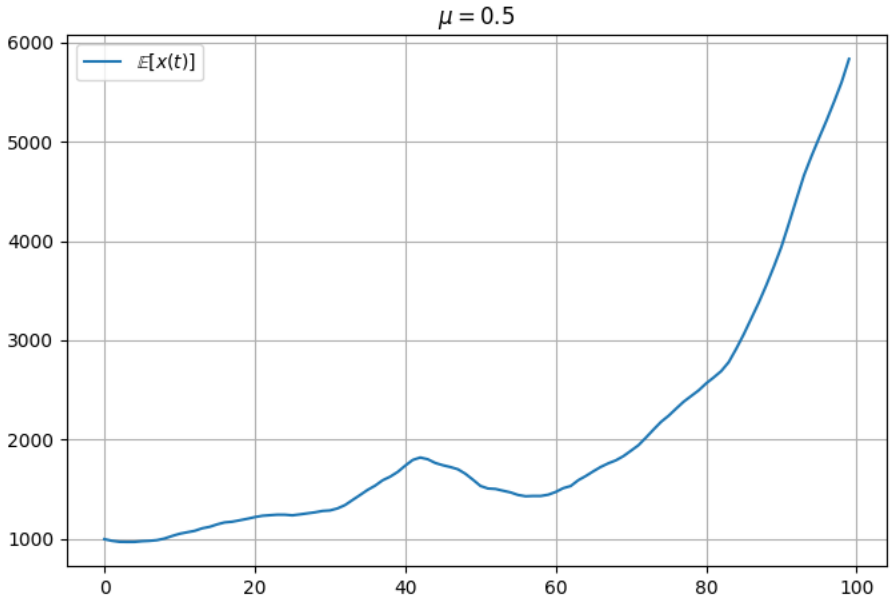
Substituting $\beta\gamma^* = \mathbb{E}\left[x(T)\right] - \alpha x_0$ in the above and noting (226) we obtain

$$\begin{aligned}\mathbb{V}\left[x^*(T)\right] &= \frac{1-\beta}{\beta}\left\{\left(\mathbb{E}\left[x^*(T)\right]\right)^2 - 2\frac{\alpha}{1-\beta}x_0\mathbb{E}\left[x^*(T)\right] + \frac{\beta\delta+\alpha^2}{1-\beta}x_0^2\right\}\\ &= \frac{1-\beta}{\beta}\left(\mathbb{E}\left[x^*(T)\right] - x_0 e^{\int_0^T r(t)dt}\right)^2\\ &= \frac{e^{-\int_0^T \rho(t)dt}}{1-e^{-\int_0^T \rho(t)dt}}\left(\mathbb{E}\left[x^*(T)\right] - x_0 e^{\int_0^T r(t)dt}\right)^2 \qquad (229)\end{aligned}$$

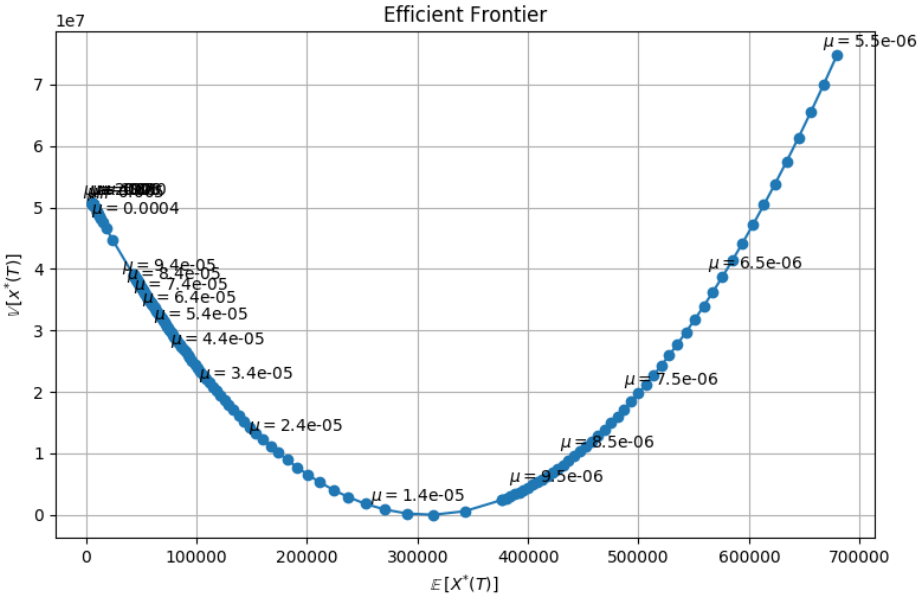
Here is an example for a bond interest rate and appreciation rates for 3 stocks.



Solving (223) numerically for $\mu = 0.5$ we obtain



and finally



Let's rewrite our cost (215) in the Mayer form:

$$
\begin{aligned}
J(t,y,u) &= \mathbb{E}\left[\frac{1}{2}\mu y^2(T)\right] = \frac{\mu}{2}y^2(0) + \frac{\mu}{2}\int_0^T \mathbb{E}\left[\frac{d}{dt}y^2(t)\right]dt \\
&= \frac{\mu}{2}y^2(0) + \frac{\mu}{2}\mathbb{E}\left[\int_0^T 2y(t)\frac{dy(t)}{dt}dt\right] \\
&= \frac{\mu}{2}y^2(0) + \mu\mathbb{E}\left[\int_0^T y(t)\left(Ay + \boldsymbol{B}\boldsymbol{u} + b + \boldsymbol{D}\boldsymbol{u}\frac{d\boldsymbol{W}}{dt}\right)dt\right] \\
&= \frac{\mu}{2}y^2(0) + \mu\int_0^T y(Ay + \boldsymbol{B}\boldsymbol{u} + b)dt + \mu\mathbb{E}\left[\int_0^T y\boldsymbol{D}\boldsymbol{u}d\boldsymbol{W}\right]
\end{aligned}
$$

The last term is an Itô integral and its expectation is 0. Therefore,

$$
J(t,y,u) = \frac{\mu}{2}y^2(0) + \mu\int_0^T y(Ay + \boldsymbol{B}\boldsymbol{u} + b)dt \tag{230}
$$

Then the stochastic HJB (197) becomes

$$
\begin{aligned}
-\frac{\partial V(t,y)}{\partial t} &= \inf_u\left\{\mu y(Ay + \boldsymbol{B}\boldsymbol{u} + b) + (Ay + \boldsymbol{B}\boldsymbol{u} + b)\frac{\partial V}{\partial y} + \frac{1}{2}(\boldsymbol{\Sigma}\cdot\boldsymbol{u})^2\frac{\partial^2 V}{\partial y^2}\right\} \\
&= \inf_u\left\{(Ay + \boldsymbol{B}\boldsymbol{u} + b)\left(\mu y + \frac{\partial V}{\partial y}\right) + \frac{1}{2}\boldsymbol{u}^T\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{u}\frac{\partial^2 V}{\partial y^2}\right\} \qquad (231)
\end{aligned}
$$

Minimizing the right side by u

$$
\left(\mu y + \frac{\partial V}{\partial y}\right)\boldsymbol{B} + \boldsymbol{u}^T\cdot\boldsymbol{\Sigma}^T\cdot\boldsymbol{\Sigma}\frac{\partial^2 V}{\partial y^2} = 0
$$

Hence

$$
\boldsymbol{u}^* = -\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\right)^{-1}\boldsymbol{B}^T\frac{\left(\mu y + \frac{\partial V}{\partial y}\right)}{\frac{\partial^2 V}{\partial y^2}} \tag{232}
$$

Since $\boldsymbol{\Sigma}$ is positive definite, this will only be valid at points where $\frac{\partial^2 V}{\partial y^2} > 0$ or at the boundary of possible $\boldsymbol{u}$ (u_j can change from 0% to 100% of the portfolio value).

Substituting this back into HJB (231) we obtain

$$
\begin{aligned}
-\frac{\partial V(t,y)}{\partial t} &= \left(Ay - \boldsymbol{B}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\right)^{-1}\boldsymbol{B}^T \frac{\left(\mu y + \frac{\partial V}{\partial y}\right)}{\frac{\partial^2 V}{\partial y^2}} + b \right)\left(\mu y + \frac{\partial V}{\partial y}\right) \\
&\quad - \frac{1}{2}\left(\boldsymbol{\Sigma}\cdot\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\right)^{-1}\boldsymbol{B}^T \frac{\left(\mu y + \frac{\partial V}{\partial y}\right)}{\frac{\partial^2 V}{\partial y^2}} \right)^2 \frac{\partial^2 V}{\partial y^2} \\
&= \left(Ay - \rho\frac{\left(\mu y + \frac{\partial V}{\partial y}\right)}{\frac{\partial^2 V}{\partial y^2}} + b \right)\left(\mu y + \frac{\partial V}{\partial y}\right) - \frac{1}{2}\boldsymbol{B}\boldsymbol{\Sigma}^{-1^T}\boldsymbol{\Sigma}^{-1}\boldsymbol{B}^T\left(\frac{\mu y + \frac{\partial V}{\partial y}}{\frac{\partial^2 V}{\partial y^2}}\right)^2 \frac{\partial^2 V}{\partial y^2} \\
&= \left(Ay - \rho\frac{\left(\mu y + \frac{\partial V}{\partial y}\right)}{\frac{\partial^2 V}{\partial y^2}} + b \right)\left(\mu y + \frac{\partial V}{\partial y}\right) - \frac{1}{2}\rho\frac{\left(\mu y + \frac{\partial V}{\partial y}\right)^2}{\frac{\partial^2 V}{\partial y^2}} \\
&= (Ay + b)\left(\mu y + \frac{\partial V}{\partial y}\right) - \frac{3}{2}\rho\frac{\left(\mu y + \frac{\partial V}{\partial y}\right)^2}{\frac{\partial^2 V}{\partial y^2}} \qquad (233)
\end{aligned}
$$

To solve this numerically we can use Crank-Nicolson scheme. We'll use the index i for time and j for y. Then

$$
\begin{aligned}
-\frac{V_{i+1,j} - V_{i,j}}{\Delta t} &= (A_i y_{i+1/2} + b_i)\left(\mu y_{i+1/2} + \frac{V_{i+1/2,j+1} - V_{i+1/2,j}}{\Delta y}\right) \\
&\quad - \frac{3}{2}\rho_i \frac{\left(\mu y_{i+1/2} + \frac{V_{i+1/2,j+1} - V_{i+1/2,j}}{\Delta y}\right)^2}{\frac{V_{i+1/2,j+1} - 2V_{i+1/2,j} + V_{i+1/2,j-1}}{(\Delta y)^2}} \\
&= \left(A_i\frac{y_{i+1} + y_i}{2} + b_i\right)\left(\mu\frac{y_{i+1} + y_i}{2} + \frac{V_{i+1,j+1} + V_{i,j+1} - V_{i+1,j} - V_{i,j}}{2\Delta y}\right) \\
&\quad - \frac{3}{4}\rho_i \frac{\left(\mu(y_{i+1} + y_i)\Delta y + V_{i+1,j+1} + V_{i,j+1} - V_{i+1,j} - V_{i,j}\right)^2}{V_{i+1,j+1} + V_{i,j+1} - 2V_{i+1,j} - 2V_{i,j} + V_{i+1,j-1} + V_{i,j-1}} \qquad (234)
\end{aligned}
$$

By definition $y \sim u_j$, therefore $\Delta y = \Delta u$. We start from $i = 0$ (i.e. at $t = 0$) and can use $u_j(0) = \frac{1}{m}x_0$. For a next step we change u_j to get Δy. Using $V = \mu y(Ay + \boldsymbol{B}\boldsymbol{u} + b)$ we can obtain ΔV (y is calculated using (216) without the stochastic term as we are interested in the expectation). Then we subtract the right side of (234) from V_i to obtain V_{i+1}. Once (233)/(234) is solved we can calculate V_y, V_{yy}, put it back into (232) to find $\boldsymbol{u}^*$, and finally y from (216).

▲

4.7 Optimal Dealer Pricing

We consider a single dealer trading a single stock.

Let's assume that the mid-price process $S(t)$ follows an Itô diffusion, i.e.

$$dS(t) = \mu\left(t, S(t)\right) dt + \sigma\left(t, S(t)\right) dW(t) \tag{235}$$

where $W(t)$ is a standard Brownian motion in a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{0 \le t \le T}, \mathbb{P})$.

We denote the bid and ask quotes as $S^-(t)$ and $S^+(t)$ respectively, and the market-maker's spreads: $\delta^+(t) := S^+(t) - S(t)$ and $\delta^-(t) := S(t) - S^-(t)$. We'll allow $\delta^\pm \in \mathbb{R}$ and interpret $\delta^\pm \le 0$ as market orders.

The inventory $Q(t) \in \mathbb{Z}$ process

$$dQ(t) = dN^-(t) - dN^+(t)$$

When the dealer sales (public purchases) at the ask price, the inventory changes as $dN^+(t)$ with intensity of process λ^+ resulting in the jump size of dN^+ (number of shares in a transaction), i.e. λ^+ is the average number of sell transactions per unit time, and $\lambda^+ dt$ is the probability of a dealer sale over the next instant. So $dQ/dt = \lambda dN$.

For simplicity, we assume a constant frequency Λ of market buy and sell orders. This could be estimated by dividing the total volume traded over a day by the average size of market orders on that day.

The distribution of the size of market orders has been found by several studies to obey a power law. In other words, the density of market order size is

$$f^Q(x) \sim x^{-1-\alpha}$$

for large x with $\alpha = 1.4 : 1.53$. There is less consensus on the statistics of the market impact in the econophysics literature. We'll use

$$\Delta p \sim \ln Q$$

Combining the two above equations we derive the Poisson intensity at which our agent's orders are executed

$$\begin{aligned}
\lambda(\delta) &= \Lambda P(\Delta p > \delta) \\
&= \Lambda P(\ln Q > K\delta) \\
&= \Lambda P(Q > e^{K\delta}) \\
&= \Lambda \int_{e^{K\delta}}^{\infty} x^{-1-\alpha} dx \\
&= A e^{-k\delta}
\end{aligned} \tag{236}$$

where $A = \Lambda/\alpha$ and $k = \alpha K$.

Alternatively, since we are interested in short term liquidity, the market impact function could be derived directly by integrating the density of the limit order book.

The cash $X(t) \in \mathbb{R}$ process

$$dX(t) = \left(S(t) + \delta^+\right) dN^+(t) - \left(S(t) - \delta^-\right) dN^-(t)$$

Thus, shares enter inventory at $S(t)$ whereas the associated cash flow in the cash account is $(S - \delta^-)$ or $(S + \delta^+)$ per share. The difference reflects the dealer's profit.

In this framework, the PnL or wealth of the market-maker is

$$PnL(t) = X(t) + Q(t)S(t) \in \mathbb{R}$$

Our goal is to derive the optimal bid and ask prices that maximize the dealer's expected utility of terminal wealth.

We will assume that the market-maker has a utility function $\phi(s, q, x)$ and an associated value function

$$u(t, s, q, x) = \max_{\delta^+, \delta^-} \left\{ \mathbb{E}\left[\phi\left(S(T), Q(T), X(T)\right)\right] \right\} \tag{237}$$

where $t \in [0, T]$ is the current time, $s = S(t)$ is the current mid-price of the asset, $x = X(t)$ is the current cash and $q = Q(t)$ is the current inventory level.

This is one period portfolio model where dealer costlessly liquidates all assets at the horizon date T.

Since there is no intermediate consumption prior to T, the fundamental recurrence relation implied by the principle of optimality of dynamic programming is simply that

$$\max_{\delta^+, \delta^-} \left\{ du(t, s, q, x) \right\} = 0 \tag{238}$$

The reservation bid price is the price that would make the agent indifferent between his current portfolio and his current portfolio plus one stock. This is a subjective valuation from the point of view of the agent and does not reflect a price at which trading should occur.

$$u(t, s, q + 1, x - r^b(t, s, q)) = u(t, s, q, x) \tag{239}$$

Since we, the dealer, establish the bids and asks, we set them such that

$$\delta^- = S - r^b$$

Analogously,

$$u(t, s, q - 1, x + r^a(t, s, q)) = u(t, s, q, x) \tag{240}$$

and

$$\delta^+ = r^a - S$$

We will refer to the average of these two prices as the reservation or indifference price:

$$r(t,s,q) = \frac{r^b + r^a}{2} \tag{241}$$

Writing out the partial differential equation implied by (238) – Bellman's equation – gives

$$\begin{aligned}
&\max_{\delta^+,\delta^-}\left\{\frac{du(t,s,q,x)}{dt}\right\} = \frac{\partial u}{\partial t} + \mathcal{L}u \\
&+ \max_{\delta^-}\left\{\lambda^-(\delta^-)\left[u\left(t,s,q+1,x-(s-\delta^-)\right) - u(t,s,q,x)\right]\right\} \\
&+ \max_{\delta^+}\left\{\lambda^+(\delta^+)\left[u\left(t,s,q-1,x+(s+\delta^+)\right) - u(t,s,q,x)\right]\right\} = 0 \\
&u(T,s,q,x) = \phi(s,q,x)
\end{aligned} \tag{242}$$

where $\mathcal{L}$ is defined as the operator that takes the differential with respect to time of the mean returns and Wiener risk components of the function u using Itô's Lemma. (Noting that $\mathbb{E}[dy] = d\mathbb{E}[y]$ and that $\mathbb{E}[XdW_X] = 0$ it follows that $d\mathbb{E}[y]/dt = f_t + \mathcal{L}f$ where $\mathcal{L}f = f_X r_X X + \frac{1}{2}\sigma_X^2 X^2 f_{XX}$.) Thus,

$$\mathcal{L} := \mu(t,s)\partial_s + \frac{1}{2}\sigma^2(t,s)\partial_{ss}$$

The philosophy is very simple: given the utility function we choose a possible form of the solution of the nonlinear PDE equation (i.e. we make an ansatz); we plug this ansatz into the equation and compute the (implicit) optimal controls; we then plug the (implicit) controls and separate the equation into several simpler ones, normally linear; we then compute explicitly the controls and the solution for the equation.

Feynman-Kac formula

$$\frac{\partial u}{\partial t}(x,t)+\mu(x,t)\frac{\partial u}{\partial x}(x,t)+\frac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2}(x,t)-V(x,t)u(x,t)+f(x,t)=0$$

defined for all $x\in\mathbb{R}$ and $t\in[0,T]$, subject to the terminal condition

$$u(x,T)=\psi(x)$$

where μ,σ,ψ,V,f are known functions, T is a parameter and $u:\mathbb{R}\times[0,T]\to\mathbb{R}$ is the unknown. Then the Feynman-Kac formula tells us that the solution can be written as a conditional expectation

$$u(x,t)=\mathbb{E}\left[\int_t^T e^{-\int_t^r V(X_\tau,\tau)d\tau}f(X_r,r)dr+e^{-\int_t^T V(X_\tau,\tau)d\tau}\psi(X_T)\middle|X_t=x\right]$$

under a probability measure such that X is an Itô process driven by the equation

$$dX=\mu(X,t)dt+\sigma(X,t)dW$$

where $W(t)$ is a Wiener process (also called Brownian motion), and the initial condition for $X(t)$ is $X(t)=x$.

Here are some heuristic interpretations of the HJB equation in conjunction with the Feynman-Kac formula.

First, suppose q and x fixed. Since the continuous variable s follows (235), from Feynman-Kac representation formula we have that if $(t,s)\mapsto u_C(t,s,q,x)$ satisfies

$$\begin{cases}(\partial_t+\mathcal{L})u_C=0\\ \mathcal{L}:=\mu(t,s)\partial_s+\frac{1}{2}\sigma^2(t,s)\partial_{ss}\\ u_C(T,s,q,x)=\phi(s,q,x)\end{cases}\tag{243}$$

then

$$u_C(t,s,q,x)=\mathbb{E}\left[\phi\left(S(T),q,x\right)\right]$$

Second, suppose now that s is fixed. The jump variables (q,x) are related via the arrival of market orders that hit the quotes of the market-maker. The value function of the jump $(q,x)\mapsto u_D(t,s,q,x)$ satisfies

$$\begin{cases}\partial_t u_D+\lambda^+(\delta^+)\left[u_D\left(t,s,q-1,x+(s+\delta^+)\right)-u_D(t,s,q,x)\right]\\ +\lambda^-(\delta^-)\left[u_D\left(t,s,q+1,x-(s-\delta^-)\right)-u_D(t,s,q,x)\right]=0\\ u(T,s,q,x)=\phi(s,q,x)\end{cases}\tag{244}$$

Now, let us put together the continuous and jump dynamics (243),(244). Suppose that the market-maker's spreads (δ^+, δ^-) are known (i.e. deterministic) then the value function $u(t, s, q, x)$ satisfies the following infinitesimal generator:

$$\begin{cases} (\partial_t + \mathcal{L})u + \lambda^+(\delta^+)\left[u\left(t, s, q-1, x+(s+\delta^+)\right) - u(t, s, q, x)\right] \\ +\lambda^-(\delta^-)\left[u\left(t, s, q+1, x-(s-\delta^-)\right) - u(t, s, q, x)\right] = 0 \\ u(T, s, q, x) = \phi(s, q, x) \end{cases} \tag{245}$$

Notice that (245) is valid only when the spreads (δ^+, δ^-) are known, but in the current case they are part of the set of unknowns of the problem. In consequence, from the stochastic control theory it follows that the value function $u(t, s, q, x)$ with unknown controls (δ^+, δ^-) is solution of the Hamilton-Jacobi-Bellman equation

$$\begin{cases} (\partial_t + \mathcal{L})u + \max_{\delta^+}\left\{\lambda^+(\delta^+)\left[u\left(t, s, q-1, x+(s+\delta^+)\right) - u(t, s, q, x)\right]\right\} \\ +\max_{\delta^-}\left\{\lambda^-(\delta^-)\left[u\left(t, s, q+1, x-(s-\delta^-)\right) - u(t, s, q, x)\right]\right\} = 0 \\ u(T, s, q, x) = \phi(s, q, x) \end{cases} \tag{246}$$

We used max instead of sup assuming that $\mu(t, s)$ and $\sigma(t, s)$ are Lipschitz and the jump dynamic is bounded.

Substituting (236) into (246) yields

$$\begin{cases} (\partial_t + \mathcal{L})u + \max_{\delta^+}\left\{Ae^{-k\delta^+}\left[u\left(t, s, q-1, x+(s+\delta^+)\right) - u(t, s, q, x)\right]\right\} \\ +\max_{\delta^-}\left\{Ae^{-k\delta^-}\left[u\left(t, s, q+1, x-(s-\delta^-)\right) - u(t, s, q, x)\right]\right\} = 0 \\ u(T, s, q, x) = \phi(s, q, x) \end{cases} \tag{247}$$

To solve it we'll make the following ansatz based on the utility function $\phi(s, q, x)$. For example, if

$$\phi(s, q, x) = x + \varphi(s, q)$$

we will use

$$u(t, s, q, x) = x + v(t, s, q) \tag{248}$$

We substitute (248) into (247) in order to find an easier HJB equation for v.

$$\begin{cases} (\partial_t + \mathcal{L})v + \max_{\delta^+}\left\{Ae^{-k\delta^+}\left[x+(s+\delta^+) + v(t, s, q-1) - x - v(t, s, q)\right]\right\} \\ +\max_{\delta^-}\left\{Ae^{-k\delta^-}\left[x-(s-\delta^-) + v(t, s, q+1) - x - v(t, s, q)\right]\right\} = 0 \\ v(T, s, q) = \varphi(s, q) \end{cases}$$

Hence

$$\begin{aligned}
&\max_{\delta^+}\left\{Ae^{-k\delta^+}\left[(s+\delta^+)+v(t,s,q-1)-v(t,s,q)\right]\right\}\\
\Leftrightarrow &-ke^{-k\delta^+}\left[s+\delta^++v(t,s,q-1)-v(t,s,q)\right]+e^{-k\delta^+}=0\\
&k\left[s+\delta^++v(t,s,q-1)-v(t,s,q)\right]=1
\end{aligned}$$

$$\delta_*^+=\frac{1}{k}-s-v(t,s,q-1)+v(t,s,q)$$

Analogously,

$$\begin{aligned}
&\max_{\delta^-}\left\{e^{-k\delta^-}\left[-(s-\delta^-)+v(t,s,q+1)-v(t,s,q)\right]\right\}\\
\Leftrightarrow &-ke^{-k\delta^-}\left[-s+\delta^-+v(t,s,q+1)-v(t,s,q)\right]+e^{-k\delta^-}=0\\
&k\left[-s+\delta^-+v(t,s,q+1)-v(t,s,q)\right]=1
\end{aligned}$$

$$\delta_*^-=\frac{1}{k}+s-v(t,s,q+1)+v(t,s,q)$$

We substitute the optimal controls back to the HJB to get the verification equation:

$$\begin{cases}(\partial_t+\mathcal{L})v+\frac{A}{k}\left(e^{-k\delta_*^+}+e^{-k\delta_*^-}\right)=0\\ v(T,s,q)=\varphi(s,q)\end{cases}$$

We solve it via the Feynman-Kac representation formula

$$v(t,s,q)=\mathbb{E}\left[\int_t^T\frac{A}{k}\left(e^{-k\delta_*^+}+e^{-k\delta_*^-}\right)d\tau+\varphi(s,q)\right]$$

Alternatively, we could express the optimal quotes in terms of the market-maker's bid-ask spread

$$\Delta_*:=\delta_*^++\delta_*^-$$

and the mid-point of the spread, the indifference price,

$$r_*(t,s,q):=\frac{1}{2}(r_*^++r_*^-)=s+\frac{1}{2}(\delta_*^+-\delta_*^-)$$

Notice that if $\delta_*^+=\delta_*^-$ then $r_*(t)=s=S(t)$. Therefore, r_*-s measures the level of asymmetry of the quotes with respect to the mid-price s.

Now, let us suppose that the utility function is linear, i.e.

$$\phi(s,q,x)=x+qs$$

Then the corresponding value function is

$$u(t,s,q,x) = \max_{\delta^+,\delta^-} \{\mathbb{E}\left[X(T) + Q(T)S(T)\right]\} \tag{249}$$

Alternatively, this corresponds to choosing the final condition as $\phi(s,q,x) = x + qs$. From this we will search a solution in the form of

$$u(t,s,q,x) = x + \theta_0(t,s) + q\theta_1(t,s) \tag{250}$$

Plugging (250)into (247) yields

$$\begin{cases} (\partial_t + \mathcal{L})(\theta_0 + q\theta_1) \\ + \max_{\delta^+} \left\{ Ae^{-k\delta^+} \left[x + s + \delta^+ + \theta_0(t,s) + (q-1)\theta_1(t,s) - x - \theta_0 - q\theta_1 \right] \right\} \\ + \max_{\delta^-} \left\{ Ae^{-k\delta^-} \left[x - s + \delta^- + \theta_0 + (q+1)\theta_1 - x - \theta_0 - q\theta_1 \right] \right\} = 0 \\ x + \theta_0(T,s) + q\theta_1(T,s) = x + qs \end{cases}$$

$$\begin{cases} (\partial_t + \mathcal{L})(\theta_0 + q\theta_1) + \max_{\delta^+} \left\{ Ae^{-k\delta^+} \left(s + \delta^+ - \theta_1 \right) \right\} \\ + \max_{\delta^-} \left\{ Ae^{-k\delta^-} \left(-s + \delta^- + \theta_1 \right) \right\} = 0 \\ \theta_0(T,s) = 0 \\ \theta_1(T,s) = s \end{cases}$$

We find that the maximum is attained at

$$\begin{aligned} & \max_{\delta^+} \left\{ Ae^{-k\delta^+} \left(s + \delta^+ - \theta_1 \right) \right\} \\ \Leftrightarrow & -ke^{-k\delta^+} \left(s + \delta^+ - \theta_1 \right) + e^{-k\delta^+} = 0 \\ & k(s + \delta^+ - \theta_1) = 1 \\ & \delta^+_* = \frac{1}{k} - s + \theta_1 \end{aligned}$$

Analogously,

$$\delta^-_* = \frac{1}{k} + s - \theta_1$$

In consequence, the optimal quotes (δ^+_*, δ^-_*), spread Δ_* and indifference price r_* are

$$\delta^\pm_* = \frac{1}{k} \pm (\theta_1 - s), \quad \Delta_* = \delta^+_* + \delta^-_* = \frac{2}{k}, \quad r_* = \theta_1$$

Back to HJB:

$$\begin{cases} (\partial_t + \mathcal{L})(\theta_0 + q\theta_1) + \frac{2A}{\epsilon k} \cosh\left[k(\theta_1 - s)\right] = 0 \\ \theta_0(T,s) = 0 \\ \theta_1(T,s) = s \end{cases} \tag{251}$$

We separate (251) in terms of the powers of q to obtain 2 coupled equations:

$$\begin{cases} (\partial_t + \mathcal{L})\theta_0 + \frac{2A}{ek}\cosh\left[k(\theta_1 - s)\right] = 0 \\ \theta_0(T,s) = 0 \end{cases} \tag{252}$$

and

$$\begin{cases} (\partial_t + \mathcal{L})\theta_1 = 0 \\ \theta_1(T,s) = s \end{cases} \tag{253}$$

Applying the Feynman-Kac formula to (253) and recursively to (252) yields, respectively

$$\begin{cases} \theta_1(t,s) & = \mathbb{E}\left[S(T)\right] \\ \theta_0(t,s) & = \frac{2A}{ek}\mathbb{E}\left[\int_t^T \cosh\left[k\left(\theta_1\left(\tau, S(\tau)\right) - S(\tau)\right)\right] d\tau\right] \end{cases}$$

In consequence,

$$\delta_*^{\pm} = \frac{1}{k} \pm \left(\mathbb{E}\left[S(T)\right] - s\right), \quad \Delta_* = \frac{2}{k}, \quad r_* = \mathbb{E}\left[S(T)\right] \tag{254}$$

and (250) becomes

$$u(t,s,q,x) = x + \frac{2A}{ek}\mathbb{E}\left[\int_t^T \cosh\left[k\left(\theta_1\left(\tau, S(\tau)\right) - S(\tau)\right)\right] d\tau\right] + q\mathbb{E}\left[S(T)\right] \tag{255}$$

In particular, since $\cosh(\alpha) \geq 1$ and $\cosh(\alpha) = 1 \Leftrightarrow \alpha = 0$ we obtain that

$$u(t,s,q,x) \geq \underline{u}(t,s,q,x) := x + \frac{2A}{ek}(T-t) + q\mathbb{E}\left[S(T)\right] \tag{256}$$

and $u(t,s,q,x) = \underline{u}(t,s,q,x) \Leftrightarrow \theta_1\left(\tau, S(\tau)\right) = S(\tau)\, \forall\, \tau \in [t,T]$. Since $t \in [0,T]$ is arbitrary then taking $\tau = t$ we have that

$$u(t,s,q,x) = \underline{u}(t,s,q,x) \Leftrightarrow \theta_1\left(t, S(t)\right) = S(t) = s = \mathbb{E}\left[S(T)\right] \tag{257}$$

i.e. if and only if $S(t)$ is a martingale, in which case we've got $r_*(t) = S(t), \quad \delta_*^{\pm} = \frac{1}{k} \quad and \quad u(t,s,q,x) = x + qs$.

It is interesting to note that applying perturbation methods on the variable q of the form

$$u(t,s,q,x) = x + \theta_0(t,s) + \theta_1(t,s)q + \theta_2(t,s)q^2 + \dots$$

is a very rough approximation, to say the least. Indeed, q is an integer, i.e. discrete and not small, and as such a perturbation method on q cannot be performed. However, once the ansatz is shown to solve the verification equation,

then by uniqueness it coincides with the solution of the original problem. Therefore, the separation of the equation into two terms, one with q^0 and another with q^1, is justified *a posteriori* via the maximum principle (i.e. existence and uniqueness) for the Hamilton-Jacobi-Bellman equation, and as such it does not rely at all on perturbation methods.

In applying the Feynman-Kac formula we implicitly assumed that the value function $u(t,s,q,x)$ is finite. However, this is valid if and only if

$$\mathbb{E}\left[\int_t^T \cosh\left[k\left(\theta_1\left(\tau, S(\tau)\right) - S(\tau)\right)\right] d\tau\right] < \infty, \quad \theta_1(t,s) = \mathbb{E}\left[S(T)\right] \tag{258}$$

If $S(t)$ is a martingale then the above holds trivially. For a non-martingale mid-price process $S(t)$, 2 sufficient conditions for (258) to hold are (i) the conditional expectation $\mathbb{E}\left[S(T)\right]$ is affine on s and (ii) the moment-generating function $\mathbb{E}\left[e^{\lambda S(t)}\right]$ is finite for all $\lambda \in \mathbb{R}$. This is the case for any Gaussian Markov process, e.g. an arithmetic Brownian motion with drift and the Ornstein-Uhlenbeck process. However, (258) does not hold for the geometric Brownian motion with drift.

Let us consider another case when the utility function is exponential

$$\phi(s,q,x) = -e^{-\gamma(x+qs)}$$

whose corresponding value function is

$$u(t,s,q,x) = \max_{\delta^+,\delta^-}\left\{\mathbb{E}\left[-e^{-\gamma(X(T)+Q(T)S(T))}\right]\right\} \tag{259}$$

Ansatz: from the form of the utility function we will search a solution of the form

$$\begin{cases} u(t,s,q,x) = -e^{-\gamma(x+\theta(t,s,q))} \\ \theta(t,s,q) = \theta_0(t) + \theta_1(t,s)q + \theta_2(t)q^2 \end{cases} \tag{260}$$

Plugging (260) into (247) yields the Hamilton-Jacobi-Bellman for $\theta(t,s,q)$, i.e.

$$\begin{cases} -(\partial_t + \mathcal{L})e^{-\gamma(x+\theta)} \\ + \max_{\delta^+}\left\{Ae^{-k\delta^+}\left[-e^{-\gamma(x+s+\delta^+ + \theta_0+\theta_1 q - \theta_1 + \theta_2 q^2 - 2\theta_2 q + \theta_2)} + e^{-\gamma(x+\theta_0+\theta_1 q+\theta_2 q^2)}\right]\right\} \\ + \max_{\delta^-}\left\{Ae^{-k\delta^-}\left[-e^{-\gamma(x-s+\delta^- + \theta_0+\theta_1 q + \theta_1 + \theta_2 q^2 + 2\theta_2 q + \theta_2)} + e^{-\gamma(x+\theta_0+\theta_1 q+\theta_2 q^2)}\right]\right\} = 0 \\ -e^{-\gamma(X(T)+\theta(T,s,q))} = -e^{-\gamma(x+qs)} \end{cases}$$

$$\begin{cases} \gamma e^{-\gamma(x+\theta)}\theta_t + \mu\gamma e^{-\gamma(x+\theta)}\theta_s - \frac{1}{2}\sigma^2\gamma\left[\gamma e^{-\gamma(x+\theta)}\left(\theta_s\right)^2 - e^{-\gamma(x+\theta)}\theta_{ss}\right] \\ +A\max_{\delta^+}\left\{e^{-k\delta^+}e^{-\gamma(x+\theta)}\left(1-e^{-\gamma\left(s+\delta^+-\theta_1+(1-2q)\theta_2\right)}\right)\right\} \\ +A\max_{\delta^-}\left\{e^{-k\delta^-}e^{-\gamma(x+\theta)}\left(1-e^{-\gamma\left(-s+\delta^-+\theta_1+(1+2q)\theta_2\right)}\right)\right\} = 0 \\ \theta(T,s,q) = qs \end{cases}$$

$$\begin{cases} \theta_t + \mu\theta_s - \frac{1}{2}\sigma^2\left[\gamma\left(\theta_s\right)^2 - \theta_{ss}\right] + \frac{A}{\gamma}\max_{\delta^+}\left\{e^{-k\delta^+}\left(1-e^{-\gamma\left(s+\delta^+-\theta_1+(1-2q)\theta_2\right)}\right)\right\} \\ +\frac{A}{\gamma}\max_{\delta^-}\left\{e^{-k\delta^-}\left(1-e^{-\gamma\left(-s+\delta^-+\theta_1+(1+2q)\theta_2\right)}\right)\right\} = 0 \\ \theta(T,s,q) = qs \end{cases}$$

$$\begin{aligned} &\max_{\delta^+}\left\{e^{-k\delta^+}\left(1-e^{-\gamma\left(s+\delta^+-\theta_1+(1-2q)\theta_2\right)}\right)\right\} \\ \Leftrightarrow &-ke^{-k\delta_*^+}\left(1-e^{-\gamma\left(s+\delta_*^+-\theta_1+(1-2q)\theta_2\right)}\right) + \gamma e^{-k\delta_*^+}e^{-\gamma\left(s+\delta_*^+-\theta_1+(1-2q)\theta_2\right)} = 0 \\ &-k + e^{-\gamma\left(s+\delta_*^+-\theta_1+(1-2q)\theta_2\right)}(k+\gamma) = 0 \\ &-\gamma\left(s+\delta_*^+-\theta_1+(1-2q)\theta_2\right) = \ln\frac{k}{k+\gamma} \\ &\delta_*^+ = \frac{1}{\gamma}\ln\frac{k+\gamma}{k} - s + \theta_1 - (1-2q)\theta_2 \\ &\delta_*^+ = \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) - s + \theta_1 - (1-2q)\theta_2 \end{aligned} \tag{261}$$

Analogously,

$$\delta_*^- = \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + s - \theta_1 - (1+2q)\theta_2 \tag{262}$$

In consequence,

$$\delta_*^\pm = \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)\mp s\pm\theta_1-(1\mp 2q)\theta_2, \quad \Delta_* = \frac{2}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)-2\theta_2, \quad r_* = \theta_1+2q\theta_2 \tag{263}$$

Plugging (261) and (262) back into the HJB

$$\theta_t + \mu\theta_s - \frac{1}{2}\sigma^2\left[\gamma\left(\theta_s\right)^2 - \theta_{ss}\right] + \frac{A}{\gamma}e^{-k\delta_*^+}\left(1 - e^{-\gamma\left(s+\delta_*^+-\theta_1+(1-2q)\theta_2\right)}\right)$$
$$+ \frac{A}{\gamma}e^{-k\delta_*^-}\left(1 - e^{-\gamma\left(-s+\delta_*^-+\theta_1+(1+2q)\theta_2\right)}\right) = 0$$
$$\theta_t + \mu\theta_s - \frac{1}{2}\sigma^2\left[\gamma\left(\theta_s\right)^2 - \theta_{ss}\right] + \frac{A}{\gamma}e^{-k\delta_*^+}\left(1 - e^{-\gamma\left(s+\frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)-s+\theta_1-(1-2q)\theta_2-\theta_1+(1-2q)\theta_2\right)}\right)$$
$$+ \frac{A}{\gamma}e^{-k\delta_*^-}\left(1 - e^{-\gamma\left(-s+\frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)+s-\theta_1-(1+2q)\theta_2+\theta_1+(1+2q)\theta_2\right)}\right) = 0$$
$$\theta_t + \mu\theta_s - \frac{1}{2}\sigma^2\left[\gamma\left(\theta_s\right)^2 - \theta_{ss}\right] + \frac{A}{k+\gamma}e^{-k\delta_*^+} + \frac{A}{k+\gamma}e^{-k\delta_*^-} = 0$$

The first-order Taylor expansion for the last two terms is

$$\frac{A}{k+\gamma}\left(1 - k\delta_*^+ + 1 - k\delta_*^-\right) = \frac{A}{k+\gamma}\left(2 - k(\delta_*^+ + \delta_*^-)\right)$$
$$= \frac{2A}{k+\gamma}\left(1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + k\theta_2\right)$$

and therefore

$$\theta_t + \mu\theta_s - \frac{1}{2}\sigma^2\left[\gamma\left(\theta_s\right)^2 - \theta_{ss}\right] + \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + k\theta_2\right] = 0$$

substituting θ from (260)

$$\partial_t(\theta_0 + \theta_1 q + \theta_2 q^2) + \mu\partial_s(\theta_0 + \theta_1 q + \theta_2 q^2) - \frac{1}{2}\sigma^2\left[\gamma\left(\partial_s(\theta_0 + \theta_1 q + \theta_2 q^2)\right)^2 - \partial_{ss}(\theta_0 + \theta_1 q + \theta_2 q^2)\right]$$
$$+ \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + k\theta_2\right] = 0$$
$$\partial_t(\theta_0 + \theta_1 q + \theta_2 q^2) + \mu\partial_s(\theta_0 + \theta_1 q + \theta_2 q^2)$$
$$- \frac{1}{2}\sigma^2\gamma\left[(\theta_0')^2 + 2\theta_0'\theta_1' q + (2\theta_0'\theta_2' + (\theta_1')^2)q^2\right]$$
$$+ \frac{1}{2}\sigma^2\partial_{ss}(\theta_0 + \theta_1 q + \theta_2 q^2)$$
$$+ \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + k\theta_2\right] = 0$$

where the prime indicates the partial derivative with respect to s, and we omitted the powers of $q > 2$.

We separate in terms of the powers of q, noting that only θ_1 depends on s, and obtain three coupled equations

$$\begin{cases} q^0: & \partial_t\theta_0 + \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + k\theta_2\right] = 0 \\ & \theta_0(T) = 0 \\ q^1: & \partial_t\theta_1 + \mu\partial_s\theta_1 + \frac{1}{2}\sigma^2\partial_{ss}\theta_1 = 0 \\ & \theta_1(T,s) = s \\ q^2: & \partial_t\theta_2 - \frac{1}{2}\sigma^2\gamma(\partial_s\theta_1)^2 = 0 \\ & \theta_2(T) = 0 \end{cases} \tag{264}$$

Applying the Feynman-Kac formula to the middle equation we find

$$\theta_1(t,s) = \mathbb{E}\left[S(T)\right] \tag{265}$$

Integrating the equation for q^2 from (264)(c) we obtain

$$\theta_2\Big|_t^T = \frac{1}{2}\gamma\int_t^T \sigma^2(\tau, S(\tau))\left[\partial_s\theta_1(\tau, S(\tau))\right]^2 d\tau$$

$$-\theta_2(t) = \frac{1}{2}\gamma\int_t^T \sigma^2(\tau, S(\tau))\left[\partial_s\theta_1(\tau, S(\tau))\right]^2 d\tau$$

However, the ansatz we have made implies that θ_2 is independent of s. Therefore, in order to solve (264)(c) we need to assume the following conditions on the price process $S(t)$:

$$\sigma = \sigma(t), \quad \mathbb{E}\left[S(T)\right] = \alpha(t,T) + \beta(t,T)s$$

Consequently,

$$\theta_2(t) = -\frac{1}{2}\gamma\int_t^T \sigma^2(\tau)\beta^2(\tau,T)d\tau \tag{266}$$

Finally, integrating (264)(a)

$$\begin{aligned}\theta_0(t) &= -\int_t^T \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + k\theta_2\right]d\zeta \\ &= \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)\right](T-t) - \frac{Ak\gamma}{k+\gamma}\int_t^T\int_\zeta^T \sigma^2(\tau)\beta^2(\tau,T)d\tau d\zeta\end{aligned} \tag{267}$$

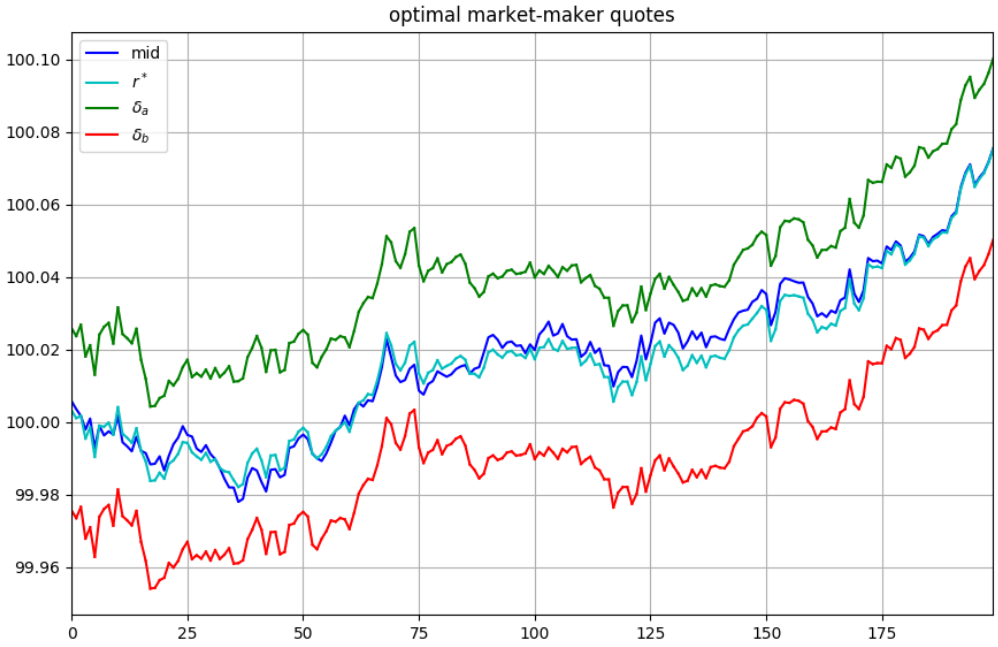
For a numerical simulation we use $\theta_1(t,s) = S(t)$ in (265), $\sigma = const$, $\alpha = 0$, $\beta = 1$ in (266), so $\theta_2(t) = -\frac{1}{2}\gamma\sigma^2(T-t)$, and from (267)

$$\theta_0(t) = \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)\right](T-t) - \frac{Ak\gamma}{k+\gamma}\sigma^2 \int_t^T \int_\zeta^T d\tau d\zeta$$
$$= \frac{2A}{k+\gamma}\left[1 - \frac{k}{\gamma}\ln\left(1+\frac{\gamma}{k}\right)\right](T-t) - \frac{Ak\gamma}{k+\gamma}\frac{\sigma^2}{2}(T-t)^2$$

Therefore, by (263)

$$\delta_*^+ = \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + (1-2q)\frac{1}{2}\gamma\sigma^2(T-t)$$
$$with\, q = 1$$
$$\delta_*^- = \frac{1}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + (1+2q)\frac{1}{2}\gamma\sigma^2(T-t)$$
$$\Delta_* = \frac{2}{\gamma}\ln\left(1+\frac{\gamma}{k}\right) + \gamma\sigma^2(T-t)$$
$$r_* = s - q\gamma\sigma^2(T-t)$$

This price r_* is an adjustment to the mid-price, which accounts for the inventory held by the agent. If the agent is long stock ($q > 0$), the reservation price is below the mid-price, indicating a desire to liquidate the inventory by selling stock. On the other hand, if the agent is short stock ($q < 0$), the reservation price is above the mid-price, since the agent is willing to buy stock at a higher price.



parameter	value
T	1 (day)
dt	0.005
σ	0.05
k	40
γ	0.1

5 Clustering

5.1 Soft K-means and Gaussian Mixture Model clustering

Suppose we have a data set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ consisting of N observations of a random D-dimensional Euclidean variable $\boldsymbol{x}$. Our goal is to partition the data set into some number K of clusters. Let $\boldsymbol{\mu}_i$ represent the center of a cluster. We introduce a corresponding set of binary indicator variables $r_{nk} \in \{0, 1\}$, where $k = 1, \ldots, K$ describe which of the K clusters the data point $\boldsymbol{x}_n$ is assigned to. We then define an objective function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \tag{268}$$

Our goal is to find values for the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$ so as to minimize J. We can do this through an iterative procedure in which each iteration involves

two successive steps corresponding to successive optimizations with respect to the r_{nk} and the $\boldsymbol{\mu}_k$. First we choose some initial values for the $\boldsymbol{\mu}_k$. Then in the first phase we minimize J with respect to the r_{nk}, keeping the $\boldsymbol{\mu}_k$ fixed. In the second phase we minimize J with respect to the $\boldsymbol{\mu}_k$, keeping r_{nk} fixed. This two-stage optimization is then repeated until convergence. These two stages of updating r_{nk} and updating $\boldsymbol{\mu}_k$ correspond respectively to the E (expectation) and M (maximization) steps of the EM algorithm.

Consider first the determination of the r_{nk}. Because J in (268) is a linear function of r_{nk}, this optimization can be performed easily to give a closed form solution. The terms involving different n are independent and so we can optimize for each n separately by choosing r_{nk} to be 1 for whichever value of k gives the minimum value of $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$. In other words, we simply assign the n^{th} data point to the closest cluster center. More formally, this can be expressed as

$$r_{nk} = \begin{cases} 1 & if\ k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & otherwise \end{cases} \tag{269}$$

Now consider the optimization of the $\boldsymbol{\mu}_k$ with the r_{nk} held fixed. The objective function J is a quadratic function of $\boldsymbol{\mu}_k$, and it can be minimized by setting its derivative with respect to $\boldsymbol{\mu}_k$ to zero giving

$$2\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

which we can easily solve for $\boldsymbol{\mu}_k$ to give

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}} \tag{270}$$

The denominator in this expression is equal to the number of points assigned to cluster k, and so this result has a simple interpretation, namely set $\boldsymbol{\mu}_k$ equal to the mean of all of the data points $\boldsymbol{x}_n$ assigned to cluster k. For this reason, the procedure is known as the K-means algorithm.

The two phases of re-assigning data points to clusters and re-computing the cluster means are repeated in turn until there is no further change in the assignments (or until some maximum number of iterations is exceeded). Because each phase reduces the value of the objective function J, convergence of the algorithm is assured. However, it may converge to a local rather than global minimum of J.

In practice, a better initialization procedure would be to choose the cluster centers $\boldsymbol{\mu}_k$to be equal to a random subset of K data points. It is also worth noting that the K-means algorithm itself is often used to initialize the parameters in a Gaussian mixture model before applying the EM algorithm.

A direct implementation of the K-means algorithm as discussed here can be relatively slow, because in each E step it is necessary to compute the Euclidean distance between every prototype vector and every data point. Various schemes have been proposed for speeding up the K-means algorithm, some of which are

based on precomputing a data structure such as a tree such that nearby points are in the same subtree (Ramasubramanian and Paliwal, 1990; Moore, 2000). Other approaches make use of the triangle inequality for distances, thereby avoiding unnecessary distance calculations (Hodgson, 1998; Elkan, 2003).

So far, we have considered a batch version of K-means in which the whole data set is used together to update the prototype vectors. We can also derive an on-line stochastic algorithm (MacQueen, 1967) by applying the Robbins-Monro procedure to the problem of finding the roots of the regression function given by the derivatives of J in (268) with respect to $\boldsymbol{\mu}_k$. This leads to a sequential update in which, for each data point $\boldsymbol{x}_n$ in turn, we update the nearest prototype $\boldsymbol{\mu}_k$ using

$$\boldsymbol{\mu}_k^{new} = \boldsymbol{\mu}_k^{old} + \eta_n(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{old}) \tag{271}$$

where η_n is the learning rate parameter, which is typically made to decrease monotonically as more data points are considered.

The Gaussian mixture distribution can be written as a linear superposition of Gaussians in the form

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{272}$$

Let us introduce a K-dimensional binary random variable $\boldsymbol{z}$ having a 1-of-K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, and we see that there are K possible states for the vector $\boldsymbol{z}$ according to which element is nonzero. We shall define the joint distribution $p(\boldsymbol{x}, \boldsymbol{z})$ in terms of a marginal distribution $p(\boldsymbol{z})$ and a conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$. The marginal distribution over $\boldsymbol{z}$ is specified in terms of the mixing coefficients π_k, such that

$$p(z_k = 1) = \pi_k$$

where the parameters $\{\pi_k\}$ must satisfy

$$0 \leq \pi_k \leq 1$$

together with

$$\sum_{k=1}^{K} \pi_k = 1$$

in order to be valid probabilities. Because $\boldsymbol{z}$ uses a 1-of-K representation, we can also write this distribution in the form

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Similarly, the conditional distribution of $\boldsymbol{x}$ given a particular value for $\boldsymbol{z}$ is a Gaussian

$$p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

which can also be written in the form

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

The joint distribution is given by $p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$, and the marginal distribution of $\boldsymbol{x}$ is then obtained by summing the joint distribution over all possible states of $\boldsymbol{z}$ to give

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{273}$$

Thus the marginal distribution of $\boldsymbol{x}$ is a Gaussian mixture of the form (272). If we have several observations $x_1, \ldots, x_N$, then, because we have represented the marginal distribution in the form $p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z})$, it follows that for every observed data point x_n there is a corresponding latent variable z_n.

Another quantity that will play an important role is the conditional probability of $\boldsymbol{z}$ given $\boldsymbol{x}$. We shall use $\gamma(z_k)$ to denote $p(z_k = 1|\boldsymbol{x})$, whose value can be found using Bayes' theorem

$$\begin{aligned} \gamma(z_k) \equiv p(z_k = 1|\boldsymbol{x}) &= \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \end{aligned} \tag{274}$$

We shall view π_k as the prior probability of $z_k = 1$, and the quantity $\gamma(z_k)$ as the corresponding posterior probability once we have observed $\boldsymbol{x}$. We can also interpret $\gamma(z_k)$ as the responsibility that the component k takes for"explaining" the observation $\boldsymbol{x}$.

We can use the technique of ancestral sampling to generate random samples distributed according to the Gaussian mixture model. To do this, we first generate a value for $\boldsymbol{z}$, which we denote $\hat{\boldsymbol{z}}$, from the marginal distribution $p(\boldsymbol{z})$ and then generate a value for $\boldsymbol{x}$ from the conditional distribution $p(\boldsymbol{x}|\hat{\boldsymbol{z}})$. We can depict samples from the joint distribution $p(\boldsymbol{x}, \boldsymbol{z})$ by plotting points at the corresponding values of $\boldsymbol{x}$ and then coloring them according to the value of $\boldsymbol{z}$. We can also use the posterior probability and color according to $\gamma(z_{nk})$ with proportions of the main k colors.

We can represent our data set as $N \times D$ matrix $\boldsymbol{X}$ in which the n^{th} row is given by $\boldsymbol{x}_n^T$. Similarly, the corresponding latent variables will be denoted by an $N \times K$ matrix $\boldsymbol{Z}$ with rows $\boldsymbol{z}_n^T$. If we assume that the data points are drawn

independently from the distribution, then the log of the likelihood function is given by

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{275}$$

The maximization of this function is not a well posed problem because often a cluster may contain one point coinciding with its center $\boldsymbol{\mu}_k$ at which the variance in the denominator collapses to zero.

Let us write down the conditions that must be satisfied at a maximum of the likelihood function. Setting the derivatives of $\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in (275) with respect to the means $\boldsymbol{\mu}_k$ of the Gaussian components to zero, we obtain

$$\begin{aligned} 0 &= -\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \\ &= -\sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{\Sigma}_k (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \end{aligned}$$

Note that the posterior probabilities, or responsibilities, given by (274) appear naturally on the right-hand side. Multiplying by Σ_k^{-1} (which we assume to be nonsingular) and rearranging we obtain

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \tag{276}$$

where we have defined

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

We can interpret N_k as the effective number of points assigned to cluster k. We see that the mean $\boldsymbol{\mu}_k$ for the k^{th} Gaussian component is obtained by taking a weighted mean of all of the points in the data set, in which the weighting factor for data point $\boldsymbol{x}_n$ is given by the posterior probability $\gamma(z_{nk})$ that component k was responsible for generating $\boldsymbol{x}_n$.

If we set the derivative of $\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\Sigma}_k$ to zero, and follow a similar line of reasoning, we obtain

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \tag{277}$$

Finally, we maximize $\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to the mixing coefficients π_k. Here we must take account of the constraint which requires the mixing coefficients to sum to one. This can be achieved using a Lagrange multiplier and maximizing the following quantity

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

which gives

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

If we now multiply both sides by π_k and sum over k making use of the constraint, we find $\lambda = -N$. Using this to eliminate λ and rearranging we obtain

$$\pi_k = \frac{N_k}{N} \tag{278}$$

so that the mixing coefficient for the k^{th} component is given by the average responsibility which that component takes for explaining the data points.

It is worth emphasizing that the results (276), (277), and (278) do not constitute a closed-form solution for the parameters of the mixture model because the responsibilities $\gamma(z_{nk})$ depend on those parameters in a complex way through (274). However, these results do suggest a simple iterative scheme for finding a solution to the maximum likelihood problem, which as we shall see turns out to be an instance of the EM algorithm for the particular case of the Gaussian mixture model. We first choose some initial values for the means, covariances, and mixing coefficients. Then we alternate between the following two updates that we shall call the E step and the M step. In the expectation step we use the current values for the parameters to evaluate the posterior probabilities, or responsibilities, given by (274). We then use these probabilities in the maximization step to re-estimate the means, covariances, and mixing coefficients using the results (276), (277), and (278). Note that in so doing we first evaluate the new means using (276) and then use these new values to find the covariances using (277), in keeping with the corresponding result for a single Gaussian distribution.

Note that the EM algorithm takes many more iterations to reach (approximate) convergence compared with the K-means algorithm, and that each cycle requires significantly more computation. It is therefore common to run the K-means algorithm in order to find a suitable initialization for a Gaussian mixture model that is subsequently adapted using EM. The covariance matrices can conveniently be initialized to the sample covariances of the clusters found by the K-means algorithm, and the mixing coefficients can be set to the fractions of data points assigned to the respective clusters. As with gradient-based approaches for maximizing the log likelihood, techniques must be employed to avoid singularities of the likelihood function in which a Gaussian component collapses onto a particular data point. It should be emphasized that there will generally be multiple local maxima of the log likelihood function, and that EM is not guaranteed to find the largest of these maxima.

EM algorithm for Gaussian mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

   where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

4. Evaluate the log likelihood

$$\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

   and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

To generalize, we denote the set of all model parameters as $\boldsymbol{\theta}$. Then the log likelihood function is given by

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln \left[\sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \right] \tag{279}$$

Because we cannot observe the complete data set $\{\boldsymbol{X}, \boldsymbol{Z}\}$, only the incomplete data $\boldsymbol{X}$, our state of knowledge of the values of the latent variables in $\boldsymbol{Z}$ is given only by the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})$. Therefore we cannot use the complete-data log likelihood, and we consider instead its expected value

under the posterior distribution of the latent variable, which corresponds to the E step. In the subsequent M step, we maximize this expectation.

In the E step, we use the current parameter values $\boldsymbol{\theta}^{old}$ to find the posterior distribution of the latent variables given by $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old})$. We then use this posterior distribution to find the expectation of the complete-data log likelihood evaluated for some general parameter value $\boldsymbol{\theta}$. This expectation, denoted $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$, is given by

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \tag{280}$$

In the M step, we determine the revised parameter estimate $\boldsymbol{\theta}^{new}$ by maximizing this function

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) \tag{281}$$

The General EM Algorithm

Given a joint distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$ over observed variables $\boldsymbol{X}$ and latent variables $\boldsymbol{Z}$, governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\boldsymbol{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{old}$.
2. **E step**. Evaluate $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old})$.
3. **M step**. Evaluate $\boldsymbol{\theta}^{new}$ given by (281) and (280).
4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

$$\boldsymbol{\theta}^{old} \leftarrow \boldsymbol{\theta}^{new}$$

and return to step 2.

The EM algorithm can also be used to find MAP (maximum posterior) solutions for models in which a prior $p(\boldsymbol{\theta})$ is defined over the parameters. In this case the E step remains the same as in the maximum likelihood case, whereas in the M step the quantity to be maximized is given by $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) + \ln p(\boldsymbol{\theta})$.

Here we have considered the use of the EM algorithm to maximize a likelihood function when there are discrete latent variables. However, it can also be applied when the unobserved variables correspond to missing values in the data set. The distribution of the observed values is obtained by taking the joint distribution of all the variables and then marginalizing over the missing ones. EM can then be used to maximize the corresponding likelihood function. An example of the application of this technique would be in the context of principal component analysis. This will be a valid procedure if the data values are missing

at random, meaning that the mechanism causing values to be missing does not depend on the unobserved values, which is often not the case.

Note that if we tend the covariance matrix $\mathbf{\Sigma}$ to zero then in the limit $\gamma(z_{nk}) \to r_{nk}$ and we come to the regular K-means algorithm.

For many models of practical interest, it will be infeasible to evaluate the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X})$ or indeed to compute expectations with respect to this distribution. Therefore, we need to resort to approximation schemes, and these fall broadly into two classes, according to whether they rely on stochastic or deterministic approximations. Stochastic techniques such as Markov chain Monte Carlo have enabled the widespread use of Bayesian methods across many domains. They generally have the property that given infinite computational resource, they can generate exact results, and the approximation arises from the use of a finite amount of processor time. In practice, sampling methods can be computationally demanding, often limiting their use to small-scale problems. Also, it can be difficult to know whether a sampling scheme is generating independent samples from the required distribution.

Deterministic approximation schemes are based on analytical approximations to the posterior distribution, for example by assuming that it factorizes in a particular way or that it has a specific parametric form such as a Gaussian. As such, they can never generate exact results, and so their strengths and weaknesses are complementary to those of sampling methods.

Suppose we partition the elements of $\boldsymbol{Z}$ into disjoint groups that we denote by $\boldsymbol{Z}_i$ where $i = 1, \ldots, M$. We then assume that the q distribution factorizes with respect to these groups, so that

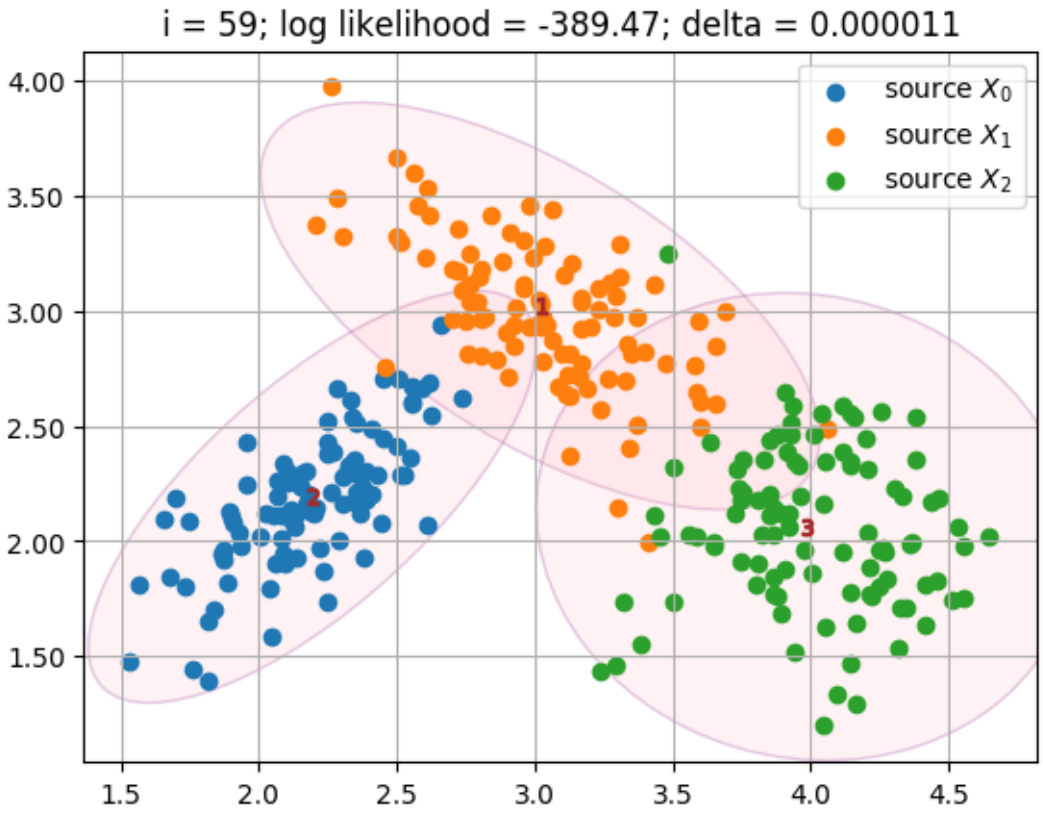
$$q(\boldsymbol{Z}) = \prod_{i=1}^{M} q_i(\boldsymbol{Z}_i)$$

This factorized form of variational inference corresponds to an approximation framework developed in physics called *mean field theory*.

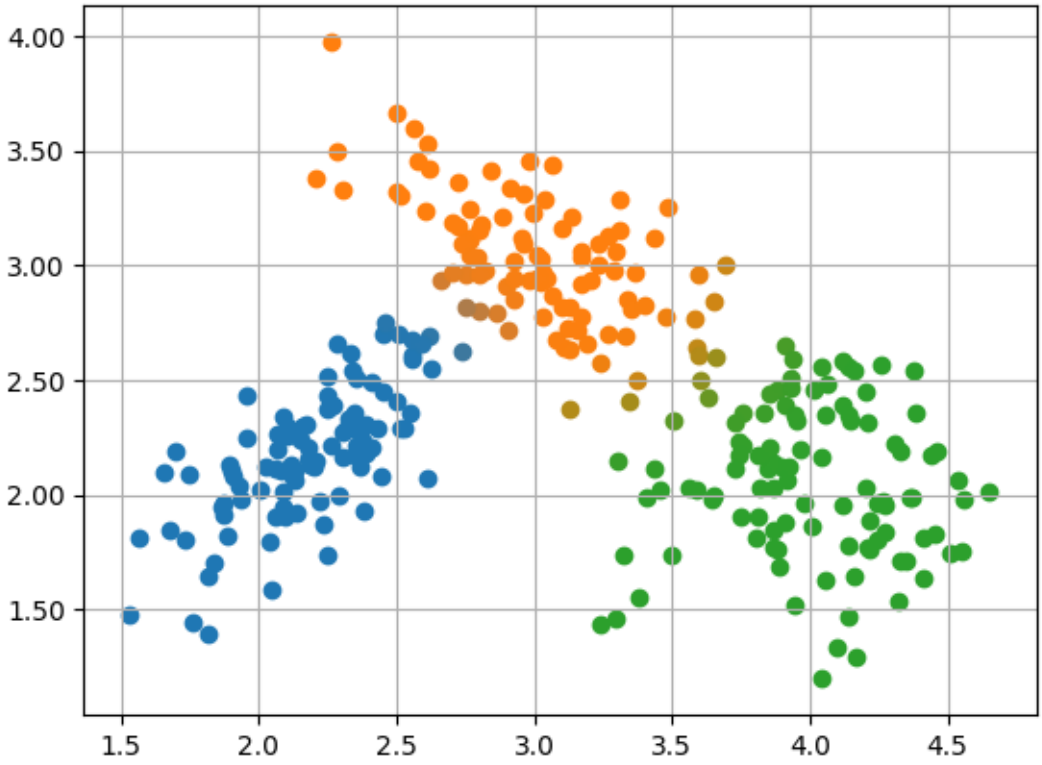
$\triangle$ Gaussian mixture example

K = 3

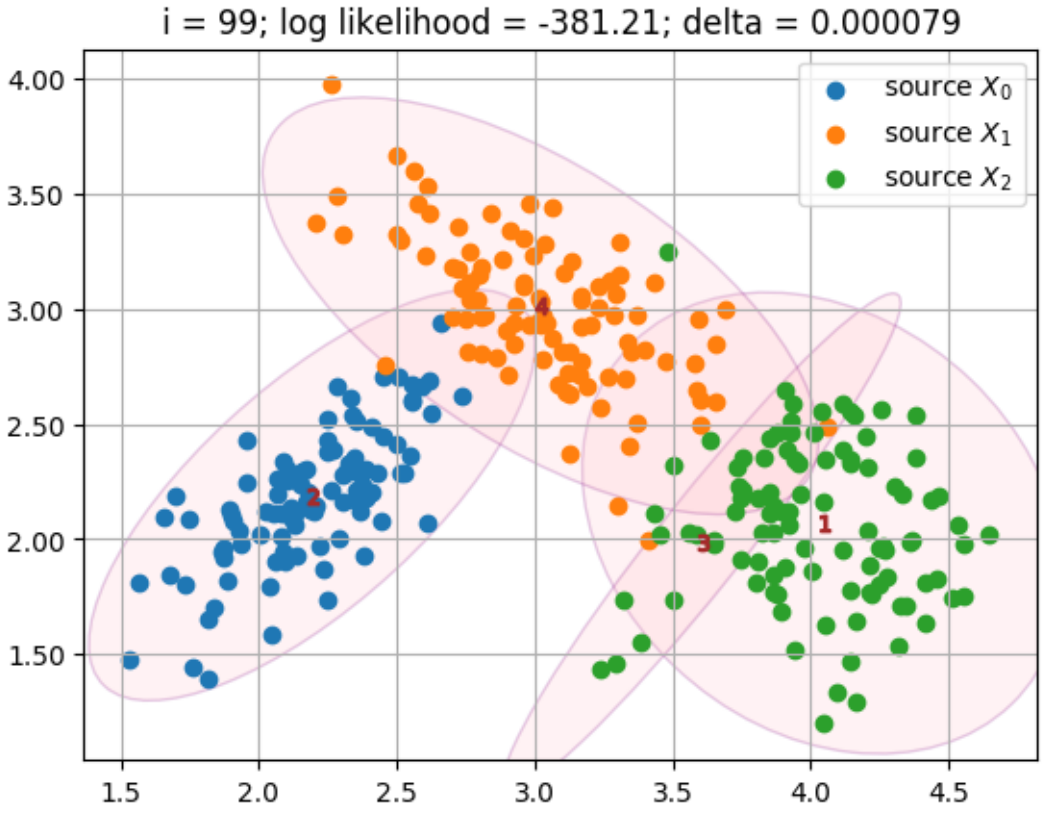
The final centers μ_k are shown with numbers and their corresponding $\mathbf{\Sigma}_k$ with pink ellipses.

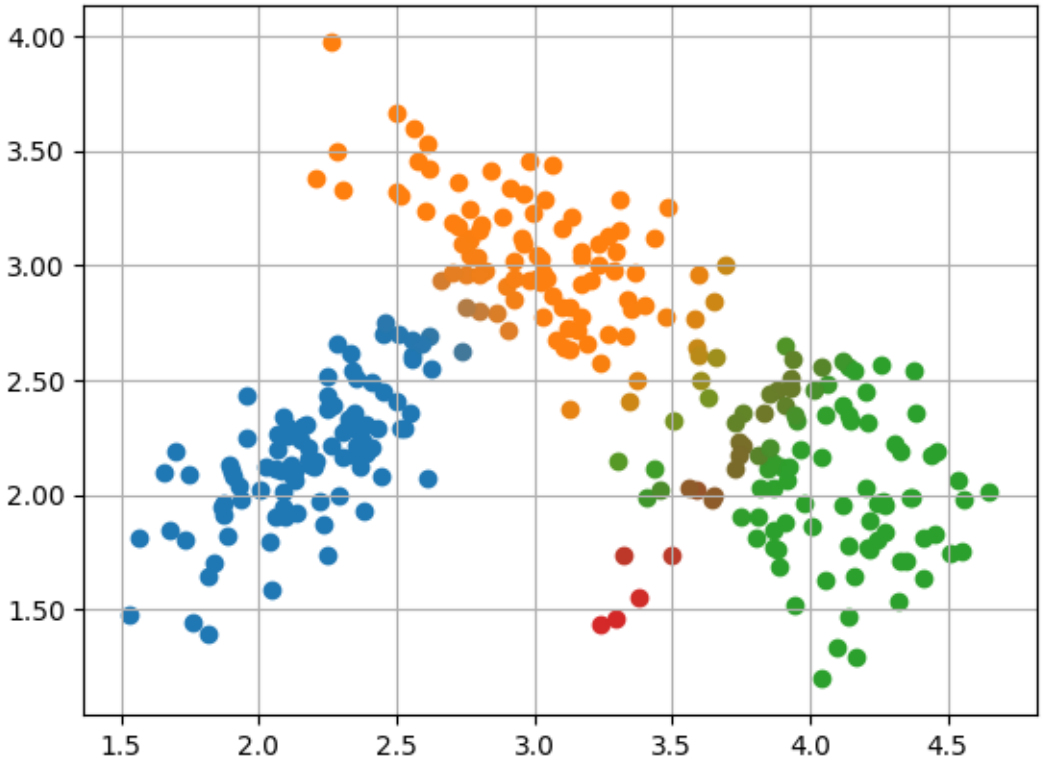


The data points X colored proportionally to the degree of belonging to a cluster:

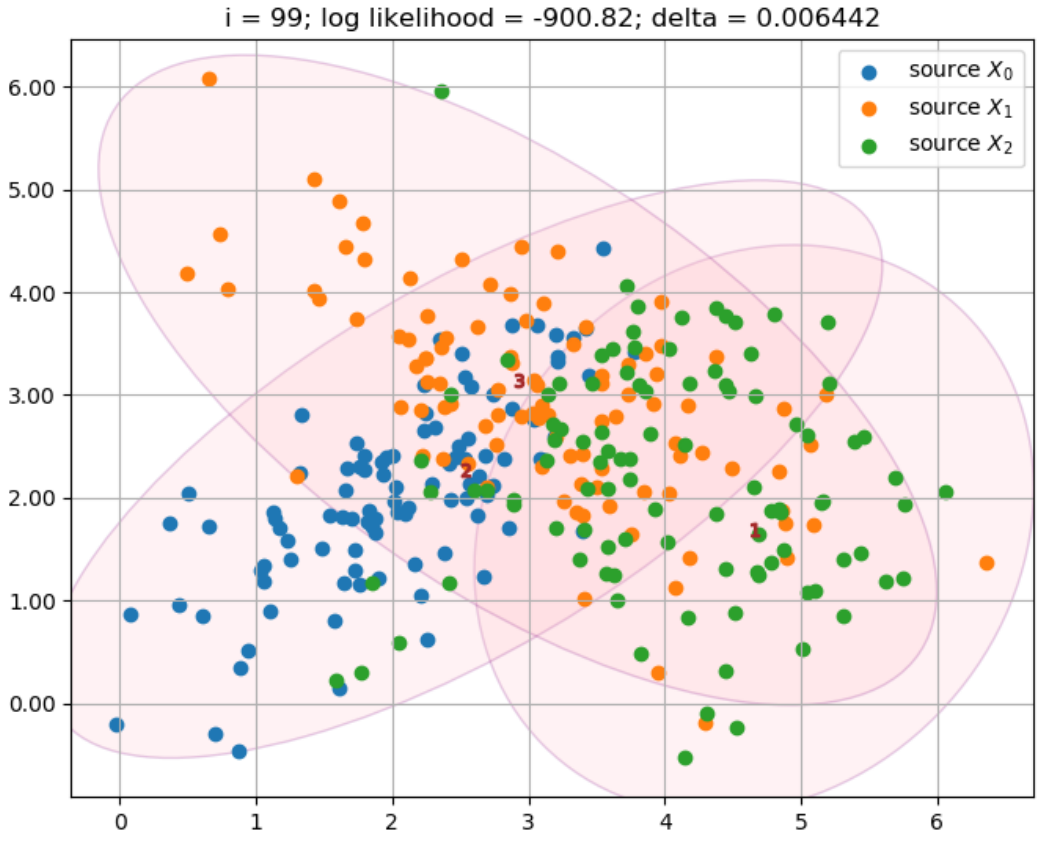


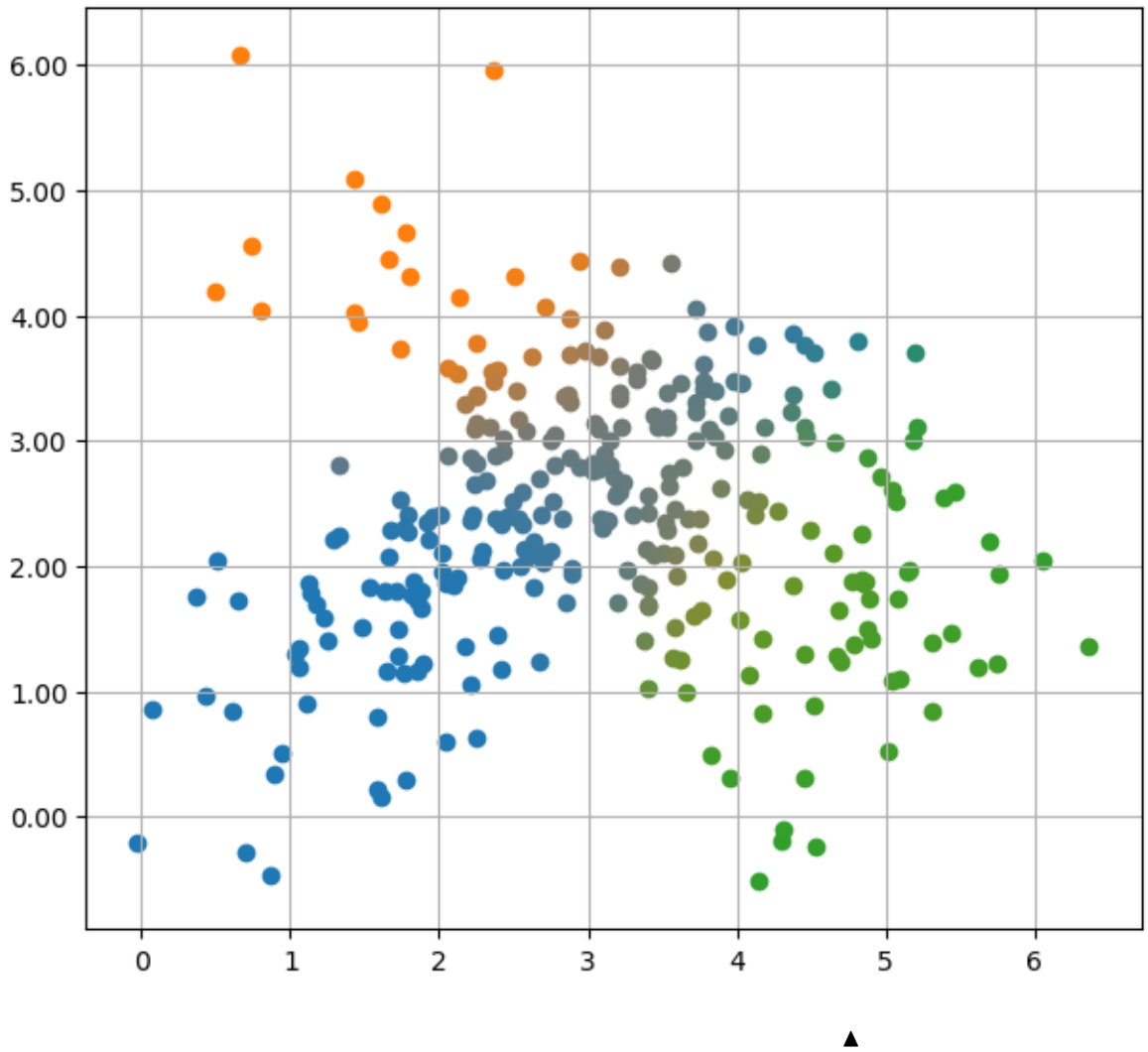
For comparison, here is trying to find 4 clusters:


i = 99; log likelihood = -381.21; delta = 0.000079
source X_0
source X_1
source X_2
4.00
3.50
3.00
2.50
2.00
1.50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
1
2
3
4



This is the first case with a high variance distribution:


i = 99; log likelihood = -900.82; delta = 0.006442
source X_0
source X_1
source X_2
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0
1
2
3
4
5
6
1
2
3



5.2 Spectral Clustering

Let's consider an undirected graph G as a pair (V, E) with n being the number of vertices. Its adjacency matrix is defined as

$$A_G(i,j) = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & otherwise \end{cases}$$

We define the degree of a vertex a to be the number of edges in which it participates. We naturally encode this in a vector

$$\boldsymbol{d}(a) = |\{b : (a,b) \in E\}|$$

where we write $|S|$ to indicate the number of elements in a set S. We then define the degree matrix

$$D_G(a,b) = \begin{cases} \boldsymbol{d}(a) & if\ a = b \\ 0 & otherwise \end{cases}$$

The diffusion matrix of G, also called the walk matrix of G, is then defined by

$$W_G = A_G D_G^{-1}$$

It acts on a vector $\boldsymbol{x}$ by

$$(W_G \boldsymbol{x})(a) = \sum_{b:(a,b)\in E} \boldsymbol{x}(b)/\boldsymbol{d}(b)$$

The most natural quadratic form to associate with a weighted graph is the Laplacian, which is given by

$$\boldsymbol{x}^T L_G \boldsymbol{x} = \sum_{(a,b)\in E} w(a,b)\left(\boldsymbol{x}(a) - \boldsymbol{x}(b)\right)^2 \tag{282}$$

This form measures the smoothness of the function $\boldsymbol{x}$. It will be small if the function $\boldsymbol{x}$ does not jump too much over any edge. In other words, Laplacian is the 2nd derivative (divergence at a vertex is the net outbound flow). The matrix defining this form is the Laplacian matrix of the graph G,

$$L_G := D_G - A_G = \begin{cases} -1 & if\ (a,b) \in E \\ \boldsymbol{d}(a) & if\ a = b \\ 0 & otherwise \end{cases}$$

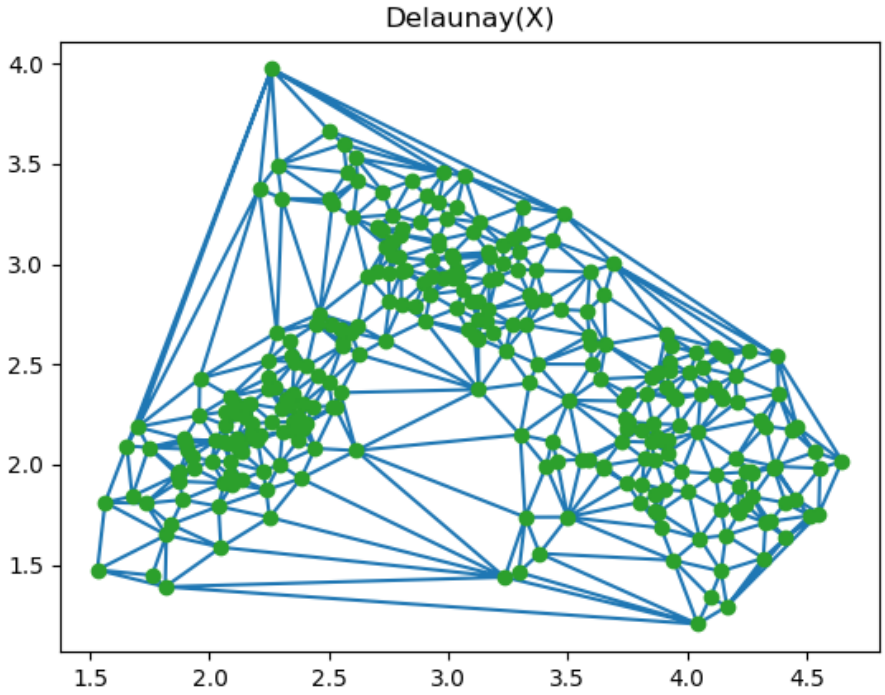
The Laplacian matrices of weighted graphs arise in many applications. For example, they appear when applying the certain discretization schemes to solve Laplace's equation with Neumann boundary conditions. They also arise when modeling networks of springs or resistors.

When studying random walks on a graph, it often proves useful to normalize the Laplacian by its degrees. The normalized Laplacian of G is defined by

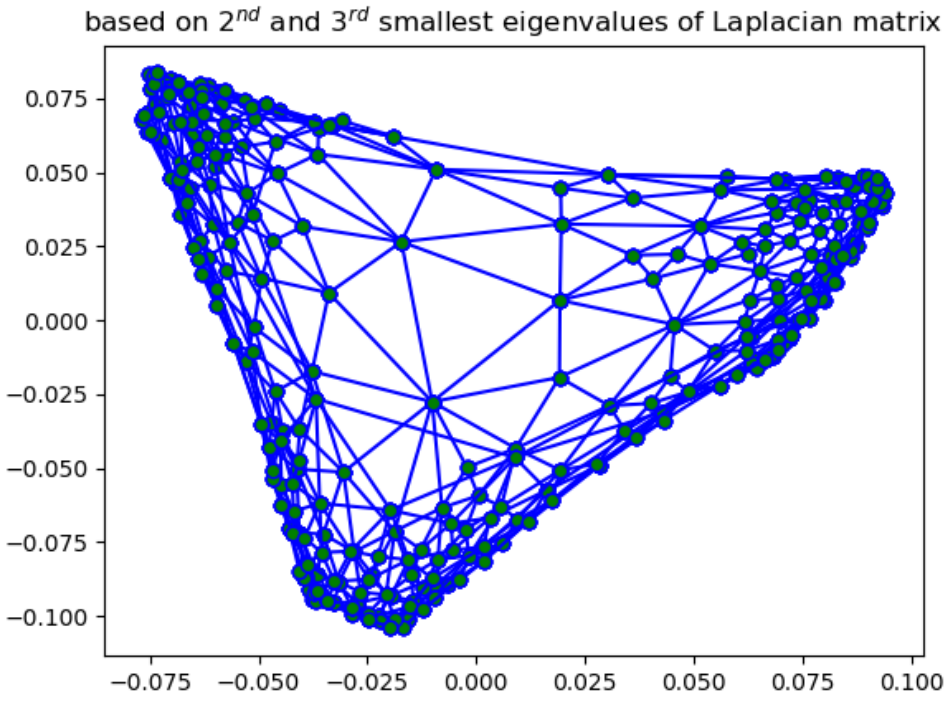
$$\mathcal{L}_G = D_G^{-1/2} L_G D_G^{-1/2} = I - D_G^{-1/2} A_G D_G^{-1/2}$$

For d-regular graphs the normalized Laplacian becomes $I - \frac{1}{d}A$.

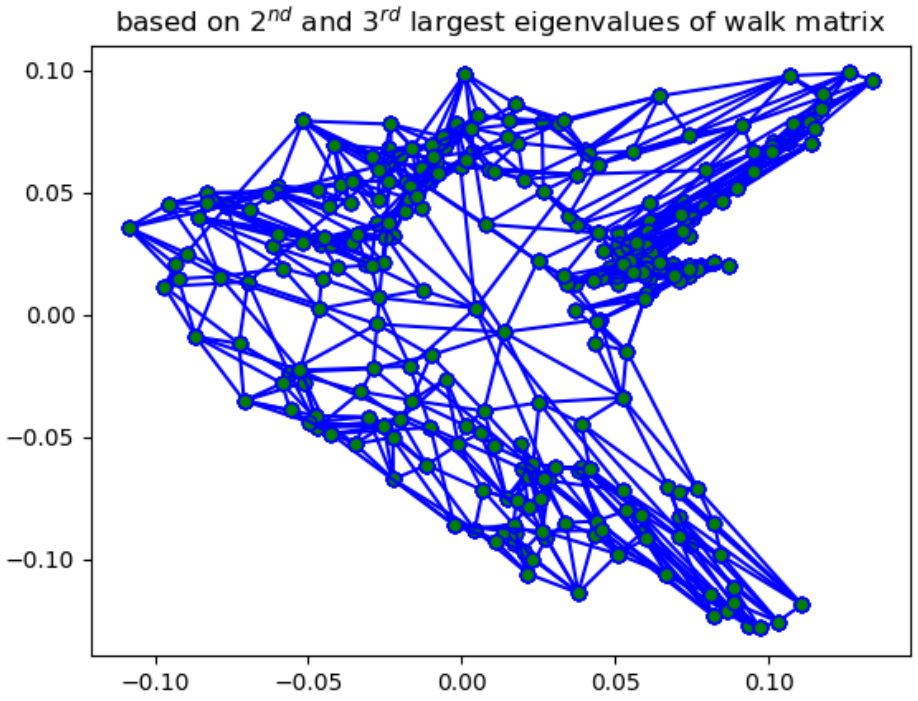
Let's plot the Delaunay graph of the Gaussian 3-cluster from the previous example:



We will now discard the information we had about the coordinates of the vertices, and draw a picture of the graph using only the eigenvectors of its Laplacian matrix.



Amazingly, this process produces a very nice picture of the graph, in spite of the fact that the coordinates of the vertices were generated solely from the combinatorial structure of the graph. Alternatively, we can use the walk matrix:



For a given complex Hermitian matrix M and nonzero vector $\boldsymbol{x}$, the Rayleigh quotient $R(M, \boldsymbol{x})$, is defined as:

$$R(M, \boldsymbol{x}) = \frac{\boldsymbol{x}^* M \boldsymbol{x}}{\boldsymbol{x}^* \boldsymbol{x}}$$

The Courant-Fischer Theorem: Let A be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Then

$$\lambda_k = \max_{\substack{S \subseteq \mathbb{R}^n \\ dim(S)=k}} \min_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{x} \neq 0}} \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \min_{\substack{T \subseteq \mathbb{R}^n \\ dim(T)=n-k+1}} \max_{\substack{\boldsymbol{x} \in T \\ \boldsymbol{x} \neq 0}} \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

The maximum in the first expression is taken over all subspaces of dimension k, and the minimum in the second is over all subspaces of dimension $n-k+1$. For example,

$$\lambda_1 = \max_{\boldsymbol{x} \in \mathbb{R}^n} \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

and

$$\lambda_n = \min_{\boldsymbol{x} \in \mathbb{R}^n} \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

We recall that a symmetric matrix A is positive semidefinite, written $A \succeq 0$, if all of its eigenvalues are non-negative. From (282) we see that the Laplacian is positive semidefinite. Adjacency matrices and walk matrices of non-empty graphs are not positive semidefinite as the sum of their eigenvalues equals their trace, which is 0. For this reason, one often considers the lazy random walk (formed by adding a loop of weight d_v to each vertex v) on a graph instead of the ordinary random walk. This walk stays put at each step with probability $1/2$. This means that the corresponding matrix is $(1/2)I + (1/2)W_G$, which can be shown to positive semidefinite.

For a collection of vertices $A \subseteq V(G)$, let ∂A denote the collection of all edges going from a vertex in A to a vertex outside of A:

$$\partial A = \{(a, b) \in E(G) \,:\, a \in A,\, b \in V(G) \setminus A\}$$

Then the Cheeger constant (or isoperimetric number) is defined by

$$h(G) = \min \left\{ \frac{|\partial A|}{|A|} \,:\, A \subseteq V(G),\, 0 < |A| \leq \frac{1}{2}|V(G)| \right\}$$

The Cheeger constant is strictly positive iff G is a connected graph. Intuitively, if the Cheeger constant is small but positive, then there exists a "bottleneck", in the sense that there are two "large" sets of vertices with "few" links (edges) between them. The Cheeger constant is "large" if any possible division of the vertex set into two subsets has "many" links between those two subsets.

The typical measure just assigns weight 1 to each vertex, so the measure of a subset is its number of vertices. We also can use a measure that takes into consideration the degree at a vertex. For a subset S of the vertices of G, we define the volume of S to be the sum of the degrees of the vertices in S

$$vol\, S = \sum_{x \in S} d_x$$

for $S \subseteq V(G)$. In these terms the Cheeger constant for a subset is

$$h_G(A) = \frac{|\partial A|}{\min(vol\, A,\, vol\, \bar{A})}$$

and the Cheeger constant of a graph is defined to be

$$h_G = \min_A h_G(A)$$

The Cheeger inequalities relate the eigenvalue gap of a graph with its Cheeger constant:

$$2h(G) \geq |\lambda_n| - |\lambda_{n-1}| \geq \frac{h^2(G)}{2d_{max}(G)}$$

where $d_{max}(G)$ is the maximum degree for the nodes in G and $|\lambda_n| - |\lambda_{n-1}|$ is the spectral gap of the Laplacian matrix of the graph. For a d-regular (expander) graph and its adjacency matrix A the above inequalities become

$$\frac{1}{2}(d - \lambda_2) \leq h(G) \leq \sqrt{2d(d - \lambda_2)}$$

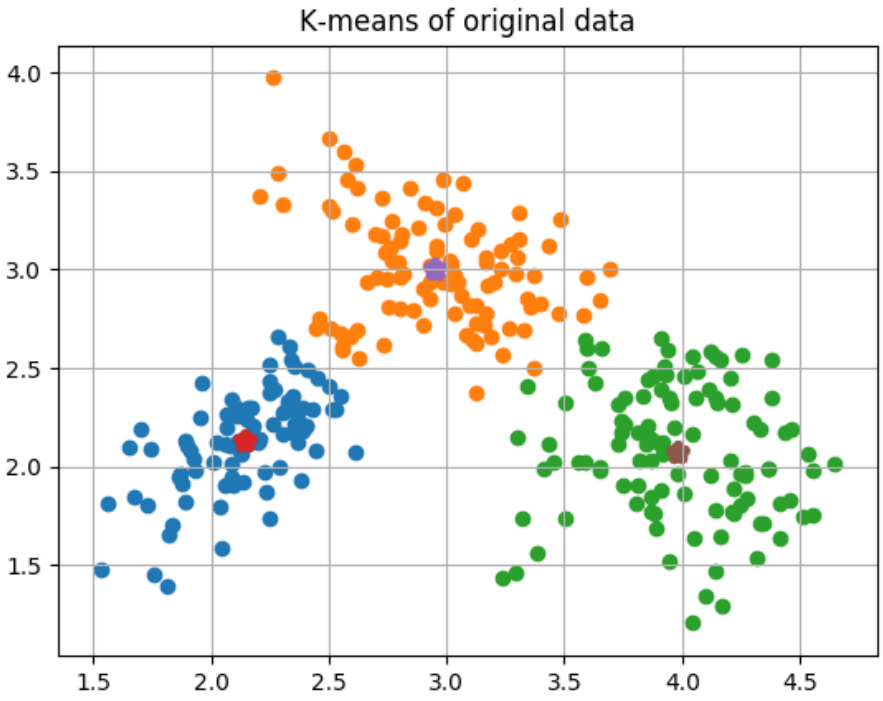
If λ_2 is large, the connected graph is an expander. If it is small, then the graph can be cut into 2 pieces without removing too many edges.

Roughly speaking, isoperimetric problems involving edge-cuts correspond in a natural way to Cheeger constants in spectral geometry.

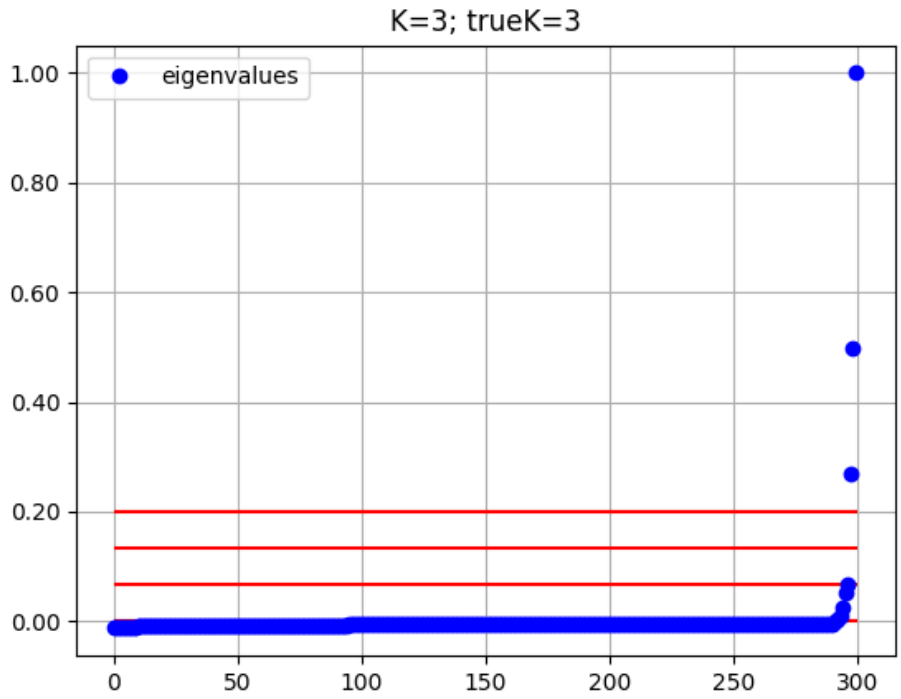
$\triangle$ **Spectral clustering example**.

We use the Gaussian similarity kernel to form the adjacency matrix. Then we build the normalized Laplacian $\mathcal{L}_G$. We find the k largest eigenvectors of $\mathcal{L}_G$ (chosen to be orthogonal to each other in the case of repeated eigenvalues), thus performing the dimensionality reduction. We renormalize them to have unit length. This means that more "outlying" points get down-weighted. Treating each row (of Y) as a point in $\mathbb{R}^k$ we cluster them into k clusters via K-means in the low-dimensional space. In this context, K-means is essentially used as a rounding algorithm. Finally, we assign the original points according to the clusters of Y.

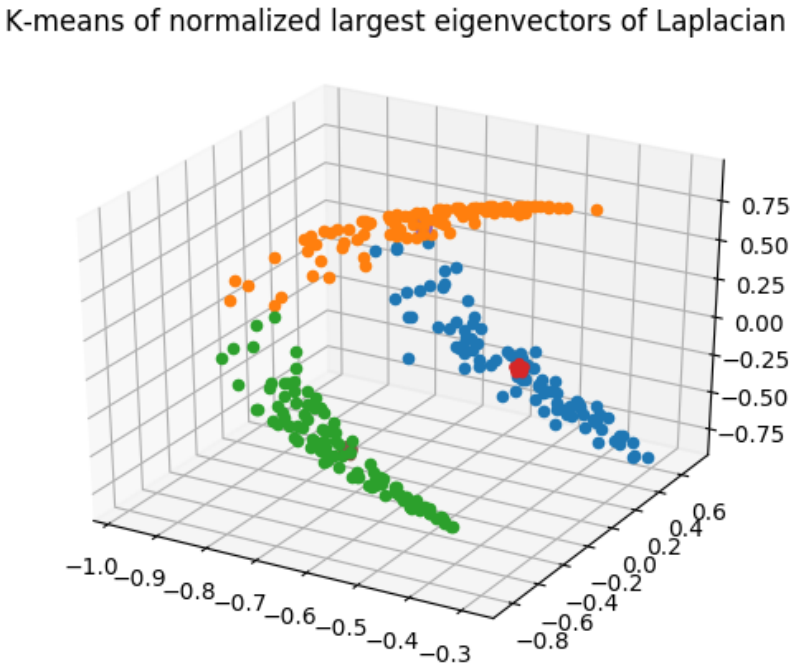
This is for the same data as in the previous section.



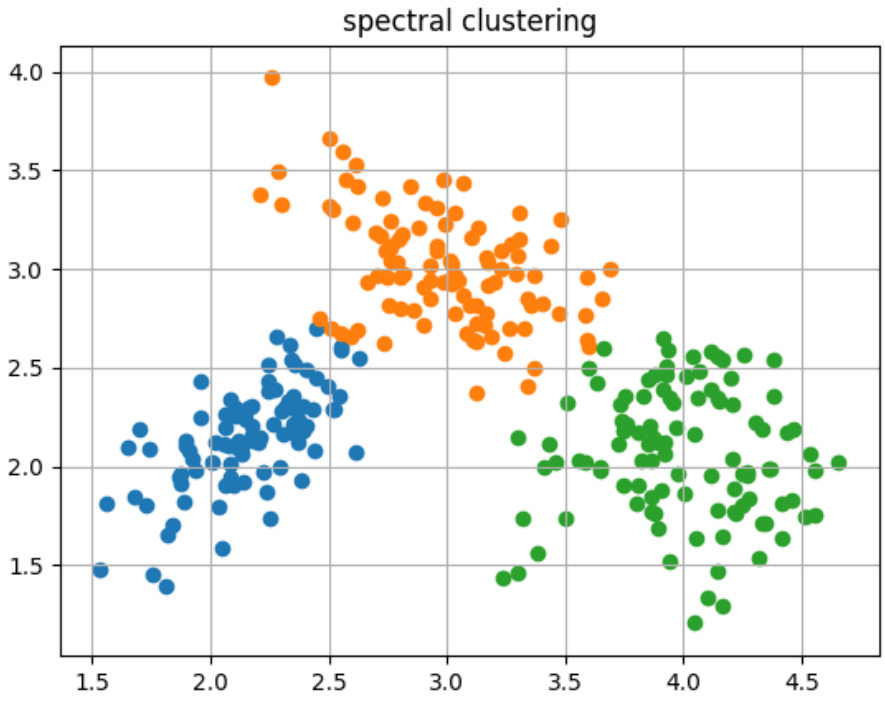
Largest eigenvalues of the normalized Laplacian matrix:



Pick values over 3 standard deviations, and perform K-means over these normalized eigenvectors.

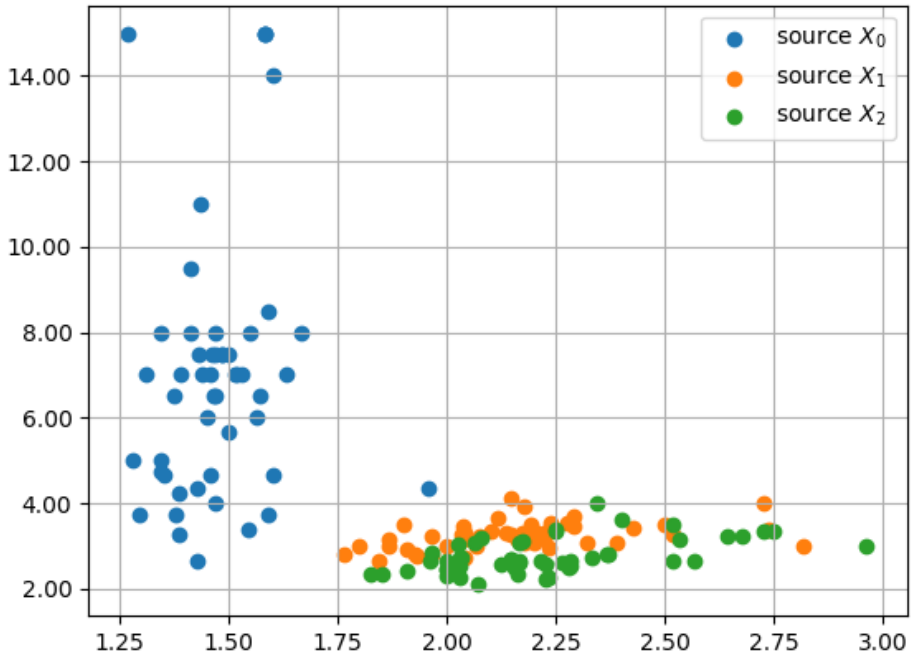


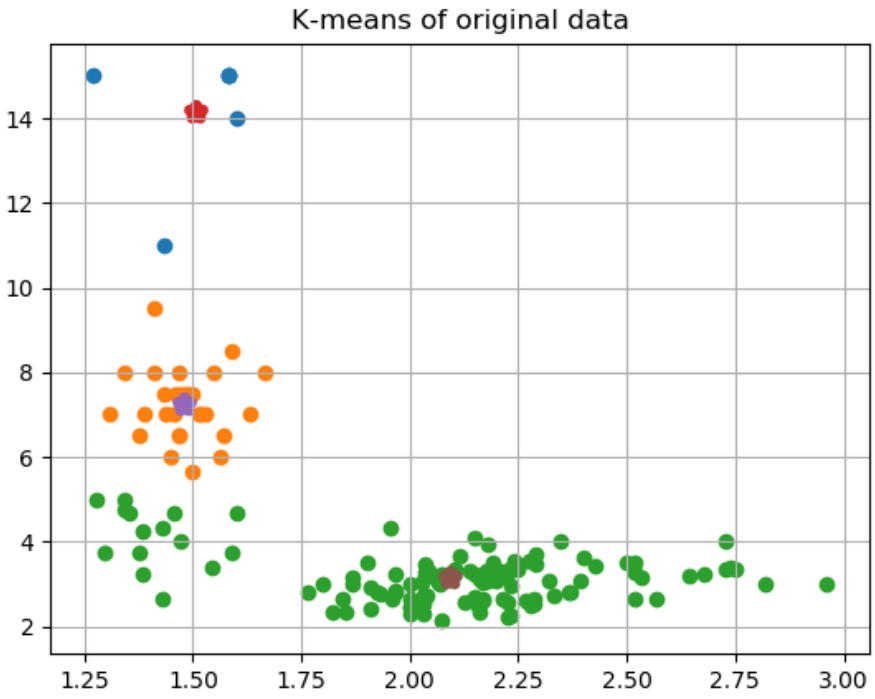
and draw the original points with the same clustering index:

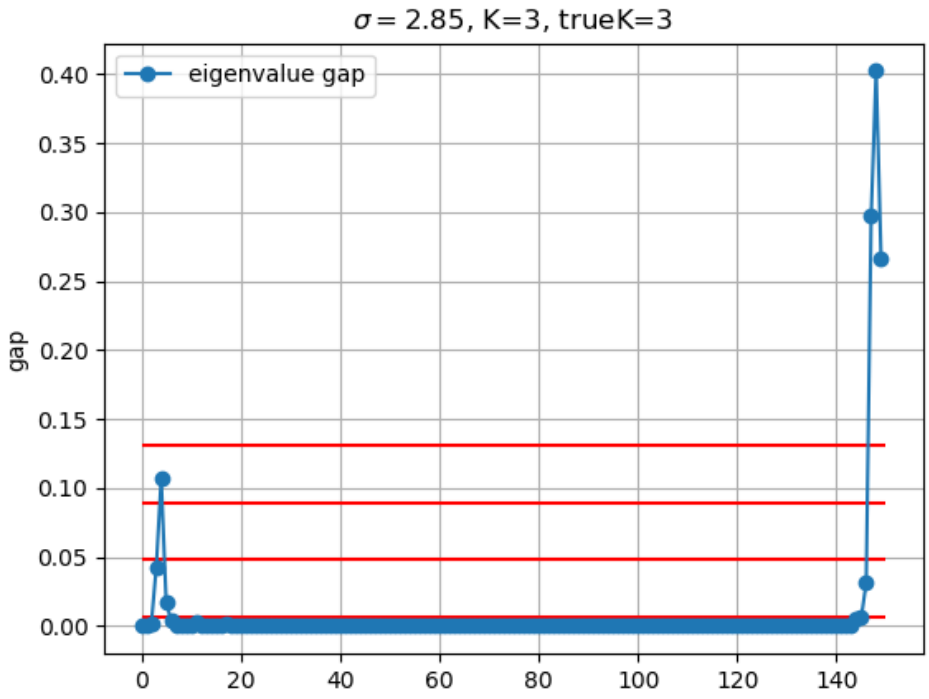


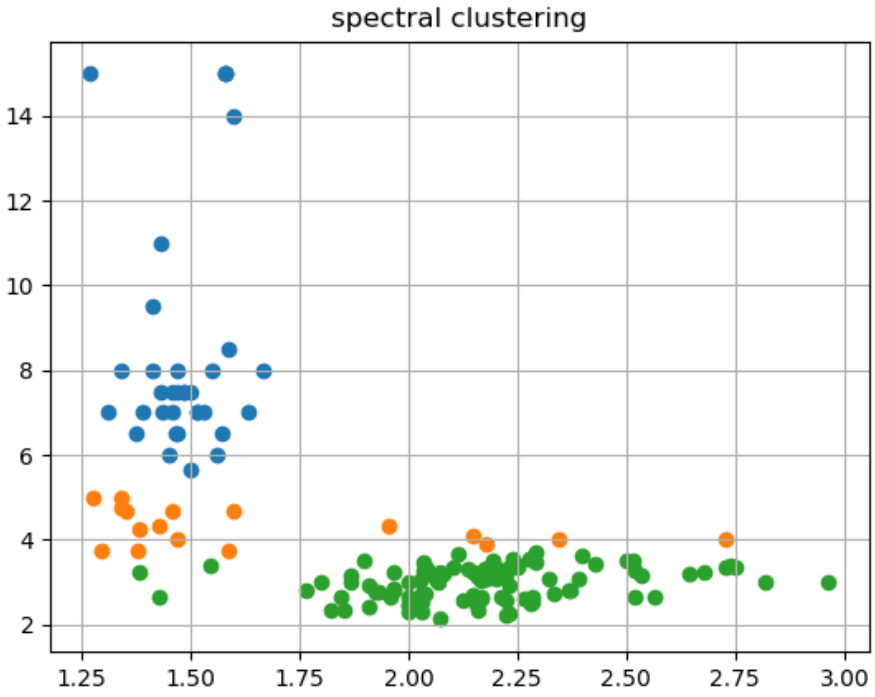
▲

△ **The iris data set.**


source X_0
source X_1
source X_2
14.00
12.00
10.00
8.00
6.00
4.00
2.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00


K-means of original data
14
12
10
8
6
4
2
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00


$\sigma = 2.85$, K=3, trueK=3
eigenvalue gap
gap
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100
120
140



▲

$$GMM \rightarrow K-means \rightarrow \begin{matrix} spectral \\ clustering \end{matrix} \rightarrow DBSCAN \rightarrow \begin{matrix} mean \\ shift \end{matrix}$$

Let $x \in \mathbb{R}^d$ be a d-dimensional random vector. A Gaussian distribution with mean vector $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ can be written in the form:

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \tag{283}$$

where $|\Sigma|$ denotes the determinant.

Given a dataset X composed by n i.i.d. data points $x_1, \ldots, x_n$ we can use a Gaussian distribution as a model to fit this dataset. A common criterion for fitting the parameters μ and Σ given an observed dataset is to find the values that maximize the likelihood function:

$$p(X|\mu,\Sigma) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu,\Sigma) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right] \tag{284}$$

This procedure is known as Maximum Likelihood Estimation (MLE). Given the form of the Gaussian distribution, it is more convenient to maximize the log of the likelihood function. Taking log on both sides:

$$\ln p(X|\mu,\Sigma) = -\frac{nd}{2}\ln(2\pi) - \frac{n}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) \tag{285}$$

Maximizing with respect to μ we obtain

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{286}$$

Similarly, maximizing w.r.t. Σ

$$\Sigma_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_{MLE})(x_i-\mu_{MLE})^T \tag{287}$$

A mixture of Gaussians is described in the previous subsection.

Let's consider a GMM with $\Sigma_k = \epsilon I \, for \, \forall k$. Then for point x and cluster k (283) becomes

$$p(x|\mu_k,\Sigma_k) = \frac{1}{(2\pi\epsilon)^{d/2}} exp\left(-\frac{1}{2\epsilon}\left\|x-\mu_k\right\|^2\right) \tag{288}$$

In the E step the responsibilities $\gamma(z_k)$ from (274) for data point x_i are now given by

$$\gamma_i(z_k) = \frac{\pi_k exp\left(-\frac{1}{2\epsilon}\left\|x_i-\mu_k\right\|^2\right)}{\sum_{j=1}^{K}\pi_j exp\left(-\frac{1}{2\epsilon}\left\|x_i-\mu_j\right\|^2\right)} \tag{289}$$

If we consider the limit $\epsilon \to 0$, we observe that in the denominator the term for which $\|x_i-\mu_j\|^2$ is smallest will go to zero most slowly, and hence the responsibilities $\gamma_i(z_k)$ for the data point x_i all go to zero except for term j, for which the responsibility $\gamma_i(z_j)$ will go to one. Note that this holds independently of the values of π_k as long as none of the π_k is zero. Therefore, in the limit $\epsilon \to 0$, EM algorithm for GMM returns a hard assignment: it assigns every data point x_i to just one specific cluster k having the closest mean μ_k.

In the M step we observe that we don't need the second equation for $\mathbf{\Sigma}_k^{new}$ as all covariances are constant and $= \epsilon$, and the third equation since π_k^{new} are not required for the E step. Also we can merge the first equation which updates the means μ_k to the mean of the current cluster (after the E step update):

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{290}$$

which is the same as (270). Note that for K-means, "closest" means closest w.r.t. to Euclidean distance, while for a GMM, "closest" means closest w.r.t. to Mahalanobis distance.

We can reformulate K-means as following. Given a dataset X with N data points the algorithm tries to find K points in the space (the means μ_k) and assign every data point x_i to one of these K points such that the distance between x_i and μ_k is minimized. We can then think of the K-means as an iterative algorithm that returns a local minima for the following optimization problem:

$$\begin{gathered} \min_{r_{i,k},\mu_k} \sum_{k=1}^{K} \sum_{i \in \Gamma_k} \|x_i - \mu_k\|^2 \\ s.t.\, r_{i,k} \in \{0,1\} \quad \forall i = 1,2,\ldots,N;\, k = 1,2,\ldots,K \end{gathered} \tag{291}$$

where we have used the notation $i \in \Gamma_k$ to denote that point x_i belongs to cluster Γ_k, i.e. $r_{i,k} = 1$ and $r_{i,j} = 0\ \forall j \neq k$. A solution for this problem consists of K cluster means μ_k and a hard assignment $r_{i,k}$ for every data point x_i. The K-means algorithm returns a local minima for this optimization problem using the EM algorithm.

A major drawback of the K-means algorithm is that it can not separate clusters that are non-linearly separable in input space S (i.e., if the intersection of the convex hull of the clusters is not empty). One approach for tackling this problem is using the kernel method, leading to an algorithm called kernel K-means.

A kernel function $\varkappa : S \times S \to \mathbb{R}$ computes the dot product between a couple of mapped points $\varphi(x_i)$ and $\varphi(x_j)$ without explicitly computing the mappings $\varphi(x_i)$ and $\varphi(x_j)$, i.e. $\varkappa(x_i, x_j) = \varphi^T(x_i)\varphi(x_j)$. Popular kernel functions are the polynomial kernel $\varkappa(x_i, x_j) = (x_i^T x_j + c)^b$, $c, b \in \mathbb{R}$ or the Gaussian kernel $\varkappa(x_i, x_j) = exp\left(-\frac{\|x_i - \mu_j\|^2}{2\sigma^2}\right)$, $\sigma \in \mathbb{R}$.

Another extension to the K-means algorithm is to associate each data point x_i with a weight $w_i \in \mathbb{R}$. Depending on the context, weight w_i usually express how important point x_i is. Adding both the kernel method and the weights to the optimization formulation of the K-means problem (291) results in the following optimization problem:

$$\begin{gathered} \min_{r_{i,k},\mu_k} \sum_{k=1}^{K} \sum_{i \in \Gamma_k} w_i \|\varphi(x_i) - m_k\|^2 \\ s.t.\, r_{i,k} \in \{0,1\} \quad \forall i = 1,2,\ldots,N;\, k = 1,2,\ldots,K \end{gathered} \tag{292}$$

Note that weights w_i are not optimization variables, they are known constants, an input to the optimization problem. Also, observe that we have used m_k to denote the weighted mean of the points in cluster Γ_k, but in the space $\mathcal{F}$:

$$m_k = \frac{\sum_{i \in \Gamma_k} w_i \varphi(x_i)}{s_k} \tag{293}$$

where $s_k = \sum_{i \in \Gamma_k} w_i$ is the total weight of cluster Γ_k. Observe that m_k is the point that minimizes

$$\sum_{i \in \Gamma_k} w_i \left\| \varphi(x_i) - m_k \right\|^2$$

i.e. m_k is the "best" cluster representative for cluster Γ_k in space $\mathcal{F}$. We can rewrite the objective function of problem (292) as follows

$$\begin{aligned}
\sum_{k=1}^{K} \sum_{i \in \Gamma_k} w_i \left\| \varphi(x_i) - m_k \right\|^2 &= \sum_{k=1}^{K} \sum_{i \in \Gamma_k} w_i \left(\varphi(x_i) - m_k \right)^T \left(\varphi(x_i) - m_k \right) \\
&= \sum_{k=1}^{K} \sum_{i \in \Gamma_k} w_i \left(\varphi^T(x_i)\varphi(x_i) - 2\varphi^T(x_i)m_k + m_k^T m_k \right) \\
&= \sum_{k=1}^{K} \left(\sum_{i \in \Gamma_k} w_i \varphi^T(x_i)\varphi(x_i) - 2 \sum_{i \in \Gamma_k} w_i \varphi^T(x_i)m_k + \sum_{i \in \Gamma_k} w_i m_k^T m_k \right) \\
&= \sum_{k=1}^{K} \left(\sum_{i \in \Gamma_k} w_i \varphi^T(x_i)\varphi(x_i) - 2 s_k m_k^T m_k + s_k m_k^T m_k \right) \\
&= \sum_{k=1}^{K} \left(\sum_{i \in \Gamma_k} w_i \varphi^T(x_i)\varphi(x_i) - s_k m_k^T m_k \right) \\
&= \sum_{k=1}^{K} \sum_{i \in \Gamma_k} w_i \varphi^T(x_i)\varphi(x_i) - \sum_{k=1}^{K} s_k m_k^T m_k
\end{aligned}$$

where the fourth equality is due to (293). Also, note that the first term in the last equation does not depend on the assignments $r_{i,k}$ or the cluster centers m_k, so we can remove it from the objective function, obtaining the following objective function

$$-\sum_{k=1}^{K} s_k m_k^T m_k = -\sum_{k=1}^{K} \frac{\sum_{i,j \in \Gamma_k} w_i w_j \varphi^T(x_i)\varphi(x_j)}{s_k} \tag{294}$$

From the form of this equation, we can see that the quantity to minimize (across all the clusters) is the negative sum of the weighted dot product (in space $\mathcal{F}$) between all pair of points in the same cluster, normalized by the total weight of the cluster. In other words, and interpreting the dot product as a similarity measure (in terms of the direction of the data points), we are trying to maximize the normalized similarity among points in the same cluster, where the normalization term is the weight of the cluster, and the similarity function between two points is defined by the kernel $\varkappa$.

Letting $W \in \mathbb{R}^{N \times N}$ be the diagonal matrix with all the weights w_i in the diagonal, $W_k \in \mathbb{R}^{|\Gamma_k| \times |\Gamma_k|}$ (where $|\Gamma_k|$ denotes the number of elements in cluster Γ_k) be the diagonal matrix of weights in cluster Γ_k; $\Phi \in \mathbb{R}^{dim(\mathcal{F}) \times |\Gamma_k|}$ be

the matrix formed by horizontally concatenating the points $\varphi(x_i)$ (i.e. $\Phi = [\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_n)]$), and e_k the vector of ones of size $|\Gamma_k|$ we can rewrite m_k as

$$m_k = \frac{\Phi_k W_k e_k}{s_k} \tag{295}$$

Using this equation, we can rewrite equation (294) as

$$\begin{aligned}
-\sum_{k=1}^{K} s_k m_k^T m_k &= -\sum_{k=1}^{K} \frac{e_k^T W_k \Phi_k^T \Phi_k W_k e_k}{s_k} \\
&= -\sum_{k=1}^{K} \frac{e_k^T W_k \Phi_k^T}{\sqrt{s_k}} \frac{\Phi_k W_k e_k}{\sqrt{s_k}} \\
&= -Tr\left(Y^T W^{1/2} \Phi^T \Phi W^{1/2} Y\right) \\
&= -Tr\left(Y^T W^{1/2} \mathcal{K} W^{1/2} Y\right)
\end{aligned}$$

where $\mathcal{K} = \Phi^T \Phi$ is the kernel matrix of the data, i.e. $\mathcal{K}_{i,j} = \varkappa(x_i, x_j)$, and

$$Y = \begin{bmatrix}
\frac{W_1^{1/2} e_1}{\sqrt{s_1}} & 0 & 0 & 0 \\
0 & \frac{W_2^{1/2} e_2}{\sqrt{s_2}} & 0 & 0 \\
0 & 0 & \ddots & 0 \\
0 & 0 & 0 & \frac{W_K^{1/2} e_K}{\sqrt{s_K}}
\end{bmatrix} \tag{296}$$

is the assignment matrix, where all the zeros denotes zero matrices of appropriate size. Note that Y is an $n \times K$ orthonormal matrix, i.e., $Y^T Y = I$. We can then re-cast the optimization problem (292) as

$$\begin{aligned}
&\max_{Y \in \nu_Y^{n \times K}} Tr\left(Y^T W^{1/2} \mathcal{K} W^{1/2} Y\right) \\
&s.t.\, Y^T Y = I
\end{aligned} \tag{297}$$

where $\nu_Y = \{0\} \cup \left\{\sqrt{\frac{w_i}{s_k}}\right\}$, $i \in \Gamma_k$. We will denote problem (297) as the matrix version of the weighted kernel K-means problem. There are two important things to observe here. The first one is that, if we set all the weights w_i equal to one and set the function φ to be the identity function, then problem (292) turns into problem (291), the vanilla K-means problem, and the corresponding matrix version of this problem turns into the following optimization problem

$$\begin{aligned}
&\max_{Y \in \nu_Y^{n \times K}} Tr\left(Y^T G Y\right) \\
&s.t.\, Y^T Y = I
\end{aligned} \tag{298}$$

where $\nu_Y = \{0\} \cup \left\{\sqrt{\frac{1}{s_k}}\right\}$, $i \in \Gamma_k$ and G is the Gram matrix of the data points, i.e. $G_{ij} = x_i^T x_j$.

The second thing is that, just like the K-means algorithm can find a local minima for the vanilla K-means problem, we can use a modified version of the K-means algorithm to find a local minima for the weighted kernel K-means problem.

The spectral decomposition of matrix $W^{1/2}\mathcal{K}W^{1/2}$ can provide a different solution for the weighted kernel K-means problem. By allowing Y to be an arbitrary orthonormal matrix in (297), we can obtain an optimal Y by taking the top K eigenvectors (i.e., the eigenvectors associated to the K largest eigenvalues) of the matrix $W^{1/2}\mathcal{K}W^{1/2}$. Each row of the resulting matrix Y is then interpreted as an "embedding" version of the original data point in a lower dimensional space with dimension equal to K. The typical way to proceed is to compute a discrete partition of the embedded data points, usually using the vanilla K-means algorithm with the embedded points.

In the spectral clustering algorithm we used

$$\mathcal{K}_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

and

$$W^{-1} = diag(\mathcal{K}\mathbf{1}_n)$$

It can be shown that this problem is analogous to the min cut graph problem or, in other words, we are trying to maximize the normalized similarity among points in the same cluster, where the normalization term is the "weight" of the cluster.

Unlike the (continuous) Gaussian kernel above let's introduce a (discontinuous) Heaviside kernel as

$$\mathcal{K}_H(x_i, x_j) = \begin{cases} 1 & if\ \epsilon - \|x_i - x_j\| \geq 0 \\ 0 & otherwise \end{cases} \tag{299}$$

The 2nd modification to the algorithm is that after computing matrix of weights D we introduce a filtering step: we will filter the nodes in G using their weights $d_i = D_{ii}$ (in this case, due the use of the Heaviside kernel, the weight d_i is equal to the degree of its corresponding node, plus one). This filtering process will be simple:

- If $d_i = 1$ (i.e., if node i is an isolated node), we will label the corresponding node (and therefore, the corresponding data point x_i) as an outlier, removing it from the graph G, with all its adjacent edges, and we will remove the corresponding row and column from the matrix A.
- If $1 < d_i \leq minPts$, the corresponding node (and therefore, the corresponding data point x_i) will be labeled as unprocessed, removing it from the graph G, with all its adjacent edges, and we will remove the corresponding row and column from the matrix A.

- Finally, if $d_i > minPts$, we will label the corresponding node (and therefore, the corresponding data point x_i) as a core node, preserving it in the graph G, with all its adjacent edges, and preserving the corresponding row and column in the matrix A.

We can observe that, although we haven't done it explicitly, if two points have a similarity value of zero, the corresponding edge will be removed (or more precisely, not taken into account) from graph G. Therefore, we can think of (post-filtered) matrix A as the sum of the adjacency matrix A_G of (post-filtered) graph G plus the identity matrix of appropriate size

$$A = A_G + I$$

Similarly, we can think of (post-filtered) matrix D as the sum of the degree matrix D_G of (post-filtered) graph G plus the identity matrix of appropriate size

$$D = D_G + I$$

There is something important to highlight with respect to this last point. Since matrix A has been modified, the post-filtered matrix D needs to be recomputed $D = diag(A\mathbf{1}_n)$. The Laplacian matrix $L \in \mathbb{R}^{n_{core} \times n_{core}}$ of (post-filtered) graph G is defined as:

$$L = D_G - A_G = D_G - A_G + (I - I) = D - A$$

Given the Laplacian matrix L and the degree matrix D_G of a graph G, the normalized Laplacian of a matrix is defined as:

$$\mathcal{L} = D_G^{-1/2} L D_G^{-1/2}$$

A very well known result in Graph Theory is that eigenvectors related to zero eigenvalues of the normalized Laplacian indicate connected components in the graph G. The results still holds if we use the matrix D instead of D_G as can be easily verified. Therefore, given an eigenvector v_0 related to a zero eigenvalue

$$\begin{aligned} 0 &= \mathcal{L} v_0 \\ &= D_G^{-1/2} L D_G^{-1/2} v_0 \\ &= D^{-1/2} L D^{-1/2} v_0 \\ &= D^{-1/2} (D - A) D^{-1/2} v_0 \\ &= v_0 - D^{-1/2} A D^{-1/2} v_0 \end{aligned}$$

which implies

$$D^{-1/2} A D^{-1/2} v_0 = v_0$$

i.e., normalized Laplacian's eigenvector v_0 is also an eigenvector of the matrix $D^{-1/2} A D^{-1/2}$, in this case with eigenvalue equal to one.

This last observation has an important consequence: the eigenvectors obtained in the next step (eigen-decomposition step) with associated eigenvalue equal to one will be indicator vectors for the connected components in (post-filtered) graph G: for any of these eigenvectors, all the non-zero entries corresponds to points belonging to the same cluster. This property will allows to modify the eigen-decomposition step in order to remove the dependency to the parameter K. We also don't need to calculate other eigenvalues, and drop the partition step, too.

In reality there is a more efficient algorithm for DBSCAN but this one explains its connection to spectral clustering.

One common assumption in Machine Learning is that the data points $x_i \in \mathbb{R}^d$ are i.i.d. samples from a unknown density function $\mathcal{D}$. We can think of this density function as a hyper-surface in $\mathbb{R}^{d+1}$. The peaks of this hyper-surface corresponds to the denser parts of the space $\mathbb{R}^d$, i.e., where data points are more concentrated.

We can find the core points in a dataset X sampled from a density function $\mathcal{D}$ by each data point x_i in X "climbing the hill" trying to reach its closest peak (i.e. its local maxima). Because we don't know how the density function looks like, we will use the data itself to approximate the shape of the density function and the desired altitude.

In order to "climb the hill", we will follow a simple procedure: given a dataset X, a data point x_i and a radius ε, we first compute the ε-neighborhood $N_\varepsilon(x_i) = \left\{ x_j \in X \;\middle|\; \|x_i - x_j\| \leq \varepsilon \right\}$, and then, using this local information, we compute the mean of the points in $N_\varepsilon(x_i)$. This mean is the first stop in the path to the peak, and the algorithm iterates until the point reaches the desired altitude, for every data point. The main idea is that this procedure estimates the gradient of the density function $\mathcal{D}$. To see this, consider the kernel density estimator $f_\varkappa(x)$:

$$f_\varkappa(x) = \frac{1}{nh^d} \sum_{i=1}^{n} \varkappa\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)$$

where h^d is the volume of a hypercube of side h in a d dimensional space. It can be seen that, as its name suggest, $f_\varkappa(x)$ is an estimator of the value of the density $\mathcal{D}$ at the point x. This estimation is done using the sampled data points x_i. Given the estimator $f_\varkappa(x)$, we can compute its gradient w.r.t. x

$$\begin{aligned}
\nabla f_\varkappa(x) &= \frac{2}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) \varkappa'\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \\
&= \frac{2}{nh^{d+2}} \left[\sum_{i=1}^{n} \varkappa'\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[x - \frac{\sum_{i=1}^{n} x_i \varkappa'\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} \varkappa'\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} \right]
\end{aligned}$$

We can observe that the first term of $\nabla f_\varkappa(x)$ is proportional to $f_{\varkappa'}(x)$. The second term is the (negative) mean shift:

$$m(x) = x - \frac{\sum_{i=1}^{n} x_i \varkappa'\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} \varkappa'\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \tag{300}$$

which is the difference between x and a weighted mean of the data points x_i. Using the last 2 equations:

$$\nabla f_{\varkappa}(x) = \frac{2}{h^2} f_{\varkappa'}(x) m(x) \tag{301}$$

yielding

$$m(x) = \frac{h^2}{2} \frac{\nabla f_{\varkappa}(x)}{f_{\varkappa'}(x)} \tag{302}$$

This last expression shows that the mean shift vector $m(x)$ is proportional to the gradient of the kernel density estimator $f_{\varkappa}(x)$ and therefore, it is an estimate of the gradient of $\mathcal{D}$ at the point x. In order to climb the hill, we want to move in a direction proportional to $m(x)$:

$$\begin{aligned} x_i^{new} &= x - m(x) \\ &= \frac{\sum_{i=1}^{n} x_i \varkappa'\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} \varkappa'\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \\ &= \frac{\sum_{i=1}^{n} x_i \varkappa_H(x, x_i)}{\sum_{i=1}^{n} \varkappa_H(x, x_i)} \\ &= \frac{\sum_{x_j \in N_\varepsilon(x_i)} x_j}{|N_\varepsilon(x_i)|} \end{aligned}$$

where we have set $h = 1$ and used the Heaviside kernel $\varkappa_H$ as $\varkappa'$ in the third equality.

DBSCAN can be viewed as a climbing procedure that stops once a data point has reached a given value of density. Mean shift follows a similar procedure, but its stopping criteria is different, it stops until a data point has reached its closest local maxima.

In summary:

- Setting all the covariance matrices equal to εI and considering the limit $\varepsilon \to 0$ take us from GMM to K-means.
- Adding flexibility to K-means via a Gaussian kernel $\varkappa_G$ and introducing a weight w_i for every data point yields a Spectral Clustering algorithm.
- Using a different kernel $\varkappa_H$ and introducing a filtering step take us from Spectral Clustering to DBSCAN.
- Climbing to the peak instead of stopping at a certain level take us from DBSCAN to Mean shift.

References

[1] Christopher Bishop “Pattern recognition and machine learning”, 2006

[2] Mussa-Ivaldi Shadmehr “Biological learning and control”

[3] R. E. Kalman “A New Approach to Linear Filtering and Prediction Problems”, 1960

[4] Simon Haykin “Kalman filtering and neural networks”

[5] Simon Haykin “Neural networks and learning machines”

[6] Charles Geyer “Markov chain Monte Carlo lecture notes”

[7] Paul Atzberger “The Monte-Carlo Method”

[8] Simon Julier and Jeffrey Uhlmann “A general method for approximating nonlinear transformations of probability distributions”

[9] Simon Julier, Jeffrey Uhlmann, Hugh Durrant–Whyte “A new method for the nonlinear transformation of means and covariances in filters and estimators”

[10] Julier, Simon J.; Uhlmann, Jeffrey K. "A new extension of the Kalman filter to nonlinear systems", 1997

[11] Eric Wan and Rudolph van der Merwe “The Unscented Kalman Filter for Nonlinear Estimation”

[12] van der Merwe, de Freitas, Doucet, Wan ”The unscented particle filter”, 2001

[13] Arnaud Doucet, Adam Johansen “A Tutorial on Particle Filtering and Smoothing: Fifteen years later”

[14] Simon Julier “The scaled unscented transformation”

[15] Grewal, Andrews “Kalman filtering. Theory and practice using Matlab”

[16] Anderson, Moore “Optimal filtering”

[17] Crassidis, Junkins “Optimal estimation of dynamic systems”

[18] Bain, Crisan “Fundamentals of stochastic filtering”

[19] Stephen Marsland “Machine learning. An algorithmic perspective”

[20] Snoek, Larochelle, Adams “Practical Bayesian optimization of machine learning algorithms”

[21] Rasmussen, Wiliams “Gaussian processes for machine learning”

[22] Chow, Ferrer, Nesselroade "An unscented Kalman filter approach to the estimation of nonlinear dynamical systems models"

[23] Klaas, de Freitas, Doucet "Toward Practical N^2 Monte Carlo: the Marginal Particle Filter", 2012

[24] Zhe Chen "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond"

[25] anuncommonlab.com, github.com/tuckermcclure/

[26] dynare.org

[27] github.com/zlongshen/ReBEL

[28] cs.ubc.ca/~nando/software.html

[29] stats.ox.ac.uk/~doucet/smc_resources.html

[30] John Hull "Options, futures, and other derivatives"

[31] O. Gueant, J. Pu "Mid-price estimation for European corporate bonds: a particle filtering approach", 2018

[32] Daniel Liberzon "Calculus of variations and optimal control", 2011

[33] A. Meucci "Review of Statistical Arbitrage, Cointegration, and Multivariate Ornstein–Uhlenbeck", 2009

[34] Peter Forsyth "A Hamilton Jacobi Bellman approach to optimal trade execution", 2010

[35] J. Wang, P. Forsyth "Numerical Solution of the Hamilton-Jacobi-Bellman Formulation for Continuous Time Mean Variance Asset Allocation", 2008

[36] Ulrich Rieder, Nicole Bäuerle "Portfolio optimization with unobservable Markov-modulated drift process", 2016

[37] Dang, Forsyth "Better than pre-commitment mean-variance portfolio allocation strategies: a semi-self-financing Hamilton-Jacobi-Bellman equation approach", 2015

[38] M.R. Niavarani, Alan J.R. Smith "Modeling and Generating Multi-Variate-Attribute Random Vectors Using a New Simulation Method Combined with NORTA Algorithm", 2012

[39] S. Albosaily, S. Pergamenshchikov "Optimal investment and consumption for Ornstein-Uhlenbeck spread financial markets with logarithmic utility", 2018

[40] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishechenko "The Mathematical Theory of Optimal Processes", 4th ed., 1983

[41] A. P. Kartashev, B. L. Rozhdestvensky "Ordinary differential equations and fundamentals of the calculus of variations", 3rd ed., 1986

[42] R. Bellman "Dynamic programming", 6th printing, 1972

[43] N. Stokey "The Economics of Inaction. Stochastic Control Models with Fixed Costs", 2009

[44] H. J. Kappen "Stochastic optimal control theory", 2008

[45] T. Ho, H. Stoll "Optimal dealer pricing under transactions and return uncertainty", 1980

[46] J. Yong, X. Y. Zhou "Stochastic Controls. Hamiltonian Systems and HJB Equations", 1999

[47] J. Granada, L. Galvez, O. Vasyunkina "From Black-Scholes to Hamilton-Jacobi", 2018

[48] S. Albosaily, S. Pergamenshchikov "Optimal investment and consumption for pairs trading financial markets on small time interval", 2018

[49] M. Avellaneda, S. Stoikov "High-frequency trading in a limit order book", 2008

[50] O. Guéant, C. Lehalle, J. Tapia "Dealing with the Inventory Risk. A solution to the market making problem", 2012

[51] P. Forsyth "A Hamilton Jacobi Bellman Approach to Optimal Trade Execution", 2010

[52] U. Rieder, N. Bäuerle "Portfolio Optimization With Unobservable Markov-Modulated Drift Process", 2005

[53] S. Kilianova, D. Sevcovic "Expected Utility Maximization and Conditional Value-at-Risk Deviation-based Sharpe Ratio in Dynamic Stochastic Portfolio Optimization", 2018

[54] E. Porteus "Foundations of Stochastic Inventory Theory", 2002

[55] P. Forsyth, G. Labahn "Numerical Methods for Controlled Hamilton-Jacobi-Bellman PDEs in Finance", 2007

[56] F. Sayas "A gentle introduction to the Finite Element Method", 2008

[57] W.Press, S.Teukolsky, W.Vetterling, B.Flannery "Numerical Recipes", 3rd ed., 2007

[58] C.Bellei "Crank-Nicolson scheme", 2016

[59] "FiPy: A Finite Volume PDE Solver Using Python", https://www.ctcms.nist.gov/fipy/examples/README.html

[60] B.Law, F.Viens "Market Making under a Weakly Consistent Limit Order Book Model", 2019

[61] P. Fodra, M. Labadie "High-frequency market-making with inventory constraints and directional bets", 2012

[62] Clustering benchmarks, https://github.com/deric/clustering-benchmark/

[63] A. Ng, M. Jordan, Y. Weiss "On Spectral Clustering: Analysis and an algorithm", 2001

[64] F. Chung "Spectral Graph Theory", 1997

[65] Daniel Spielman "Spectral Graph Theory"

[66] M. Mahoney "Lecture Notes on Spectral Graph Methods", 2016

[67] B. Gonzalez "An algorithmic introduction to clustering", 2020

[68] M.Ester, H.Kriegel, J.Sander, X.Xu "A Density-Based Algorithm for Discovering Clusters", 1996

[69] E. Schubert, S. Hess, K. Morik "The Relationship of DBSCAN to Matrix Factorization and Spectral Clustering", 2018