

Chapter 2

Using the Method of Simulated Moments for System Identification

Mohammad S. Jalali*, Hazhir Rahmandad, and Hamed Ghoddusi

* jalali@mit.edu

Increasingly, dynamic modelers face problems where estimating model parameters from numerical empirical data is a requirement. This trend is partly motivated by increasing availability of numerical data from a large number of ongoing and one-off data collection projects that survey different concepts of interest to dynamic modelers, from individuals and firms to disease incidences and measures of economic performance, just to name a few. For example, as of February 2015, the Data.gov portal contains data on over 138,000 machine-readable data sets. Another driver of this trend is the increasing application of dynamic models, beyond case-specific corporate projects, to theoretical and academic problems (Repenning 2003; Sterman 2006). In these cases, generic models for a category of objects (e.g., individuals, firms, and countries) are desired. Parameterizing such models requires specifying the different parameters that quantify similarities and differences across different objects, a goal often dependent on using robust and replicable parameter-estimation procedures. In fact, in light of the rapid growth and dissemination of improved parameter-estimation methods for model calibration, hypothesis testing, and policy recommendation in social and behavioral disciplines, continued relevance of any modeling subdiscipline may partially be tied to its ability to remain up to date with the best available tools in this domain.

Closely tied to the advances of the digital computer revolution, the field of system dynamics (SD) at its inception was ahead of many approaches available in social sciences in using advanced analytical methods (Forrester 1961), and kept this edge for many years. For example, advanced filtering and estimation methods were introduced into the SD literature in the 1970s (Peterson 1975). However, over the last three decades the research in SD has largely focused on diverse applications of the original

toolbox, with limited methodological expansions in the parameter-estimation domain. In contrast, research in econometrics and other related fields has provided many relevant tools over this period (Greene 2012). As a result, many dynamic modeling studies do not currently report formal parameter estimates common in social science research, or, when calibration is pursued, typical measures of confidence in estimated parameter values are not reported. While formal parameter estimation may not be feasible for many modeling problems, expert dynamic modelers should be equipped with the relevant tools when numerical data is available, when model purpose requires reliable parameter estimates, or when the audience requires formally estimated parameters and confidence intervals.

Hand-calibration is commonly practiced for assigning parameter values (Lyneis and Pugh 1996). When formal estimation procedures are used, modelers typically compare time-series data against the same variables in a model, and minimize the weighted sum of a function of the error term by changing the uncertain parameters until best-fitting estimates are found through a nonlinear optimization algorithm (Oliva 2003). The error function is frequently defined as the squared error, but absolute error and absolute percent error terms are also common (Stermann 2000). Weights for different data points are given based on the confidence the researcher has in the accuracy of the data and its relevance to the problem at hand. When reported, confidence intervals are calculated using normality and independence assumptions for error terms, which, with weights proportional to the reciprocal of error variance, would turn least-squared error estimates into maximum-likelihood estimates (MLE). Bootstrapping methods are also sometimes used for estimating confidence intervals (Dogan 2007). While these approaches cover many important estimation challenges, they each include some shortcomings. Ad hoc selection of the error term and the weights for different data points reduces the consistency of the methods and their ability to provide confidence intervals. Normality and independence may regularly be violated, which negate the benefits of MLE when using squared errors. Bootstrapping, while flexible, increases the computational costs significantly and as a result may prove infeasible for many realistically sized problems. Finally, the majority of these methods rely on having time-series data, and cannot extract from distributions in cross-sectional data the dynamics that have led to those distributions. In general, the estimation procedures ideal for dynamic modelers have the following characteristics.

Model independence. Given that most dynamic models do not follow a fixed structural form (e.g., linearity), estimation procedures that are independent of model structure are most beneficial.

Analytical confidence intervals. The ability to find confidence intervals analytically is important because of the computational costs of optimizing nonlinear dynamic models and replications needed for bootstrapping methods.

Assumption-free error terms. Independence and distributional assumptions on error terms for dynamic models are not always easy to justify, so methods with fewer such assumptions are preferred. Applicability to diverse data types. Both time-series and cross-sectional data are included. No single method fully satisfies all these requirements. Therefore, modelers need to choose from a menu of available estimation methods to match their problem requirements.

In this chapter, we offer an introduction to the method of simulated moments (MSM) for application to dynamic modeling problems. The basic idea of this method is to define appropriate moments of data and, by changing uncertain parameters, minimize the difference between those moments and their simulated counterparts resulting from the model simulations. Moments are statistics that are calculated using some function that gets as input the empirical data points available and provides as output a single numerical value. For example, population mean could be a moment, getting the individual data points as input and generating the population mean as the output. While any function can be used for generating moments, for analytical confidence intervals to be available, one needs these moments to be normally distributed, often meaning that each moment is an average across a function of multiple independent observations coming from the same underlying distribution (then normality follows from the central limit theorem). In practice, those observations (that feed into the moments calculations) are picked either from time-series data when a system is in steady state (e.g., stock prices over time), or at similar points in the life of similar units of observation (e.g., all 5-year-old individuals in a country). Typically, randomness plays an important role in how these units have ended up with different observed values (e.g., different weights for similarly aged individuals). As a result, the MSM is best fitted for dynamic modeling problems when some of the following problem characteristics are present:

- Population data. The MSM is suitable for estimation of generic models to population data. Different units of data such as individuals, firms, and countries could be available. For each unit one or more data items (e.g., weight, height, and age for individual data) could be available.
- Role of random processes. The MSM could be a good choice when models include stochastic processes that drive the model, and their impact on the model behavior is reflected in the data against which the model is to be calibrated (e.g., when we are trying to match the variance observed across multiple units).
- Cross-sectional data. The MSM applies to both cross-sectional and time-series data. Whereas time-series data include multiple data points for the same unit over time, cross-sectional data includes data points for multiple units at the same time. The MSM may be the only viable choice for estimating dynamic models when data is cross-sectional, since it allows us to extract the information about the historical trajectories of units hidden in their cross-sectional distributions.

- Confidence intervals. The MSM would be a suitable choice when analytical confidence intervals are sought.

The material in this chapter complements the chapter on MLE. Both are parameter-estimation methods. However, MLE requires distributional assumptions on error terms. When such assumptions are justified (either empirically, based on the observed error, or based on theoretical considerations), MLE provides a more efficient estimation method than the MSM. When error terms do not follow any well-established distribution, the MSM may be a better choice. An alternative method, maximum simulated likelihood, extends MLE's basic premise to situations when error terms do not follow predefined distributions. We do not discuss the similarities and differences between these methods, but refer interested readers to Adda and Cooper (2003) and Gouriéroux and Monfort (1996).

Consider an example from the dynamic modeling literature. While the MSM has become a major econometrics tool in the past two decades, it has rarely been applied in the system dynamics literature. Barlas (2006), in the design of the behavior pattern testing (BTS II) approach and software, uses some of the basic ideas of the MSM to match moments of the model against data, but does not draw on the MSM literature or discuss issues related to confidence levels. Rahmandad and Sabounchi (2011) adapt the MSM to estimating the parameters of an individual weight gain and loss model. In this section we provide a brief overview of their application to provide a more concrete example of the use of the MSM. A simple model of an individual's body mass, consisting of fat mass and fat-free mass, was developed. The model included a few uncertain parameters. In the absence of time-series data, those parameters were estimated from cross-sectional data on individual weights from the National Health and Nutrition Examination Survey (NHANES). The NHANES 2005–2006 population of 5,971 subjects was categorized into 110 subpopulations based on different ethnicities (5 ethnicities), genders (2 genders) and ages (11 age groups). For each population group, two moments, average body weight and variance of body weight, were calculated as the moments to be matched, leading to a total of 220 moments to match.

On the other hand, the model was replicated (using subscripts in Vensim¹ software) for 5,971 instances that matched the demographic characteristics (age, gender, and ethnicity) of the NHANES sample of year 2006. Initial body weight and fat fraction for these individuals were drawn from distributions of another NHANES sample of 1999–2000. Note that each round of NHANES uses a sample different from other rounds; thus, we cannot track the same individual over time, and the data is cross-sectional. The model was then simulated to grow this synthetic population from their initial age in year 2000 to their final age (consistent with the NHANES sample) in year 2006. Mean and variance of weight for different subpopulations in the simulated population were calculated for year 2006 and compared against the 220 moments coming from the data. The weighted sum of squared errors was calculated using

weights of the reciprocal of variance in each moment, itself calculated using variance and kurtosis of different moments. This error was minimized by changing 17 uncertain parameters using the Vensim internal optimization engine. The estimated parameters provided the minimum error. As a result, the authors were able to estimate a dynamic model, including individual growth mechanisms, from cross-sectional data with individuals in different age groups.

While this application follows the basic ideas of the MSM, it has some differences from the canonical MSM procedure. First, in this application the number of moments (220) was larger than many typical applications, in which the number of moments and parameters to be estimated are of the same order of magnitude. Second, given the computational costs in this setting, each moment was only simulated once, whereas typically multiple simulations, using different noise seeds, provide the estimation for the moment before it is compared with data. Finally, confidence intervals were not reported in this application.

Historical Background

The MSM is an offspring of the method of moments. Here we provide a quick review of this method and its basics, and refer interested readers to common econometric handbooks for further details.

Method of Moments

As a classical estimation method in statistics, the method of moments (MM) is based on finding unknown parameters of a certain distribution by relating these parameters to the moments of the distribution and then using empirical moments (obtained from data) to back up the unknown distribution parameters. We explain this using a few examples.

Example 1: Normal Distribution

The most convenient (and a straightforward) example for MM estimation can be expressed using a case for the normal distribution. Suppose you have collected a large sample of independent and identically distributed (i.i.d) observations for an experiment (e.g., a sample of heights of individuals in a country, y_i). Let us assume that we are confident the true (or the best-fitting) functional form for the distribution is normal, $N(\mu, \sigma^2)$. However, we do not know the values for the mean (μ) and variance (σ^2) of this distribution to fully characterize it.

Estimating the mean is easy since we can rely on the law of large numbers (LLN), which suggests that the mean of a large sample of trials will converge to its true mean. So we can simply calculate

$$\mu_s = \frac{\sum_{i=1}^n y_i}{n}$$

and use it as the best estimator of μ . Now we need to estimate the variance. Remember the

formula for variance: $\sigma^2(X) = E(X^2) - (E(X))^2$. Here we know $E(X)$ from our estimation of mean.

Moreover, we can calculate $E(X^2)$ using our sample $E(X^2) \approx \frac{\sum_{i=1}^n (y_i^2)}{n}$. Plugging these two values back

into the variance equation, we obtain $\sigma^2 = \frac{\sum_{i=1}^n (y_i^2)}{n} - \left(\frac{\sum_{i=1}^n y_i}{n}\right)^2$. Note that our ability to estimate the two

parameters here is dependent on knowing the analytical formulas that specify the unknown parameters (μ, σ^2) as a function of quantities that can be directly measured.

Example 2: Binomial Model

Now suppose that our data points are drawn from a binomial distribution $B(p, n)$ in which p is the probability of success and n is the number of trials. Let us assume we have several observations from this distribution, but do not know the value of parameters p and n . We know that the mean and variance of a binomial distribution is given by $\mu = np$ and $\sigma^2 = np(1-p)$. Similar to the previous example, we can calculate the first and second moments of the data ($E(X)$ and $E(X^2)$).

Using the formula for the variance, $\sigma^2(X) = E(X^2) - (E(X))^2$, we express the second moment of data using the parameters of the binomial model, $E(X^2) = np(1-p) + (np)^2$. Therefore, we can now use the first- and second-moment

equations together to provide a system of two equations and two unknowns that can be solved to recover (n, p) . Specifically, $p = 1 - \frac{E(X^2) - E(X)^2}{E(X)}$ and $n = \frac{E(X)}{p}$, and equations for $E(X)$ and $E(X^2)$ are given

above.

These two examples provide some intuition concerning the merits and difficulties of MM techniques. While for certain probability distributions MM can be used to recover parameter values through analytical expressions, it faces two major challenges. First, we need to know the true functional form of the distribution of outcomes. Second, we should be able to express the parameters of the distribution in terms of the data moments, a task only feasible for a small set of probability distributions. For many distributions, we cannot find an analytical (closed-form) solution to relate moments to parameters. Realistic dynamic models usually do not have an analytical solution to relate the output of the

model to its structural parameters. Therefore, the classical method of moments discussed above is not directly applicable to these models.

From Method of Moments to Method of Simulated Moments

Mcfadden (1989) was the first to propose using simulation to find moments as a function of model parameters, instead of trying to solve the moment conditions analytically. His paper was focused on discrete-response models (multinomial Probit); however, he provided theoretical foundations for more general models. Mcfadden (1989) works on the basis that an unbiased simulator can be used to generate a sample of moments given a set of parameters, that the simulation errors are independent across observations, and that the variance in estimates of moments will be normally distributed due to the law of large numbers operating across simulations of those moments. Lee and Ingram (1991) and Duffie and Singleton (1993) extended the framework and provided a rigorous treatment of the MSM estimators for time-series and panel-data cases and provided relevant statistics for making tests. Duffie and Singleton (1993) showed that the MSM estimator is, under regularity conditions, consistent and asymptotically normal.

Since then, the MSM has been widely used in various subfields of economics such as finance (both asset-pricing and corporate finance), macroeconomics, industrial organization (IO), international trade, and labor economics. Novales (2000) and Ruge-Murcia (2012) provide a useful review of MSM methods for estimating macroeconomic and dynamic stochastic general equilibrium (DSGE) models and show various examples step by step.

Basics of the Method of Simulated Moments

Let us go back to the body weight example discussed in the introduction. Suppose you have built a model that captures the dynamics of people's body weight as a function of their initial weight, eating and physical exercise habits, genetics, age, gender, and other fixed and time-varying characteristics.

People differ both in terms of their idiosyncratic characteristics (genetics, initial weight, etc.) and environmental factors (e.g., quality of food, cost to exercise, social eating habits, etc.). By changing initial conditions and model parameters, one will get different dynamic paths for an individual's weight as a function of her age. Suppose we have data on the weight of several children of age 10 (our initial value) as well as samples of children of ages 11 and 12, one and two years later. Further, assume that we are interested in estimating a structural parameter (e.g., average weight growth per year), which determines the weight path as a function of initial weight.

By fixing this (unknown) parameter to an initial value and simulating the model with all empirical values for the initial weight (age 10), we will generate different paths of weight-age for a simulated population the same size as the number of subjects in our data set. Now we can compare the distribution of model-predicted weight profiles at ages 11 and 12 against the empirical distributions. Specifically, we can compare the mean and variance of weight for a simulated population at ages 11 and 12 against the mean and variance at the same ages observed in the data. It is likely that our initial choice for the structural parameters leads to mean and variance weights different from those observed in the data. However, these simulated moments are a function of the parameter. By changing the structural parameter of the model, we will change both the mean and the variance of simulated weight values. We can therefore use an optimization method to search for the parameter value that minimizes the difference between model-generated mean and variance and their empirical values over all available moments (i.e., mean and variance at ages 11 and 12). This is the core idea behind the method of simulated moments: we simulate the moments of the model to find simulated counterparts for observed data, then change the structural parameters until the simulated moments match the observations as closely as possible.

Formal Definitions

The core of the MSM is to minimize the (weighted) difference between the empirical and simulated moments by changing the unknown parameters. In general, we can assume that our empirical data $\{y_i\}_t, i \in \{1, \dots, N\}, t \in \{1, \dots, T\}$ are observed for N different agents in T conditions or instances (e.g., T different times). To apply the MSM, first we need to calculate the empirical moments. For those empirical data in each population group of agents (e.g., profits of N firms each 3 years old, or body weights in a sample of 11-year-old children in the body weight example), different moment conditions (such as mean, variance, etc.) can be calculated. Those moment functions (sometimes called descriptive statistics) can be put together as different elements of a vector of empirical moments M_D . M_D is a vector $\langle p \times 1 \rangle$, where p is the number of moment functions and the p^{th} element of M_D vector would be calculated using function m_p : $m_p(\{y\}_t)$.

For instance, let us say that mean is the first moment condition (average) for N agents at time t , hence the respective element in the vector of empirical moments M_D is $m_p(\{y\}_t) = \frac{1}{N} \sum_{i=1}^N \{y_i\}_t$. Step one in the recipe for MSM discusses how to choose moment functions. Since we only have access to a sample of

data for estimating moments, the true moments of population from which the data sample is collected are approximated by empirical moments M_D .

Now we need to determine simulated moments \tilde{M}_s to be matched against the M_D . Consider a fully specified model—that is, a model that can be simulated given a set of parameter values. Assume there are d unknown parameters (θ) we are interested in estimating; that is, θ is a $\langle d \times 1 \rangle$ vector. The true functional form of the system's dynamics that leads to output $\{y_i\}_t = g(\cdot)$ is approximated by the model's output $\{\hat{y}_i\}_t = \hat{g}(\cdot)$. The output of the model is a function of known parameters vector C , unknown parameters vector θ (to be estimated) and random inputs ξ . Choosing different values for ξ will generate different values for $\hat{g}(\cdot)$. We assume that the model is correctly specified so that $\hat{g}(\cdot)$ is an unbiased estimator of the true model $g(\cdot)$, such that:

$$E(\hat{g}(C, \theta, \xi)) = E(g(\cdot)) \quad (1)$$

This ensures that if we generate a large enough (K) sample of outputs using a true random stream of inputs ξ , the arithmetic average of the model output should generate a reasonable approximation of the real-world processes that generate the observations:

$$m_p(\hat{g}(\cdot)) \approx \frac{1}{K} \sum_{n=1}^K m_p(\hat{g}(C, \theta, \xi^{(n)})) \quad (2)$$

\tilde{M}_s is then the vector of simulated moments, elements of which are calculated using Equation (2), in parallel with the calculations for the empirical moments (M_D). The averaging component in Equation (2) is necessary to make sure that the moments we estimate from the model are least affected by sampling bias. Ruge-Murcia (2012) shows that relatively small K values are sufficient for accurate estimates; however, we need to use a much larger number of replications to estimate asymptotic properties of the MSM (e.g., to calculate confidence intervals; see step 2 in the recipe for MSM).

A necessary (but not sufficient) condition for being able to identify the model is $p \geq d$. Otherwise, we will have an unidentified model with free parameters and the unknowns will not be uniquely determined. Ideally, we need more moment functions than the unknown parameters ($p \gg d$), which is called overidentification. Overidentification allows us not only to estimate the parameters but also to

evaluate the model's overall goodness of fit. A test of overidentification is discussed in step 5 of the recipe for MSM.

The core objective of the MSM is to minimize the (weighted) difference between the empirical and simulated moments by changing the unknown parameters. The estimated parameter set is the value of parameters that minimizes this difference. Specifically, with vector of simulated moments \tilde{M}_s consisting of $m_p(\hat{g}(\cdot))$ elements and the $\langle p \times p \rangle$ matrix W for weighting the moment conditions, unknown parameters are estimated as:

$$\hat{\theta} = \operatorname{argmin} (\tilde{M}_s - M_D)' W (\tilde{M}_s - M_D). \quad (3)$$

Note that \tilde{M}_s and M_D are both vectors $\langle p \times 1 \rangle$. Calculation of W is discussed in step 2 of the recipe for MSM.

Recipe for MSM

Step 1: Choose the Moment Conditions

The first step is to determine which moment conditions to use. Usually, the first and second moments of a model's outcomes (mean and variance) are good candidates. Remember that the number of moment conditions (p) should be (equal to or) larger than the number of unknown parameters (d). Thus, depending on the number of parameters you should decide to use informative moment conditions. The most informative moments are the ones that: 1) are sensitive to at least one of the unknown parameters (i.e., if we do a sensitivity analysis on unknown parameters, the moment changes significantly when changing at least one of the parameters); 2) have rather small variances, since the larger the variance of a moment (across multiple simulations), the more noisy, and thus less informative, it is for estimation. For more discussions on which moments to match, see Gallant and Tauchen (1996).

In addition to single-variable moments (e.g., mean and variance of one variable), one can also try cross-variable moments such as the correlation/covariance between two output variables, autocorrelation of a variable with itself, skewness of a variable, and so on. For instance, Nikolov and Whited (2009) use the MSM to quantify the magnitude of agency conflicts in corporate finance. They use 16 moments to identify the model, which includes first and second moments as well as first-order autoregression of firm's investment, profits, and cash holdings. Franke (2009) also uses the MSM to estimate an agent-based asset pricing model. He uses mean and variance of returns and several short, medium, and long lags

of autocorrelations between returns. Moments that are asymptotically normally distributed (typically due to the central limit theorem, when a moment can be seen as a summation or average over a large number of statistics) are more useful due to their asymptotic properties, which allow us to calculate the confidence intervals directly. If moments are not normally distributed, then calculation of confidence intervals will require bootstrapping or other nonparametric methods.

Conditions for Identification

The right choice of moment conditions is the most crucial step in identifying the model and recovering model parameters. Identification of a model using the MSM requires that the model-generated moment conditions fit their empirical counterparts if and only if the structural parameters equal their true values. Otherwise, the model will generate spurious results. Furthermore, the sufficient condition for identification is a one-to-one mapping between the structural parameters and a subset of the moment restrictions of the same dimension. Because dynamic models often do not yield such a closed-form mapping, to help ensure an identified model, one should choose moments that are sensitive to variations in the structural parameters. One way to check this is to run one-dimensional sensitivity analyses on each parameter being estimated, and check if selected moment conditions vary substantially with changes in parameter values. If a moment condition does not vary much or if its response to changes in parameters is not smooth and monotonic, then using that moment may not be very informative and may even cause the optimization engine to stop in a local optimum or never converge to an optimal solution.

Figure 2.1a shows examples of informative and noninformative moment conditions. The moment specified by the solid line moves smoothly as the unknown parameter changes and has a unique, well-defined value for each parameter value. Therefore, minimizing the distance between this function and the empirical moment will generate a unique parameter value. On the other hand, the moment represented by the dashed line is not informative. It is not very sensitive to changes in parameter value. We cannot even be sure that these small changes are due to a true response of the model to various parameter values or are the artifact of computational or sampling errors (though if the graph is smooth, these conjectures will be less valid). Moreover, the moment shows multiple extreme values and thus minimizing its distance from the empirical value will not identify a unique value.

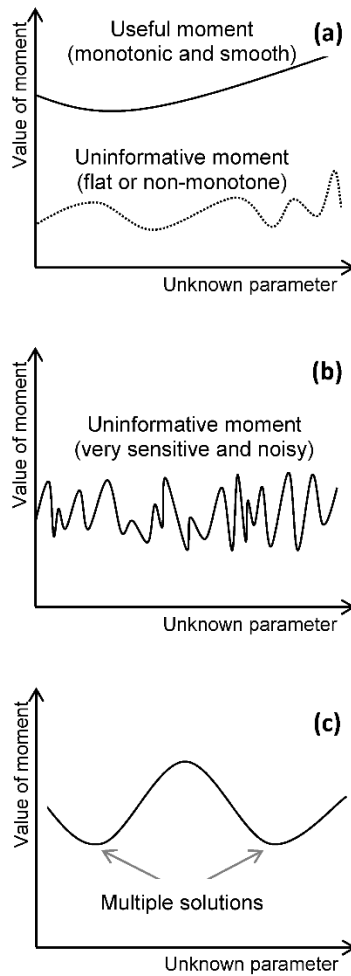


Figure 2.1: Relationship between moments and parameters. (a) Monotone versus nonmonotone moments. (b) Oversensitive nonmonotone moment. (c) Smooth but nonmonotone moment.

Figure 2.1b shows an example of a noisy (oversensitive) moment. In this case, the value of the moment changes very abruptly as the parameter value changes. From a theoretical perspective, it is harder to argue that a socioeconomic system will have such an oversensitive moment. Nonmonotone oversensitive moments also make it difficult to search for parameter values using numerical methods. Another issue with the choice of moments is the multiplicity of parameter values that make a good fit with the data. A sufficient condition for identifying parameters is the existence of a one-to-one relationship between parameter values and the moments, so if multiple parameter values fit the moment condition well, we may have a challenge with identifying the model (figure 2.1c).

Step 2: Weighting Matrix

Suppose you have d unknown parameters and exactly the same number of moment functions p . Under regularity conditions, we expect that these equations will yield the exact values of d unknowns. However, in the more common case when we have more moment functions than unknowns, it is very unlikely that all moment conditions will be precisely satisfied. This case is called an overidentification condition. In such a case, rather than solving a system of d unknowns and p moment functions, we need to conduct a minimization to get the simulated moments as close to empirical ones as possible. This raises the issue of weighting those moments; specifically, we need to use a symmetric, positive-definite weighting matrix W in Equation (3).

A two-step procedure should be used to calculate the optimal weighting matrix. The right choice of weighting matrix ensures that we get the smallest asymptotic variance for our estimated values and thus the tightest confidence intervals possible. In the first step a simple matrix is usually used as the initial value of the weighting matrix (e.g., $W = I$, the identity matrix). Using an identity matrix for weighting provides a consistent estimate but is not optimal. Not being optimal means that this weight will not generate the estimation results with the lowest possible variance. While popular in econometric applications, using an identity matrix may cause problems if the scales of different moments differ significantly. An identity matrix implies the application of the same error weight to all moments, which may unrealistically increase the importance of some moments over others. For instance, if one moment has uncertainty in the order of thousands and the other in the order of fractions, the first moment will dominate the optimization and the second moment will be practically ignored, which is not a good outcome. One way to overcome this problem is to use a scaling factor in the first step. We have found that weighting the error terms for each moment by the reciprocal of corresponding empirical moments—that is, using percentage error of moments instead of absolute error—provides a good alternative first step moment. To get percentage error, diagonal elements of W should be $1/(M_D)^2$ and the other elements should be zero.

Once the model is estimated using this initial W (first step estimates), the estimation can be repeated for a second step using a more efficient weighting matrix that is calculated using the estimated model from step 1. To estimate the optimal weighting matrix for the second step, we first need to estimate the variance-covariance matrix of simulated moments, \hat{S} :

$$\hat{S} = \frac{1}{L_1} \sum_{l_1=1}^{L_1} \left[m^{l_1}(\hat{g}(C, \hat{\theta}, \xi_1^{l_1})) - \frac{1}{L_2} \sum_{l_2=1}^{L_2} m^{l_2}(\hat{g}(C, \hat{\theta}, \xi_2^{l_2})) \right] \left[m^{l_1}(\hat{g}(C, \hat{\theta}, \xi_1^{l_1})) - \frac{1}{L_2} \sum_{l_2=1}^{L_2} m^{l_2}(\hat{g}(C, \hat{\theta}, \xi_2^{l_2})) \right]' \quad (4)$$

Note that in Equation (4) the function m^l includes the p dimensional output vector of simulated moments for replication l . To estimate \hat{S} in practice, we do the first-round estimation using Equation (3) in which W can be a diagonal matrix with elements $1/(M_D)^2$ (or alternative weighting options). Then a large number of replications are generated; the moments of interest are calculated for each replication, and the variance-covariance matrix of these moments are calculated using Equation (4). Note that for a good estimation of \hat{S} , we need a very large number of replications for L_1 and L_2 (practically, a number over a thousand is often large enough; Adda and Cooper 2003; Gouriéroux and Monfort 1996). This large number of replications is used only once and does not significantly increase the computational costs of the method. The optimal weighting matrix W^* will be the inverse of the variance-covariance matrix \hat{S} , $W^* = \hat{S}^{-1}$. By choosing this W^* , a new round of optimization using Equation (3) can take place to achieve a better estimate for $\hat{\theta}$.

The intuition behind this choice of weighting matrix is that the more noisy and uncertain a moment, the less weight we want to put on matching it. If an estimated moment is very uncertain (i.e., sensitive to the random inputs we do not control), its value in the empirical sample is likely to be far from the true population level value (i.e., if we had a very large sample). The weight of this moment should therefore be smaller in the optimization problem, as reflected in the larger corresponding element in the diagonal of matrix \hat{S} . On the other hand, those moments that are more robust against the choice of the sample will show smaller dispersion and will have small corresponding \hat{S} elements and (thus) larger W^* elements. Essentially, using the inverse of variance-covariance \hat{S} as W^* helps us give more weight to more robust moments and reduce the importance of those that change a lot from one simulation to another. Ruge-Murcia (2012) shows that the efficiency gained with the use of optimal W^* increases with the nonlinearity of the model. In fact, if you find that parameter estimates from step 2 are significantly different from those coming from step 1, it is recommended that you continue with additional iterations (three or more steps) until the estimated parameters converge.

For example, suppose that our data is for weights of 50 individuals ($N = 50$) and we want to use two moments, mean and variance. Running the model for 50 similarly parameterized individuals gives us 50 simulated observations for weight. We call this one round of simulation. Using these 50 model-generated data points, we can calculate the moments of interest: that is, the mean and the variance of simulated weights. Saving these moments and repeating the previous step, we can get a new set of model-generated moments (simulated moments). Notice that the only factor that changes between rounds of simulation is the random input ξ fed into the model; the structure of the model embedded in $\hat{g}(\cdot)$, the initial conditions, and the parameters C and $\hat{\theta}$ remain unchanged. After repeating the simulation step for, for example, 2,000 rounds ($L_1 = L_2 = 1,000$), we have generated 2,000 different observations for our moments, the mean and the variance of weight. We then calculate \hat{S} , the variance-covariance matrix of the estimated moments, which includes elements such as variance of mean, variance of variance, and covariance between mean and variance (a 2x2 matrix). Finally the inverse of \hat{S} gives us the optimal weighting matrix W^* .

Step 3: Initial Value and Simulation

For nonequilibrium models, we want to start the simulations from initial stock/state values as close to the empirical sample as possible. This reduces the error in simulated moments (compared to the empirical ones) that is due to differences in the initialization of the model from the real cases. In such models, we can only use the data points at times that correspond to the empirical sample. For example, we can only pick the data for 10-year-olds in our simulation model to compare with 10-year-olds in the empirical sample. For steady-state models, where all of the dynamics are in the steady state, we can be more efficient in the use of data generated by simulation. Essentially, rather than running the model N different times, we can use data from N subsequent time points coming from a single simulation, because the model is in a stochastic steady state and the differences across different points in time represent the steady-state distribution of the outcomes. If this process is pursued, we should discard early observations (in the time-series sense), since these observations are sensitive to the initial value of the model and are not in the steady state. If we need n observations over time, it is recommended to generate a vector of $2n$ observations over time and discard the first n observations. Note that these considerations are only relevant if the empirical sample is from a system that can be assumed to be in a steady state.

Step 4: Optimization Routine and Iteration

This is the most computationally intense step of the MSM procedure. We need to minimize the weighted distance of model-generated moments from empirical moments. More formally:

$$\theta^* = \operatorname{argmin} F = (\tilde{M}_S - M_D)' W^* (\tilde{M}_S - M_D) \quad (5)$$

We need to use numerical optimization routines to find the minimum of the total error function. A smart choice of initial values for parameters may significantly facilitate quicker convergence of the optimization routine. Any numerical optimization method requires a tolerance rule to stop. This is given as the error tolerance for the objective function $\|F^i - F^{i-1}\| \varepsilon_F$ as well as for the parameters $\|\hat{\theta}^i - \hat{\theta}^{i-1}\| \varepsilon_\theta$.

Similar to any nonlinear optimization routine, the MSM estimator may fall into the trap of a local optimum. Moreover, if some of the moment conditions are not very informative, they will have low sensitivity to parameter values and the problem may face a flat value function, which makes it very difficult to progress and converge. The MSM uses numerical methods to find the minimum of the objective function. Therefore, the results might be sensitive to initial values, the precision of the search algorithm (the level of error tolerance), and the quality of the algorithm to distinguish local and global extreme points. We recommend rerunning the optimization using distant initial values to check whether the results are sensitive to the choice of initial values for parameters.

Another important implementation concern is consistency in random number streams. To avoid introducing sampling error into rounds of simulation-optimization, we should work with the same sample of random numbers in each simulation-optimization step. This ensures that changes in results are due to changes in structural parameters and not the random sample.²

Step 5: Interpretation and Making Inference

The results we get from the previous steps are consistent point estimates of true parameter values. Confidence intervals of estimated values allow us to assess the accuracy of those estimates and the range of potential errors.

Variance-Covariance Matrix

In order to estimate the confidence interval of the estimated parameters, we need their distribution form and the parameters of those distributions. The simulated moments estimator is (asymptotically) normal when the original moments are (asymptotically) normal. This can be achieved through selecting moments that are calculated by summing some statistic over the N instances of agents (e.g., mean, variance, autocorrelation) and following the central limit theorem (normality follows as $N \rightarrow \infty$). In practice, much smaller samples (e.g., with $N > 30$) lead to results that are adequately close to normal. In this case, the asymptotic distribution of estimated parameters (θ^*) is normally distributed and given by:

$$(\theta^* - \theta) : \text{Normal}(0, Q) \quad (6)$$

The asymptotic variance-covariance matrix of parameters (Q) is given by the following formula:

$$Q = (1 + \frac{1}{K})[\hat{D}'\hat{S}^{-1}\hat{D}]^{-1} \quad (7)$$

\hat{D} measures the sensitivity of the moments to small changes of unknown parameter values, $\hat{D} = E_0(\frac{\partial m}{\partial \theta})$,

and can be calculated numerically by changing different model parameters with very small increments ($\partial \theta$) and measuring the impact on the moments (∂m). This procedure should be repeated multiple times and $\frac{\partial m}{\partial \theta}$ should be averaged over that sample; however, only a small number of replications is typically

enough to precisely estimate the $\frac{\partial m}{\partial \theta}$ (Gourieroux and Monfort 1996). In Equation (7), \hat{S} is the variance-covariance matrix of moments introduced in step 2 of the recipe for MSM for which large numbers of replications (L_1 & L_2) are required. Once the variance-covariance matrix for the estimated parameters (Q) is found, it is easy to find confidence intervals at different levels of confidence: for example, $\theta_i^* \pm 1.96q_{i,i}$ for the 95%³ confidence interval for parameter θ_i^* , using $q_{i,i}$, the i^{th} diagonal element of Q matrix. The variance-covariance matrix remains valid even if the moments are not asymptotically normal; in that case, calculation of confidence bounds using normality assumptions may include too much error, and bootstrap methods may be preferred.

Overidentification Test

When the model is overidentified ($p > d$), some of the moment conditions will be different from zero. This provides us with an opportunity to assess how well the estimated model matches the data. For a good model whose structure and parameter values are close to the true system, the value of these nonzero moment conditions should only be different from zero to the extent allowed by random error in the values of the empirical moments. If the difference between the empirical and simulated moments is larger than what is expected due only to randomness, then we can claim that the model has some systematic error (i.e., it does not match the true data-generating process very well). The J-test is used to quantify and assess the significance of this error. The J-statistic is given by:

$$J = \frac{K}{1+K} ((\tilde{M}_m - M_D)' W^* (\tilde{M}_m - M_D)) \sim \chi_{p-d}^2 \quad (8)$$

This statistic is distributed with chi-square distribution with $p-d$ degrees of freedom under the null hypothesis that the true data-generating process ($g(\cdot)$) is not different from the estimated model $\hat{g}(\cdot)$.

Other Methodological Considerations

The MSM is seen by some as a “black box” estimation method. It does not require the detailed consideration of model structure and error terms called for by likelihood-based methods. Moreover, it provides a large degree of freedom for choosing the moment conditions. By imposing different moment conditions, one may recover different structural parameters, and as long as there are enough meaningful moment conditions, the researcher has the luxury of using an arbitrary subset of moments. This is a risk if the researcher picks the moments that best serve his biases. Providing graphs showing the sensitivity of the moments to parameter values is one way to ensure that the moments are indeed informative.

Adding an additional parameter to be estimated is costly, in terms of both computation time and identification strategy. Thus, we suggest trying to estimate as many parameters as possible from other sources (e.g., review of literature, regressions, etc.) and leaving the minimum number of parameters for the MSM technique. Finally, following the reporting guidelines for simulation-based research (Rahmandad and Sterman 2012) is strongly recommended to ensure transparency and replicability of the results using the MSM.

An Applied Example

In this section, we provide a simplified example of using the MSM to estimate a dynamic model.⁴ Assume we have observations from a sample of $I=200$ firms. Our modeling project has allowed us to build a simple dynamic model of these firms. The firms have similar structures and parameters, but are exposed to different random shocks. Firms require resources (R) to produce outputs (O). The relationship between resources and outputs is given by a concave production function, $f(R)$. Concavity ensures that the rate of production grows more slowly than the rate of resource accumulation. Thus, no firm grows forever, and there is a steady-state point for each firm to stop growing.

The existing resources depreciate at a proportional rate (depreciation rate, D). Firms devote a fraction (β) of their output to investment (I), which increases their resources. Moreover, resource accumulation and depreciation are also subject to random shocks (ξ). Random shocks capture events such as sudden technological shocks, failures in adopting new technology inside the firm, shifts in customer tastes, and inherent risks of the production technology, among others. Random shocks are pink noise distributed normally, $\xi \sim N(0, \sigma_\xi^2)$, and with correlation time ρ_ξ as a half-year. Figure 2.2 shows this simple model of firms that go through a process of expansion and stabilization.

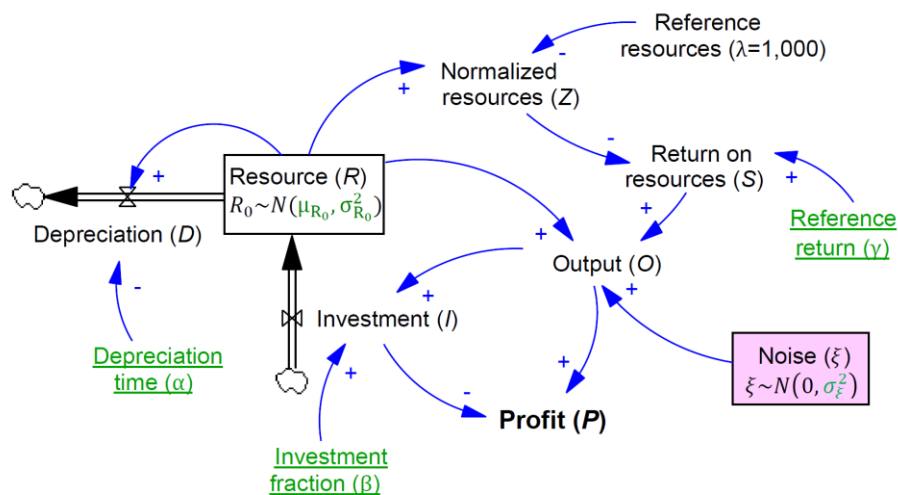


Figure 2.2: Dynamic model of firms (unknown parameters are gray). Vensim file is available in the online appendix.

Let us assume that we have data on a sample of 200 firms, all starting from $R_0 \sim N(\mu_{R_0}, \sigma_{R_0}^2)$ initial resources and growing in this market. Data include annual profit values for all the firms in this market; however, the specific firms are not identified, so we cannot connect the profit from one year to

that in the next year. We impose this additional restriction to increase the complexity of the estimation problem: if time-series data were available for the full panel of 200 firms, one could use calibration methods in which the behavior of a simulated firm is matched against the data from each firm, either multiple times or simultaneously for all the firms. Without firm IDs, all we have is the cross-sectional data on profits for each year, but we cannot tell if a firm that made a lot of money one year did well during the next. Given this data, we want to estimate the unknown parameters of our model, identified in Figure 2.2 and listed in Table 2.1.

Table 2.1: Unknown parameters in the model

Unknown parameters (θ)	Notation	Unit
Depreciation time	α	Year
Investment fraction	β	Dmnl*
Noise—standard deviation	σ_{ξ}	Dmnl
Reference return	γ	1/Year
Resource—mean (for initial resource)	μ_{R_0}	M\$
Resource—standard deviation (for initial resource)	σ_{R_0}	M\$

* Dimensionless.

Equations that identify the model are summarized in Table 2.2.

Table 2.2: Model's equations

Variables	Function (i^{th} firm)	Unit
Depreciation	$D_i = R_i / \alpha$	M\$/Year
Investment	$I_i = \beta O_i$	M\$/Year
Resources	$R_i = R_{0i} + (I_i - D_i)d_t$	M\$
Normalized resources	$Z_i = R_i / \lambda$	Dmnl*
Return on resources	$S_i = \gamma(1 - (Z - 1)^2)$	1/Year
Output	$O_i = R_i S_i (1 + \xi_i)$	M\$/Year
Profit	$P_i = O_i - I_i$	M\$/Year

* Dimensionless.

To ensure that we know the true model parameters, we generate our data by simulating the model once for a population of 200 firms for 10 years (a panel of 2,000 observations) using the parameters in Table 2.3. We then treat this data as input into an MSM estimation procedure, in which we treat these six parameter values as unknown and estimate them.

Table 2.3: Parameters to generate synthetic data

Parameter	True value
α	5
β	0.8
σ_ξ	0.1
γ	1.5
μ_{R_0}	200
σ_{R_0}	20

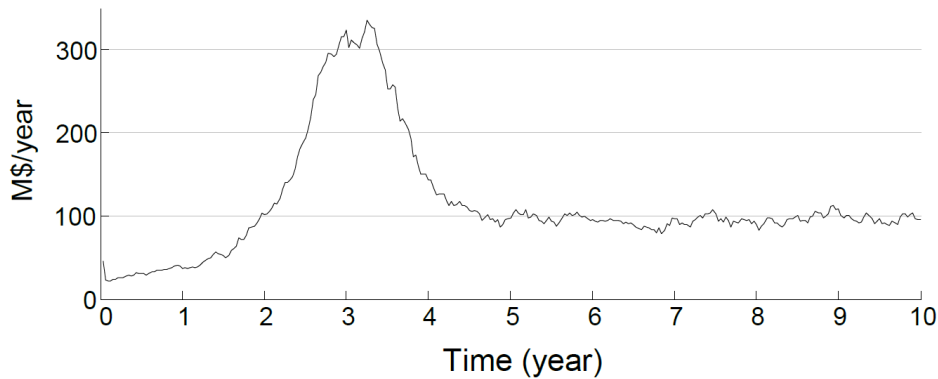


Figure 2.3: Profits of one of the 200 firms. Note that the actual data used for estimation does not include firm IDs; therefore, we cannot put together a similar time path for the growth of any single firm. The graph is provided only to offer an intuition about the type of behavior generated by this model.

Step 1: Choose the Moment Conditions

Following the recipe for MSM, we need to choose appropriate moments. Here we choose the first

moment $\mu_t = \frac{1}{200} \sum_{i=1}^{200} (P_i)_t$ and function $\mathcal{G}_t = \frac{\frac{1}{200} \sum_{i=1}^{200} (P_i^2)_t}{\mu_t}$ of profits of 200 firms per year t , for 10

years. Putting all 10 μ_t and 10 \mathcal{G}_t in one vector, M_D would be a vector of 20 elements. The reason that \mathcal{G}

is selected instead of the second central moment (variance) is to scale both μ and \mathcal{G} in a reasonably similar range (compare the vertical axes in Figure 2.4); however, it is not necessary to do so as long as a proper choice of W is selected—see the next step for more discussion. Note that the number of moments ($p=20$) is larger than the number of unknown parameters ($d=6$). Figure 2.4 shows selected moments of profits of 200 firms per year.

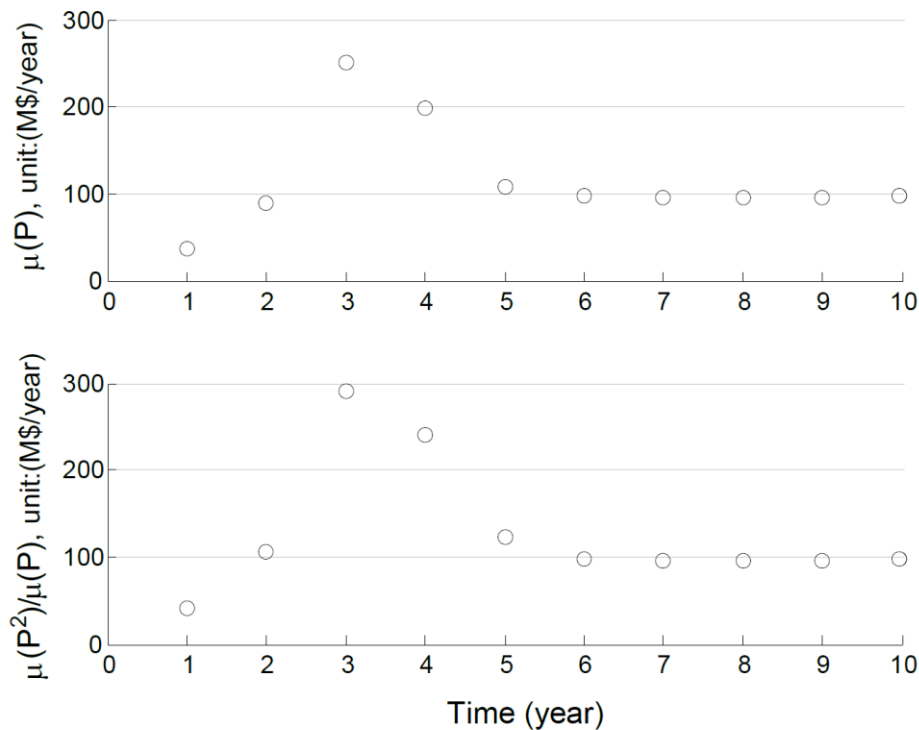


Figure 2.4: μ and $\mathcal{G} = \mu(P^2) / \mu(P)$ of profits of 200 firms (each dot represents a moment in year i).

Step 2: Weighting Matrix

In the first round of estimation, the $\langle 20 \times 20 \rangle$ weighting matrix is assumed diagonal with non-zero elements $W_{ii} = 1/(M_i)^2$. This W essentially minimizes the squared percentage error of moments, which is a good starting point but is not optimal. After running the first round of optimization and estimating $\hat{\theta}$, the W^* is approximated based on the inverse of the variance-covariance matrix of simulated moments, Equation (4). Note that the number of replications we use for estimation is $N=10$, while the number we

use in Equation (4) is $L_1 = L_2 = 1,000$. This larger number is required for getting reliable estimates for the covariance matrix.

Step 3: Initial Value and Simulation

Initial values of the stock resource in Figure 2.2 (R_0) are unknown. We estimate the distribution of R_0 by assuming that initial resources come from a normal distribution with two unknown parameters. Those parameters, μ_{R_0} and σ_{R_0} , are estimated as part of the estimation process. For any given values for these parameters, we draw 200 random numbers from the resulting distribution to initialize the simulated firms.

Step 4: Optimization and Iteration

Initial values of parameters to start the optimization could be arbitrary. Yet, if optimization restarts are limited due to computational costs, a reasonable set of initial values can help optimization not fall into local optima and will speed up convergence. We use the following initial parameters, which are not too far off the true value and simplify the optimization step:

$$\theta_0 = \{\alpha, \beta, \sigma_\xi, \gamma, \mu_{R_0}, \sigma_{R_0}\} = \{2, 0.5, 0.5, 5, 200, 10\}$$

We use *fmincon* solver in MATLAB to estimate the unknown parameters θ . The same random sample shocks are used in each optimization iteration to avoid sampling errors. Given W^* estimated in step 2, the optimization solver starts at θ_0 and, using numerical techniques, searches over the feasible space to find a minimum of the payoff, F , function of the MSM, presented in Equation (5).

Step 5: Interpretation and Making Inference

The MSM matches simulated moments \tilde{M}_S against empirical moments M_D by changing the unknown parameters θ . Estimated parameters in the first round of optimization (based on $W = 1/(M_D)^2$) and the second round of optimization (based on W^*) are presented in Table 2.4. Comparing estimated values with the true values in Table 2.4, the first round of optimization estimates $\alpha, \sigma_\xi, \gamma$ and μ_{R_0} very close to the true values, but it fails to estimate σ_{R_0} and also does not provide a good estimate for σ_ξ . The second

round of optimization provides relatively close estimates for all parameters. Figure 2.5 shows how simulated and empirical moments are matched at $\hat{\theta}^*$ after the second round.

Table 2.4: Estimated values and true values

Parameter	First optimization results	Second optimization results ($\hat{\theta}^*$)	True value
α	5.013	5.035	5
β	0.800	0.799	0.8
σ_ξ	0.209	0.100	0.1
γ	1.478	1.527	1.5
μ_{R_0}	201.213	196.140	200
σ_{R_0}	0.0003	19.702	20

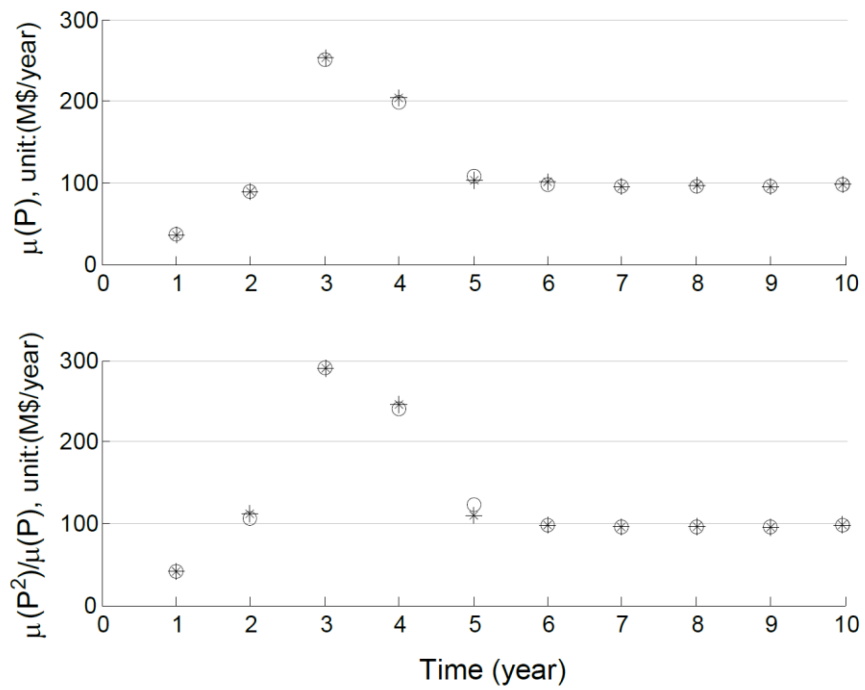


Figure 2.5: Empirical moments (○) and simulated moments (*).

To interpret the findings, we estimate the confidence intervals of estimated parameters $\hat{\theta}^*$. First we calculate the variance-covariance matrix Q of parameters using Equation (7). Given $\hat{\theta}^*$ and variances of $\hat{\theta}^*$ (which is the diagonal of the Q matrix), estimated parameters and the true values of those parameters are summarized in Table 2.5. Notice that 95% confidence intervals for all six parameters include the true values, and the estimated values are also very close to the true values. The J-test is calculated to be 5.95, which is significantly smaller than the 99% cut-off value for a chi-square distribution with $p-d = 20-6 = 14$ degrees of freedom ($J = 5.95 < \chi_{16}^2 = 29.14$), suggesting that the estimated model is not statistically distinguishable at a 99% confidence level from the true data-generating process (if $J > \chi_{p-d}^2$, the estimated model is systematically biased). This should not come as a surprise in this setting; after all, we have used the exact model that generated the “empirical” data to estimate the parameters in this exercise.

Table 2.5: Estimated values and true values

Parameter	Estimated value (std*)	95% Confidence interval	True value
α	5.035 (0.064)	[4.953, 5.117]	5
β	0.799 (0.002)	[0.797, 0.801]	0.8
σ_{ξ}	0.100 (0.014)	[0.082, 0.118]	0.1
γ	1.527 (0.023)	[1.498, 1.556]	1.5
μ_{R_0}	196.140 (3.132)	[192.127, 200.153]	200
σ_{R_0}	19.702 (0.980)	[18.446, 20.959]	20

std*: Standard deviation.

Finally, we test the reliability of the calculated confidence intervals by repeating the above procedure 100 times (using 100 different sets of data points, all generated by the same underlying dynamic model and only varying in their random noise streams). The results are summarized in Table 2.6.

Table 2.6: Percentage of estimated 80% and 95% confidence intervals including the true value over 100 estimations

Parameter	80% Confidence interval	95% Confidence interval
α	77%	90%
β	79%	92%
σ_{ξ}	84%	90%
γ	79%	89%
μ_{R_0}	81%	91%
σ_{R_0}	77%	90%

The values in Table 2.6 represent the percentage of 100 replications where the true value of the parameter lies in the estimated confidence interval. For instance, 79 runs out of 100 estimated an 80% confidence interval for β that included its true value. As shown in Table 2.6, the estimated 80% and 95% confidence intervals include approximately the same percentages of the true values over the 100 replications (i.e., 80% and 95%). The results are more precise for the 80% confidence intervals, compared to the 95% intervals. This indicates that the estimated parameters slightly diverge from the normal distribution and have somewhat fatter tails, which lead to tighter-than-real confidence intervals when using the normality assumption.

Exercise

In this section, we provide a simple exercise for practicing the use of the MSM in a dynamic setting. Assume we have data on a population of 1,000 individuals who are starting in a study and their initial body weights are normally distributed, with mean of 80 kg and standard deviation of 5 kg. The individuals are then monitored as they gain weight over time and the distribution of their weight changes. Both mean and standard deviation of the distribution increase over time, and the distribution may no longer be normal. In general, body weight changes when energy intake and energy expenditure are not balanced. When energy intake (kcal/year) from food and drink exceeds energy expenditure (kcal/year), body weight (kg) increases; otherwise, people lose weight (or remain at their current weight if energy intake and energy expenditure are equal). Energy expenditure depends on physical activity, basal metabolic rate (the energy consumed at complete rest), and digestion, but for simplicity in this exercise, we assume that energy expenditure is a fixed number (100,000 kcal/year) + weight (kg) \times energy cost of

weight (10,000 kcal/kg/year). We may not have direct measures of energy intake, but can approximate it by multiplying the energy expenditure by a dimensionless fluctuation factor. We can then quantify the fluctuation factor as a function of different inputs, such as random shocks, starvation or overfeeding feedback, and general trends in energy intake. Let us use the following assumptions in formulating the fluctuations of energy intake:

- For biological robustness, assume that energy intake never goes below the fixed energy expenditure (100,000 kcal/year) that we assume is the minimum for survival.
- Fluctuation factor changes around 1, and have the following components added to value 1.
 - A random shock factor that is a pink noise for each individual distributed normally with mean and standard deviation of 0 and 0.007, respectively, and with correlation time of 5 (years). Essentially, this noise factor recognizes that people eat more/less than their energy expenditure by chance, and there is some autocorrelation in how they change their eating behavior.
 - Energy intake balance (energy intake/energy expenditure), after a short first-order delay (delay time=0.033 year), generates either starvation or overfeeding feedback, such that:
 - if $(\text{delayed energy intake balance} - 1) \leq 1$, there is an overfeeding feedback, so the fluctuation rate is changed by adding $(\text{delayed energy intake balance} - 1) \times \text{overfeeding}$.
 - if $(\text{delayed energy intake balance} - 1) > 1$, there is a starvation feedback, so the fluctuation rate is changed by adding $(\text{delayed energy intake balance} - 1) \times \text{starvation}$.
 - Overfeeding and starvation are two unknown parameters.
 - Extra energy intake trend is assumed to be constant and the third unknown parameter.

We have generated data using the true model parameters by simulating the model once for a population of 1,000 individuals for 20 years, and we have saved only weights at years 1, 5, 10, 15, and 20 (you are provided with a panel of 5,000 observations in *MSM_Exercise_Data.xls* file). Note that the specific individuals are not identified, so you cannot connect the weights between the given years.

Based on the information above, the true model (which is discussed above and is also available in *MSM_Exercise_Model.mdl* Vensim file), and the data in *MSM_Exercise_Data.xls* file, leverage the MSM estimation procedure to find the three unknown parameters (overfeeding, starvation, and extra energy intake trend are unknown). See the Vensim model for more details of the model equations. Estimate the unknown parameters and report confidence intervals for estimated parameters along with J-test results for the validity of the model. In the online appendix, MATLAB codes and a sample solution are provided for this exercise.

Conclusion

Over the last three decades, the research in system dynamics has largely focused on diverse applications of the original toolbox, with limited methodological expansions in the parameter-estimation domain. In this chapter, we offer an introduction to the MSM for application to dynamic modeling problems. The basic idea of this method is to define appropriate moments of data and, by changing uncertain parameters, minimize the difference between those moments and their simulated counterparts coming from the model.

Given that most dynamic models do not follow a fixed structural form (e.g., linearity), estimation procedures such as the MSM that are independent of model structure are most beneficial. Moreover, independence and distributional assumptions on error terms for dynamic models are not always easy to justify, so the MSM, which has fewer such assumptions, is preferred. The MSM is especially useful when error terms do not follow any well-established distribution. It could also be a good tool when models include stochastic processes that drive the model, and their impact on the model behavior is reflected in the data against which the model is to be calibrated; for example, when we are trying to match the variance observed across multiple units.

The MSM is also applicable to diverse data types, including both time-series and cross-sectional data. When likelihood functions cannot be calculated or are too expensive to calculate numerically, the MSM may be the only viable choice for estimating dynamic models with cross-sectional data, since it allows us to extract the information about the historical trajectories of units hidden in their cross-sectional distributions.

Notes

1. Vensim software, ©2014 Ventana Systems, Inc.
2. The concept of “seed” in software packages such as MATLAB and Vensim controls random number generation. Using the same seeds across optimization iterations helps draw the same random samples.
3. The 2.5% and 97.5% critical values for standardized normal distribution are 1.96 standard deviation away from the mean.
4. The analysis is conducted in MATLAB, but could be also fully implemented within Vensim. The MATLAB implementation simplifies the optimization and estimation steps, while the Vensim implementation is easier for model building and presentation. Overall, we recommend MATLAB or other general computational packages for similar applications due to the efficiencies gained in the optimization step, which is the computational bottleneck for applying the MSM. Alternatively, one can use the Vensim or other simulation software’s connectivity features to couple them with MATLAB or other general computational software, conducting the simulations in the customized environment while running the MSM steps in general software.

References

- Adda, J., & Cooper, R. W. (2003). *Dynamic economics: quantitative methods and applications*. Cambridge, Mass.: MIT Press.
- Barlas, Y. (2006). *Model validity and testing in system dynamics: Two specific tools*. Paper presented at the 24th International Conference of the System Dynamics Society, Nijmegen, The Netherlands.
- Data.gov. (2013). a U.S. Government website. Retrieved 1 Nov, 2013, from <http://www.data.gov>
- Dogan, G. (2007). Bootstrapping for confidence interval estimation and hypothesis testing for parameters of system dynamics models. *System Dynamics Review*, 23(4), 415-436.
- Duffie, D., & Singleton, K. J. (1993). Simulated Moments Estimation of Markov-Models of Asset Prices. *Econometrica*, 61(4), 929-952.
- Forrester, J. W. (1961). *Industrial Dynamics* (1 ed.). Cambridge: The M.I.T. Press.
- Franke, R. (2009). Applying the method of simulated moments to estimate a small agent-based asset pricing model. *Journal of Empirical Finance*, 16(5), 804-815. doi: 10.1016/j.jempfin.2009.06.006
- Gallant, A. R., & Tauchen, G. (1996). Which moments to match? *Econometric Theory*, 12(4), 657-681.
- Gourieroux, C., & Monfort, A. (1996). *Simulation-based econometric methods*. Oxford ; New York: Oxford University Press.
- Greene, W. H. (2012). *Econometric analysis* (7th ed.). Boston: Prentice Hall.
- Lee, B.-S., & Ingram, B. F. (1991). Simulation estimation of time-series models. *Journal of Econometrics*, 47(2-3), 197-205. doi: 10.1016/0304-4076(91)90098-X
- Lyneis, J. M., & Pugh, A. L. (1996). *Automated vs. 'Hand' Calibration of System Dynamics Models- An Experiment With a Sample Project Model*. Paper presented at the the 14th International Conference of the System Dynamics Society, Cambridge, Massachusetts.
- Mcfadden, D. (1989). A Method of Simulated Moments for Estimation of Discrete Response Models without Numerical-Integration. *Econometrica*, 57(5), 995-1026.
- Nikolov, B., & Whited, T. (2009). Agency conflicts and cash: Estimates from a structural model.
- Novales, A. (2000). The role of simulation methods in Macroeconomics. *Spanish economic review*, 2(3), 155-181.
- Oliva, R. (2003). Model calibration as a testing strategy for system dynamics models. *European Journal of Operational Research*, 151(3), 552-568. doi: Doi 10.1016/S0377-2217(02)00622-7
- Peterson, D. W. (1975). *Hypothesis, estimation, and validation of dynamic social models- Energy demand modeling*. (Ph.D.), Massachusetts Institute of Technology, Cambridge, MA.
- Rahmandad, H., & Sabounchi, N. (2011). *Building and estimating a dynamic model of weight gain and loss for individuals and populations*. Paper presented at the the 29th International Conference of the System Dynamics Society, Washington, DC.
- Rahmandad, H., & Sterman, J. D. (2012). Reporting guidelines for simulation-based research in social sciences. *System Dynamics Review*, 28(4), 396-411. doi: Doi 10.1002/Sdr.1481
- Repenning, N. P. (2003). Selling system dynamics to (other) social scientists. *System Dynamics Review*, 19(4), 303-327. doi: Doi 10.1002/Sdr.278
- Ruge-Murcia, F. (2012). Estimating nonlinear DSGE models by the simulated method of moments: With an application to business cycles. *Journal of Economic Dynamics & Control*, 36(6), 914-938. doi: 10.1016/j.jedc.2012.01.008
- Sterman, J. (2000). *Business dynamics: systems thinking and modeling for a complex world* (1 ed.). Boston: McGraw-Hill/Irwin.
- Sterman, J. D. (2006). Learning from evidence in a complex world. *American Journal of Public Health*, 96(3), 505-514. doi: 10.2105/ajph.2005.066043