Modelling uncertainty in the production process

Takemyprocs

September 21, 2021

Abstract

We use probability theory and statistics to provide a framework to tackle uncertainty in production. In particular, we focus on the problem of getting an uncertain amount of product given a fixed amount of inputs. Therefore, we ignore the production process itself and treat it like it was a black box. We discuss uncertainty of the product distribution in terms of its mean and variance. Then, we apply this framework to intermediate inputs within a stochastic optimization problem.

1 A parametric model for stochastic products

Suppose we use a fixed integer quantity of a single input to procuce random integer quantities for n different products. Also, assume the maximum attainable quantity for every output is fixed at the integer value M. Therefore, every time we engage into the production process we end up with an output vector where each element $x_i \in \mathbb{Z}$ oscillates within the interval $[0, M]$. Although we can jointly model all products, we decided to work with a more parsimonious approach. Concretely, we assume that product outcomes processes are independent but not identical between them. If they truly are, we can safely ignore any correlation between product outcomes.

1.1 The categorical distribution

We model a representative product outcome as coming from a categorical distribution, a discrete probability distribution. The categorial distribution can be viewed as a special case of the multinomial distribution with a single drawing. The probability mass function (PMF) for this distribution is the following:

$$\begin{gathered} f(x|p_1, p_2, ..., p_M) = p_0^{\mathbb{1}[x=0]} p_1^{\mathbb{1}[x=1]} p_2^{\mathbb{1}[x=2]} ... p_M^{\mathbb{1}[x=M]} = \prod_{j=0}^{M} p_j^{\mathbb{1}[x=j]} \\ \sum_{j=0}^{M} p_j = 1 \end{gathered} \tag{1}$$

where the product x can take values from 0 to M, and $\mathbb{1}[\cdot]$ is the indicator function[1]. Also, for notation simplicity we define $p_j := \Pr(x = j)$. Some outcomes may be more

[1]This function takes the value of 1 when the argument is true, otherwise is equal to 0

likely to occur than others so in general p_j lies within the $[0, 1]$ range and all probabilities add up to 1. To this point, we must stress the fact that what we are indeed proposing is a finite count distribution to model the product quantity, customary choices are the binomial distribution or the beta-binomial distribution. However, such distributions imposes a common structure to all probabilities. Although we use the categorical distribution, we do not attemp to treat the outcome as a categorical variable but rather treat it as a integer or count variable with no a priori assumptions over the probability of ocurrence.

2 Estimation

Now we are ready to use real data in order to estimate the categorical distribution parameters (outcome probabilities). We adopt a maximum likelihood estimation (MLE) approach, that is we aim to find a set of parameter that maximize the likelihood of obtaining the real data. Before we derive the likelihood function for the categorical distribution, keep in mind that for a single event (production process) we can obtain 1 out of $M + 1$ results (from 0 to M) in terms of product, so we can arrange our data as follows:

$$\begin{array}{c} \\ i=1 \\ i=2 \\ \vdots \\ i=N \end{array} \begin{array}{c} \begin{array}{cccc} x=0 & x=1 & ... & x=M \end{array} \\ \begin{pmatrix} 0 & 1 & ... & 0 \\ 1 & 0 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 1 \end{pmatrix} \end{array}$$

2.1 Log-likelihood function

On every single event i we end up with one and only one of the $M + 1$ possible results. We use a binary indicator to highlight the outcome we obtained on that event. To deriver the likelihood function $\mathcal{L}(\cdot)$ we start by writing the PMF for the categorical function, but we instead treat data as given rather than parameters:

$$\mathcal{L}(p_0, p_1, ..., p_M|x) = \prod_{i=1}^{N} \left(\prod_{j=0}^{M} p_j^{\mathbb{1}[x_i=j]} \right) \tag{2}$$

Here we use x_i to denote the x outcome in the i-th event. Now, is customary to deal with the log-likelihood function instead of the likelihood one. The reason lies in the simplicity of computation once we use a mathematical program to estimate parameters, also recall that the log function is a monotone transformation so optimize $\log \mathcal{L}$ is equivalent to optimize $\mathcal{L}$.

$$\log \mathcal{L}(p_0, p_1, ..., p_M|x) = \sum_{i=1}^{N} \left(\sum_{j=0}^{M} \mathbb{1}[x_i = j] \log p_j \right) \tag{3}$$

We can expand this double summation to get the following expression of the log-likelihood function:

$$\log \mathcal{L}(p_0, p_1, ..., p_M|x) = \left(\sum_{i=1}^{N} \mathbb{1}[x_i = 0]\right) \log p_0 + \left(\sum_{i=1}^{N} \mathbb{1}[x_i = 1]\right) \log p_1 + ... + \left(\sum_{i=1}^{N} \mathbb{1}[x_i = M]\right) \log p_M \tag{4}$$

As we see, the log-likelihood function can be interpreted as the weighted sum of the log of parameters where weights are equal to all events where the outcome ocurred.

2.2 MLE

We aim to maximize the value of the log-likelihood function subject to the natural constraint that claims parameters must add up to 1. Essentially, there are two ways to solve this problem: the analytical and the numerical approach. At first, we derive and discuss some results under the analytical approach but in practice we use the numerical procedure since variance computations require working with a Hessian matrix which is hard to handle under the analytical approach.

$$\begin{aligned} \max_{p_0, p_1, ..., p_M} \quad & \log \mathcal{L}(p_0, p_1, ..., p_M|x) \\ \text{s.t.} \quad & \sum_{j=0}^{M} p_j = 1 \\ & p_0, p_1, ..., p_M \in [0, 1] \end{aligned} \tag{5}$$

This problem resembles the non-linear programming under equality constraints. By using Lagrange Multipliers (LM) we can obtain first order conditions (FOC) for every parameter p_j, and by including the parameter restriction per se we can obtain the maximum likelihood estimator for p_j, denoted as $\hat{p}_j$:

$$\hat{p}_j = \frac{\sum_{i=1}^{N} \mathbb{1}[x_i = j]}{N} \tag{6}$$

As you can see, the MLE for p_j is the mean of the related outcome in the sample. It is worth noting that we are not interested on these probabilities per se but instead we want to get an estimate of the average value of the count variable that takes a finite number of outcomes. As with every discrete random variable we define the mean value of such variable as follow:

$$\hat{\mathbf{E}}(x) = \hat{p}_0 \times 0 + \hat{p}_1 \times 1 + ... + \hat{p}_M \times M = \sum_{j=0}^{M} \hat{p}_j \times j \tag{7}$$

Now, we want to discuss how much confident is the measure $\hat{\mathbf{E}}(x)$. In order to do so, we delve into variance estimation issues in the MLE approach. In particular, we aim to get a confidence interval for $\hat{\mathbf{E}}(x)$ rather than $\hat{p}_j$. However, keep in mind that $\hat{\mathbf{E}}(x)$ is a function of $\hat{p}_j$.

2.3 Variance Estimation and Confidence Intervals

The maximum likelihood estimator has desirable asymptotic properties given that we work with a large sample size. It is consistent (it converges in probability to the population parameter), asymptotically normally distributed and asymptotically efficient (it reaches the minimal variance attainable by all consistent estimators, this is known as the Cramer-Rao lower bound). We summarize these properties as follows:

$$\begin{aligned} &\operatorname*{plim}_{n\to\infty} \hat{\mathbf{p}} = \mathbf{p} \\ &\hat{\mathbf{p}} \xrightarrow{a} \mathcal{N}\left[\mathbf{p}, \mathbf{I}\left(\mathbf{p}\right)^{-1}\right] \\ &\mathbf{I}\left(\mathbf{p}\right) = -E\left[\frac{\partial^2 \log \mathcal{L}}{\partial \mathbf{p} \partial \mathbf{p}^T}\right] \end{aligned} \tag{8}$$

where n is the sample size, and $\mathbf{I}\left(\mathbf{p}\right)$ is the information matrix. This matrix is equal to the expected value of the Hessian matrix associated to the optimization problem and evaluated at the population parameter vector $\mathbf{p} = p_0, p_1, ..., p_M$. We must note it is difficult to obtain $\mathbf{I}\left(\mathbf{p}\right)$ not only because we need to obtain the analytical form of a Hessian matrix (which entails calculating second and cross derivatives) but also because we need to do so many times in order to obtain its expected value. In practice we retain the negative of the numerical Hessian matrix associated to the MLE and treat it as an estimate of $\mathbf{I}\left(\mathbf{p}\right)$. In turn, we treat the inverse of such matrix as an estimate of the covariance matrix for estimator vector $\hat{\mathbf{p}}$.

2.3.1 The Delta Method

As you may remember, we are not interested in obtaining a CI for an arbitrary parameter $\hat{p}_j$ but rather we aim to obtain a CI for $\hat{\mathbf{E}}(x)$. One way to do so is by using the Delta Method for the multivariate case: To keep using the same notation, suppose we have a vector of $M+1$ estimates, called $\hat{\mathbf{p}} = (\hat{p}_0, \hat{p}_1, ..., \hat{p}_M)$, and their related covariance matrix, called $\mathbf{I}\left(\mathbf{p}\right)^{-1}$. We aim to find the covariance matrix of a vector of estimators that are a set of functions of the raw vector estimators, say $\mathbf{f}(\hat{\mathbf{p}})$. If $\mathbf{f}(\cdot)$ is a set of C^1 functions and $\hat{\mathbf{p}}$ is normally distributed as in equation (8) then:

$$\mathbf{f}(\hat{\mathbf{p}}) \xrightarrow{a} \mathcal{N}\left[\mathbf{f}(\mathbf{p}), \mathbf{f'}(\mathbf{p})\mathbf{I}\left(\mathbf{p}\right)^{-1}\mathbf{f'}(\mathbf{p})^T\right] \tag{9}$$

where $\mathbf{f'}(\mathbf{p})$ is the Jacobian matrix of first derivatives of every new estimator w.r.t each raw estimator. Since $\hat{\mathbf{E}}(x)$ is a function of raw parameters $\hat{p}_j$, then we can use the Delta Method to provide a CI for $\hat{\mathbf{E}}(x)$. Once we obtain a standard deviation for $\hat{\mathbf{E}}(x)$, say σ, we use the following formula for the CI of a normal distribution:

$$CI(\hat{\mathbf{E}}(x)) = \hat{\mathbf{E}}(x) \pm z^*_{\alpha/2}\frac{\sigma}{\sqrt{n}} \tag{10}$$

where $z^*_{\alpha/2}$ is the critical value of the standard normal distribution for a Type I error level of α and n is the sample size. The alternative is to use the bootstrapping method which we discuss in the next section.

2.3.2 The Bootstrapping Method