Dear Tak,

We have revised our MCMC method approach and are happy to provide you with an update.

1. Gamma parameters:
   * Initial gamma parameters were fit to RTM rather than the generator.

We initially did so under the concern that the Python code continuously gave errors when trying to fit a gamma distribution on the generator.

**However, after excluding the diagonal components, the gamma parameters were fitted without warning.**Thank you so much for the guidance!

* + There was confusion with the parameters for Python’s gamma distribution since “scale” in python is “1/beta” for us.

**We have fixed the code and properly changes our “tuned\_alpha” to be “q\_ij \* beta”**

* + We have noticed some negative alpha values as well. We have kept the “floored at zero” method **but also tried another version that involves the generator matrix from the EM algorithm**. We thought that since the EM algorithm, in theory, makes the same distributional assumption as the MCMC algorithm, it would make sense to use the EM algorithm-derived generator matrix entries as the averages that we would use to tune the parameters. ***In this case, no negative alpha values were given***. (i.e., original generator matrix for the beta values and EM algorithm generator matrix for the alpha values)

* + **We have not tuned the parameters for the last row, since theoretically coming back from a “D” rating was deemed impossible and the “gamma.fit” function would be erroneous. Please let us know if this assumption is not valid.**

1. MCMC algorithm:

* We thought that even if we used tuned prior parameters, we would still need to update the parameters from the posterior (page 18 of the Bank of Japan paper). We feared that if we do not update the parameter, then it would not qualify for an MCMC sampling method since ***the posterior would be the same as the prior***.

* + 1. Draw X(1) ~ p(X | Q(0), x)
    2. **Draw Q(1) ~ p(Q | X(1), x)**
    3. **Draw X(2) ~ p(X | Q(1), x)**

Nevertheless, we have included **an input argument “keep\_hyperparams” in the function “MCMC\_algo”** to accommodate (i.e., **if  “True”: alpha,** **beta**= **alpha\_0, beta\_0, if “False”: alpha,** **beta**updates)

* For the exterior approach, “Q\_gen” and “Q” were kept separately. We do not know how that error occurred but checked prior to this update that the “Q\_gen” is not over-written during the process.

**(Further points to review)**

* We thought similar to the ***Poisson Arrival Process (as pointed out in page 6 of the Bank of Japan paper)***. Since “q\_k” denotes the rate at which the obligor’s rating leaves the current rating, we thought to sample the holding time from the exponential distribution with rate “q\_k” (analogous to the Poisson Arrival Process). Regardless, we have added **an input argument “draw\_from\_exp” to the function “RTM\_Chain”** to accommodate (i.e., **if  “True”: S\_k ~ f\_k, if “False”: S\_k = f\_k**)

* Even if we draw from the exponential distribution by setting “draw\_from\_exp” as “True***”, as you pointed out, in python coding “1/q\_k” should have been passed on as scale parameter not “q\_k”.***This was a careless mistake without reviewing the python documentation. We have fixed the method in this update accordingly. We apologies and thank you for your guidance!

* We thought that if either **S\_k ~ f\_k**or **S\_k = f\_k**was less than “**delta\_t**” then the obligor would have reached an observed rating and moved on to the next row. That was our understanding of the phrase “until the process reaches an observed rating grade” **(If S\_k > delta\_t, we considered it not having reached observed rating and still under change of rating status)**. **We also assumed that rating change can occur during “delta\_t” at most once**. Please let us know if any of our assumptions were incorrect!

* Again, our understanding was that “S\_k” was drawn from “f\_k” that takes “q\_k” and “delta\_t” as a function. **We believe we will have to investigate more whether our current coding method would not be suitable if “delta\_t” is too small.**

1. RTM Chain:

* Applying K-1 was to accommodate for another error at the time. Together, the code was inefficient and foolish. **We have completely changed our code using “np.where()” and now it does not run into the same issues**. Thank you for your guidance!

* With the revised code, it runs without error or exception. **We no longer see the first column (left) having large probabilities when visualized**. Thank you for your guidance!

1. Convergence:

* We have not been able to consider the burn-in period for the initial update. Thank you for your guidance!
* As suggested, we used **ESS/autocorrelation of the parameters using the “arviz” package in Python to decide on the convergence and the burn-in period**.

Thank you for your guidance!

\*\*It took approximately ***45 minutes*** to run ***100,000 iterations*** to check for convergence\*\*

* We thought that for the tuned prior approach with “**keep\_hyperparams = True**”, ***we would not need to consider the burn-in period since the generator matrix is always sampled from the same gamma distribution with the same parameters.*Please let us know if this assumption is wrong**.
* For the new results, **we only averaged 1000 samples after the chosen burn-in period**.
* ***The prediction from the external generator seems to be still far off and we are not completely sure about convergence reached***. We fear that our **weak priors may be too weak** for this simulation. **We hope to see more convergence if we increase the number of iterations significantly (though time-costly)**.

Thank you so much for all the suggestions and guidance!

Hope to speak with you again soon!

Sincerely,

UChicago FINM Project Lab (Team Mizuho)

Hi project team,

Here are my notes/comments for the latest jupyter notebook.

Responses to your notes; see red comments for the responses

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

>> 1) Gamma parameters:

**>> We have not tuned the parameters for the last row, since theoretically coming back from a “D” rating was deemed impossible and the “gamma.fit” function would be erroneous. Please let us know if this assumption is not valid.**

Agreed.

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

>> 4) Convergence:

>> We thought that for the tuned prior approach with “**keep\_hyperparams = True**”, ***we would not need to consider the burn-in period since the generator matrix is always sampled from the same gamma distribution with the same parameters.*Please let us know if this assumption is wrong**.

I would **disagree**.

I believe we need to update parameters of gamma distribution even when giving the prior distribution.

Much more speaking, I am still confused with the conceptual difference between prior approach vs posterior approach you did.

First, I found another following paper by Bladt and Sørensen, that describe the MCMC approach in more detail (attached in case).

* Bladt, M., & Sørensen, M. (2009). Efficient estimation of transition rates between credit ratings from observations at discrete time points. *Quantitative Finance*, *9*(2), 147-160.

Let’s follow the paper below (partially copied from page 8 of the paper).

In the approach that you name prior approach, the alphas and betas are CONSTANT, which had been fit to the generator.



And through the sampling algorithm, the alphas/betas are NOT updated, however, new Ns and Rs NEED to be added up to those parameters.

A picture containing text, font, screenshot

Description automatically generated

In your current code, “**RTM\_Chain**” function should return N\_tilde and R\_tilde only, and then are added up to the constant alpha\_0 and beta\_0 in either “**gen\_MCMC**” function and “**MCMC\_algo**” function. In other words, my understanding is that the parameter adjustments should be made for “difference” from the original alpha\_0 and beta\_0, whereas your current code seems to fail to keep alpha\_0 and beta\_0 through the updating process.

A picture containing text, font, screenshot, line

Description automatically generated

A black text on a white background

Description automatically generated with low confidence

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

We would discuss further in the meeting tomorrow, but it is nicer if you could roughly take a look at the paper attached before the meeting.

Best,

Tak

**--------------------------------------------------**

**Takuya Otani**

Vice President, Counterparty Portfolio Management Desk

Fixed Income Division

Mizuho Securities USA LLC

Mizuho Americas

1271 Avenue of the Americas, New York, NY 10020

T: +1 646-949-9817

[takuya.otani@mizuhogroup.com](mailto:takuya.otani@mizuhogroup.com)