Bayesian Mallows Model for Rank Data

장태영

서울대학교
통계학과, 베이즈통계 연구실

2021. 11. 12

CONTENTS

CONTENTS

Motivation

- Ranking and comparing items
  - Crucial for collecting information about preferences in many areas from marketing to politics
  - Netflix, Spotify, 배달의 민족, ...

Example data

```
> head(sushi_rankings)
     shrimp sea eel tuna squid sea urchin salmon roe egg fatty tuna tuna roll cucumber roll
[1,]      2       8   10     3          4          1   5          9         7             6
[2,]      1       8    6     4         10          9   3          5         7             2
[3,]      2       8    3     4          6          7  10          1         5             9
[4,]      4       7    5     6          1          2   8          3         9            10
[5,]      4      10    7     5          9          3   2          8         1             6
[6,]      4       6    2    10          7          5   1          9         8             3
```

What we want to do?

- Find consensus ranking of the items
- Extensions of model for pairwise comparisons, preference prediction and clustering.

CONTENTS

Elementary settings

- Setting : n items and N assessors. $\mathbf{R}_j \in \mathcal{P}_n$ denotes the ranking(the full set of ranks given to the n items) of assessor j for each $j = 1, \cdots, N$. ($\mathcal{P}_n$ is a permutation set)
- $d(\cdot\ ,\ \cdot) : \mathcal{P}_n \times \mathcal{P}_n \to [0, \infty)$ is a distance function between two rankings.
    - Kendall distance : number of pairs of distinct elements whose order in the two rankings are the opposite.
    - Footrule distance : ℓ_1 distance
    - Spearman's distance : ℓ_2 distance

Mallows model

- Mallows model is a class of non-uniform joint distributions for a ranking $\mathbf{r}$ on $\mathcal{P}_n$.

$$P(\mathbf{r}|\alpha, \boldsymbol{\rho}) = Z_n(\alpha, \boldsymbol{\rho})^{-1} \exp\{-\frac{\alpha}{n} d(\mathbf{r}, \boldsymbol{\rho})\} I(\mathbf{r} \in \mathcal{P}_n)$$

- $\boldsymbol{\rho} \in \mathcal{P}_n$ is the latent consensus ranking.
- $\alpha > 0$ is a scale (or precision) parameter.
  i.e. α represents the level of agreement between assesssors, so that as α gets larger, ranking $\mathbf{r}$ aggregates more to $\boldsymbol{\rho}$
- $Z_n(\alpha, \boldsymbol{\rho}) = \sum_{\mathbf{r} \in \mathcal{P}_n} e^{-\frac{\alpha}{n} d(\mathbf{r}, \boldsymbol{\rho})}$ is the partition function.
  - In physics, a 'partition function' describes the statistical properties of a system in thermodynamic equilibrium. (Source : Wikipedia)
  - Here, just consider this as a normalizing factor.

Likelihood function

- Assume that observed rankings $\mathbf{R}_1, \cdots, \mathbf{R}_N$ are conditionally independent given α and $\boldsymbol{\rho}$ and each of them is distributed according to the Mallows model with these parameters.
- Likelihood takes the form as

$$P(\mathbf{R}_1, \cdots, \mathbf{R}_N | \alpha, \boldsymbol{\rho}) = Z_n(\alpha, \boldsymbol{\rho})^{-N} \exp\{-\frac{\alpha}{n} \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho})\} \prod_{j=1}^{N} I(\mathbf{R}_j \in \mathcal{P}_n)$$

- For large n, finding the MLE of $\boldsymbol{\rho}$ given fixed α is not feasible because the space of permutations $\mathcal{P}_n$ has $n!$ elements.

Right-invariant distance and partition function

- For any right-invariant distance, it holds $d(\mathbf{r}_1, \mathbf{r}_2) = d(\mathbf{r}_1\mathbf{r}_2^{-1}, \mathbf{1}_n)$ where $\mathbf{1}_n = \{1, 2, \cdots, n\}$ and $\mathbf{r}_1 \mapsto \mathbf{r}_1\mathbf{r}_2^{-1}$ is relabelling map. Note that a right-invariant distance is unaffected by a relabelling of the items.
- Partition function $Z_n(\alpha, \boldsymbol{\rho})$ does not depend on $\boldsymbol{\rho}$.

$$
\begin{aligned}
\because Z_n(\alpha, \boldsymbol{\rho}) &= \sum_{\mathbf{r}\in\mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\mathbf{r}, \boldsymbol{\rho})\} = \sum_{\mathbf{r}\in\mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\mathbf{r}\boldsymbol{\rho}^{-1}, \mathbf{1}_n)\} \\
&= \sum_{\mathbf{r}'\in\mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\mathbf{r}', \mathbf{1}_n)\}
\end{aligned}
$$

$$
Z_n(\alpha, \boldsymbol{\rho}) = Z_n(\alpha) = \sum_{\mathbf{r}\in\mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\mathbf{r}, \mathbf{1}_n)\}
$$

Right-invariant distance and partition function

- For some choice of right-invariant distance like Kendall distance, the partition function can be analytically computed.
- But there are important and natural right-invariant distances for which the computation of the partition function is not feasible, such as the footrule distance and the Spearman's distance.

Prior distributions

- Assume a priori that α and $\boldsymbol{\rho}$ are independent
- In this paper, the uniform prior $\pi(\boldsymbol{\rho}) = \frac{1}{n!} I(\boldsymbol{\rho} \in \mathcal{P}_n)$ is employed.
- Also, for the scale parameter, this paper used a truncated exponential prior with density $\pi(\alpha|\lambda) = \lambda e^{-\lambda\alpha} I(\alpha \in [0, \alpha_{max}])/(1 - e^{-\lambda\alpha_{max}})$ where the cut-off point $\alpha_{max} < \infty$ is large compared to the values supported by the data. In practice, in the computations involving the sampling of values for α, truncation was never applied. We assign λ a fixed value close to zero, implying a prior density for α which is quite flat.
    - In short, prior $\alpha \sim Exp(\frac{1}{\lambda})$ with small λ is used practically for α.

Posterior distributions

- The posterior distribution for $\boldsymbol{\rho}$ and α is given by

$$P(\boldsymbol{\rho}, \alpha | \mathbf{R}_1, \cdots, \mathbf{R}_N) \propto \frac{\pi(\boldsymbol{\rho})\pi(\alpha)}{Z_n(\alpha)^N} \exp\Big\{ -\frac{\alpha}{n} \sum_{j=1}^{N} d(\mathbf{R}_j, \boldsymbol{\rho}) \Big\} \tag{1}$$

- The purpose of MCMC algorithm following is to obtain samples from this posterior.

Metropolis-Hastings algorithm

- A general form of the Metropolis Hastings algorithm is as follows : Target probability distribution is $p_0(x)$ for r.v. X. Given a current value $x^{(s)}$ of X,
  1. Generate x^* from a proposal distribution $J_s(x^*|x^{(s)})$
  2. Compute the acceptance ratio
  $$r = \frac{p_0(x^*)}{p_0(x^{(s)})} \Big/ \frac{J_s(x^*|x^{(s)})}{J_s(x^{(s)}|x^*)} = \frac{p_0(x^*)}{p_0(x^{(s)})} \frac{J_s(x^{(s)}|x^*)}{J_s(x^*|x^{(s)})}$$
  3. set $x^{(s+1)}$ to x^* with probability $\min(1, r)$
  i.e. Sample $u \sim unif(0, 1)$ and then if $u < r$ set $x^{(s+1)} = x^*$, else set $x^{(s+1)} = x^{(s)}$
- The primary restriction placed on $J_s(x^*|x^{(s)})$ is that it does not depend on values in the sequence previous to $x^{(s)}$ so that the algorithm generates a Markov chain.
- By Ergodic Thm, the empirical distribution of samples generated from such a Markov chain will converge to the stationary distribution(of the Markov chain), which agrees with the target distribution.
- Source : Hoff 2009. (textbook)

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Updating ρ

- Leap-and-Shift Proposal(L&S)
- Leap step
  1. Fix an integer $L \in \{1, 2, \cdots, \lfloor \frac{n-1}{2} \rfloor\}$
     (which is a tuning parameter for MCMC algorithm)
  2. Draw a random number $u \sim \mathit{Unif}\{1, 2, \cdots, n\}$
  3. Define $\mathcal{S} \subset \{1, 2, \cdots, n\}$ by
     $\mathcal{S} = \big[\max(1, \rho_u - L), \min(n, \rho_u + L)\big] \setminus \{\rho_u\}$
  4. Draw a random number $r \sim \mathit{Unif}(\mathcal{S})$
  5. Let $\boldsymbol{\rho}^* \in \{1, 2, \cdots, n\}^n$ have elements
     $\begin{cases} \rho_i^* = \rho_i & i \in \{1, 2, \cdots, n\} \setminus \{u\} \\ \rho_u^* = r \end{cases}$

Updating ρ

- Shift step
  1. Let $\Delta = \rho_u^* - \rho_u$. Note that $\Delta \neq 0$
  2. Define the proposed $\boldsymbol{\rho}' \in \mathcal{P}_n$ by below :
     1. If $\Delta > 0$ then
$$\begin{cases} \rho_u' = \rho_u^* & \\ \rho_i' = \rho_i - 1 & \textit{if } \rho_u < \rho_i \leq \rho_u^* \\ \rho_i' = \rho_i & \textit{otherwise} \end{cases}$$
     2. If $\Delta < 0$ then
$$\begin{cases} \rho_u' = \rho_u^* & \\ \rho_i' = \rho_i + 1 & \textit{if } \rho_u > \rho_i \geq \rho_u^* \\ \rho_i' = \rho_i & \textit{otherwise} \end{cases}$$

Updating ρ

- Example of Leap and Shift proposal

```
> #n=8, L=3
> print(r)
[1] 4 5 8 6 3 7 1 2
> print(u)
[1] 4
> setdiff(max(r[u]-L,1):min(r[u]+L,n),r[u])
[1] 3 4 5 7 8
> print(r.star)
[1] 4 5 8 3 3 7 1 2
> print(r.prime)
[1] 5 6 8 3 4 7 1 2
```

Updating ρ

- The probability mass function associated to the transition

$$P_L(\boldsymbol{\rho}'|\boldsymbol{\rho}) = \sum_{u=1}^{n} P_L(\boldsymbol{\rho}'|U=u,\boldsymbol{\rho})P(U=u)$$

$$= \frac{1}{n}\sum_{u=1}^{n}\left\{ I_{\{\rho_{-u}\}}(\boldsymbol{\rho}^*_{-u}) I_{\{0<|\boldsymbol{\rho}_u-\boldsymbol{\rho}^*_u|\le L\}}(\boldsymbol{\rho}^*_{\boldsymbol{u}}) \left[\frac{I_{\{L+1,\cdots,n-L\}}(\boldsymbol{\rho}_u)}{2L} + \sum_{z=1}^{L}\frac{I_{\{z\}}(\boldsymbol{\rho}_u)+I_{\{n-z+1\}}(\boldsymbol{\rho}_u)}{L+z-1}\right]\right\}$$

$$+ \frac{1}{n}\sum_{u=1}^{n}\left\{ I_{\{\rho_{-u}\}}(\boldsymbol{\rho}^*_{-u}) I_{\{|\boldsymbol{\rho}_u-\boldsymbol{\rho}^*_u|=1\}}(\boldsymbol{\rho}^*_{\boldsymbol{u}}) \left[\frac{I_{\{L+1,\cdots,n-L\}}(\boldsymbol{\rho}^*_u)}{2L} + \sum_{z=1}^{L}\frac{I_{\{z\}}(\boldsymbol{\rho}^*_u)+I_{\{n-z+1\}}(\boldsymbol{\rho}^*_u)}{L+z-1}\right]\right\}$$

Updating ρ

- Simple representation for the transition probability
  - As we calculate $P(\boldsymbol{\rho}'|\boldsymbol{\rho})$, we should consider two random draws
    - Draw $u \sim Unif\{1, 2, \cdots, n\}$
    - For S dependent on $\boldsymbol{\rho}_u$, draw $r \sim Unif(S)$
    - The other works including shift step involve no randomness.
  - Simply put, $P(\boldsymbol{\rho}'|\boldsymbol{\rho}) = \frac{1}{n} \cdot \frac{1}{|S|}$ for many cases.
  - However, if $|\boldsymbol{\rho}'_u - \boldsymbol{\rho}_u| = 1$ then we should consider something more.
  - When $|\boldsymbol{\rho}'_u - \boldsymbol{\rho}_u| > 1$ then u is the only possible index that proposes $\boldsymbol{\rho}'$ from $\boldsymbol{\rho}$. On the other hand, when $|\boldsymbol{\rho}'_u - \boldsymbol{\rho}_u| = 1$, there must be only one index u' other than u s.t. $|\boldsymbol{\rho}'_{u'} - \boldsymbol{\rho}_{u'}| = 1$ so that u' can also proposes $\boldsymbol{\rho}'$ from $\boldsymbol{\rho}$.
  - In this special case, $P(\boldsymbol{\rho}'|\boldsymbol{\rho}) = \frac{1}{n} \cdot \frac{1}{|S|} + \frac{1}{n} \cdot \frac{1}{|S'|}$ where S is produced from drawing u and S' is produced from drawing u'

Updating ρ

- example of leap and shift proposal when $|\rho'_u - \rho_u| = 1$

```
> #n=8, L=3
> print(r)
[1] 7 6 2 1 4 8 5 3
> print(u)
[1] 7
> setdiff(max(r[u]-L,1):min(r[u]+L,n),r[u])
[1] 2 3 4 6 7 8
> print(r.star)
[1] 7 6 2 1 4 8 6 3
> print(r.prime)
[1] 7 5 2 1 4 8 6 3
```

Updating ρ

- Using this logic, we can rewrite the equality about $P_L(\rho'|\rho)$ as the following

$$P_L(\rho'|\rho) = \sum_{u=1}^{n} P_L(\rho'|U = u, \rho) P(U = u)$$
$$= \frac{1}{n} \sum_{u=1}^{n} I(\rho', \rho, u) \frac{1}{|S^{(u)}|}$$

where $I(\rho', \rho, u)$ is an indicator for possibility of proposal from ρ to ρ' given u is drawed and $S^{(u)}$ is the set S given u is drawed
If ρ' is proposed from ρ then typically $I(\rho', \rho, u) = 1$ for only one u but if $|\rho'_u - \rho_u| = 1$ then $I(\rho', \rho, u') = 1$ also holds for another u' different from u

Updating $\boldsymbol{\rho}$

- The acceptance probability when updating $\boldsymbol{\rho}$ is $\min\{1, r\}$ where r is given as

$$r = \frac{P(\boldsymbol{\rho}', \alpha|\mathbf{R})}{P(\boldsymbol{\rho}, \alpha|\mathbf{R})} \cdot \frac{P_L(\boldsymbol{\rho}|\boldsymbol{\rho}')}{P_L(\boldsymbol{\rho}'|\boldsymbol{\rho})}$$
$$= \frac{P_L(\boldsymbol{\rho}|\boldsymbol{\rho}')}{P_L(\boldsymbol{\rho}'|\boldsymbol{\rho})} \cdot \frac{\pi(\boldsymbol{\rho}')}{\pi(\boldsymbol{\rho})} \exp\Big\{ -\frac{\alpha}{n} \sum_{j=1}^{N} \big[d(\mathbf{R}_j, \boldsymbol{\rho}') - d(\mathbf{R}_j, \boldsymbol{\rho})\big]\Big\}$$

- Leap and shift proposal is not a symmetric proposal distribution.
- The term $\sum_{j=1}^{N} \big[d(\mathbf{R}_j, \boldsymbol{\rho}') - d(\mathbf{R}_j, \boldsymbol{\rho})\big]$ above can be computed efficiently since most elements of $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ are equal and we can put aside indices i s.t. $\rho_i = \rho_i'$

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
    1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
    2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

Metropolis-Hastings Algorithm for Complete Rankings

- To obtain samples from the posterior distribution (1), we alternate between two steps.
  1. Given α and $\boldsymbol{\rho}$, update $\boldsymbol{\rho}$ by proposing $\boldsymbol{\rho}'$
  2. Then, given α and $\boldsymbol{\rho}'$, update α by proposing α'

References I