MIS Quarterly

RESEARCH ARTICLE

THE RIGHT MUSIC AT THE RIGHT TIME: ADAPTIVE PERSONALIZED PLAYLISTS BASED ON SEQUENCE MODELING

Elad Liebman
SparkCognition, Inc., 4030 West Braker Lane #500, Austin, TX 78759 U.S.A. {eladlieb@gmail.com}

Maytal Saar-Tsechansky
McCombs School of Business, The University of Texas at Austin, 2110 Speedway, Stop B6500,
Austin, TX 78712-1277 U.S.A. {maytal@mail.utexas.edu}

Peter Stone
Department of Computer Science, The University of Texas at Austin, 2317 Speedway, Stop D9500,
Austin, TX 78712-1277 U.S.A. {pstone@cs.utexas.edu}

Appendix A

Extending DJ-MC to Incorporate Song Costs or Royalty Fees

In this appendix, we explore extensions of DJ-MC to accommodate song costs, as well as to evaluate the performance of several variants of DJ-MC.

An issue rarely explored in music recommendation is how to incorporate song costs in the content recommendation process. Not much is known about the particular royalty cost of playing each song in existing streaming services. Studies indicate that streaming services often negotiate with royalty agreements agencies,[1] and the royalties are typically determined in bulk, involving a large number of artists and songs, and are not assigned to each song individually. For example, Spotify, a leading streaming service, has recently revealed that it does not assign a different royalty fee to each song, but rather the payment to owners (labels, publishers, distributors, etc.) is determined *ex post* by the proportion of streams made relative to Spotify's total revenue from the corresponding period.[2] Nevertheless, given that the music industry is experiencing a transformation, it may also be that future models incorporate differential, individual song fees. Hence, in this section we propose how to incorporate individual song costs in the DJ-MC framework.

In principle, given a (mathematical) mapping between listener enjoyment and monetary value that the playlist service can expect, $Monetary(R)$, DJ-MC can incorporate song fees to optimize monetary returns directly and seamlessly. Once such mappings are established, song rewards R_s and song costs C_s inhabit the same (monetary) space, and DJ-MC can then operate as before, using a cost-adjusted reward model $R^* = Monetary(R) - C$. However, in the absence of an established mapping, one can consider alternative approaches with goals other than maximizing profit for streaming services to aim for.

[1] http://journals.law.stanford.edu/sites/default/files/stanford-technology-law-review/online/licensinginshadow.pdf

[2] https://www.spotifyartists.com/spotify-explained/#how-we-pay-royalties-overview

The first approach we consider selects the next song such that it aims to maximize the *ratio* between expected listener reward and song cost. Thus, in Algorithm 4 we now aim to optimize the ratio between the reward and the cost:

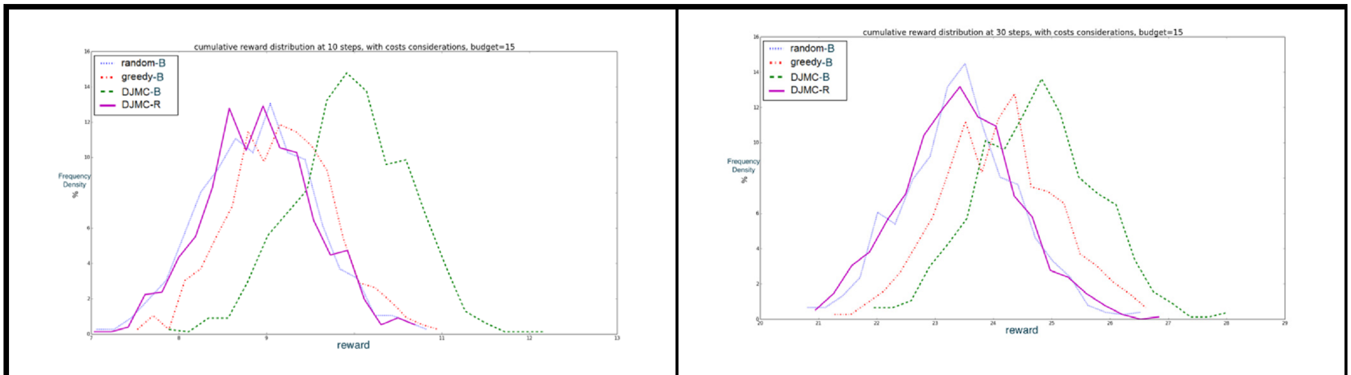
$$R_s\left(song_i\right)/C_s\left(song_i\right)+\Sigma_{i=2}^{q}\left(R_t\left(\left(song_1,\ldots,song_{i-1}\right)song_i\right)+R_s\left(song_i\right)\right)\Big/C_s\left(song_i\right)$$

This approach, henceforth referred to as DJ-MC-R (DJ-MC-RATIO), aims to promote the selection of songs that yield greater marginal enjoyment per unit cost. While DJ-MC-R may seem intuitive for handling the tradeoff between listener reward and cost, note that, because listener enjoyment is no longer the only criterion, to yield a high ratio, DJ-MC-R may also select sequences composed of songs (and transitions) that incur very low costs, even if the listener may not particularly enjoy them.

A second alternative we propose avoids a direct tradeoff between enjoyment and costs altogether. Specifically, it aims to produce sequences that yield the highest possible rewards *within a budget*, where a budget corresponds to the maximum cumulative fees one is willing to incur for a single listener over a given period of time. Henceforth, we refer to this variant as DJ-MC-B (DJ-MC-BUDGET), and we adapt DJ-MC to this framework by imposing a *budget constraint* on the exploration and planning process described in Algorithm 4. Specifically, in Algorithm 3, DJ-MC-B selects each song in the trajectory in the same way, but subject to the condition that $Cost(trajectory \cup \{song\}) \leq budget$. We similarly adapt the RANDOM and GREEDY baselines to create playlists that do not exceed the allocated budget. Specifically, the RANDOM baseline is adapted to accept only RANDOM sequences complying with the budget constraint. Similarly, GREEDY selects songs as before; however, if the next best song leads to a playlist that exceeds the budget, the song is skipped and the next highest-reward song is attempted. The budget-aware variants of GREEDY and RANDOM are henceforth referred to as GREEDY-B and RANDOM-B, respectively.

In the experiments reported below, song fees per stream for a given song are drawn uniformly from [0,1]. In addition, we identified that a budget of 15 imposes a meaningful constraint on the songs that can be played over the course of a session, such that methods that do not account for the cost of songs exhaust the budget in a significant proportion of the sessions. We therefore used a budget of 15 to explore the listener rewards produced by each approach. Finally, we examine separately the rewards and the remaining budget for each approach to draw conclusions on the tradeoffs offered by each.

Figures A1(a) and A1(b) show the listener reward distribution produced by each approach after 10 and then 30 songs, given a budget of 15. Together, Figures A1(a) and A1(b) show that both DJ-MC-B and DJ-MC-R yield higher remaining budgets in expectation as compared to the RANDOM and GREEDY benchmarks. This can be attributed to the fact that both DJ-MC-B and DJ-MC-R plan ahead, given they consider sequences, and are therefore less likely to exhaust their budget while generating the playlist. Because DJ-MC-R economizes on costs directly, it yields the highest remaining budget (uses a smaller proportion of its budget) as compared to DJ-MC-B. As expected, because DJ-MC-R selects songs for which the ratio between expected reward and cost is high, this also leads to the selection of songs that are inexpensive even if the listener may not particularly enjoy them. Consequently, the cumulative listener reward produced by DJ-MC-R is lower as compared to DJ-MC-B and GREEDY-B, whose primary criterion is to select songs that are likely to maximize listener reward.



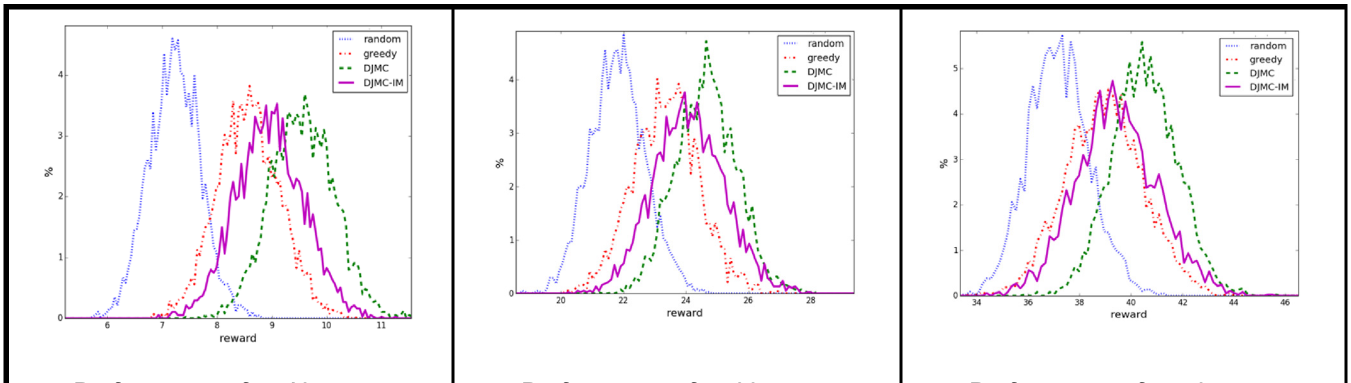
(a) Listener Reward Distribution After 10 Steps

(b) Listener Reward Distribution After 30 Steps

Figure A1. for the Simple Model (Uncorrelated LVs)

DJ-MC *with Only Immediate Reward*

Recall, because DJ-MC simultaneously *learns* and *acts*, its planning aims to manage a tradeoff between two goals: explore the listener’s preferences, and exploit its current knowledge of the listener’s preferences, so as to select a song that will be likely followed by an enjoyable sequence. To explore, DJ-MC simulates RANDOM sequences of songs. To assess the value of each simulated sequence, DJ-MC uses the reward model to (1) compute the *immediate* reward from playing the first song in the sequence, and (2) to assess how the next song may affect the enjoyment from the trajectory of future songs. Figure A2 shows results for a variant of DJ-MC, denoted DJ-MC-IMMEDIATE (DJ-MC-IM), in which DJ-MC uses its listener reward model of song and transition preferences to deterministically select the single song that yields the highest *immediate* reward. Hence, DJ-MC-im does not explore the listener’s preferences to improve its learning of these preferences, and it does not consider how the choice of the next song may affect the listener’s enjoyment from future songs.



Performance after 10 steps

Performance after 30 steps

Performance after 50 steps

Figure A2. Evaluation of DJ-MC-IM

As shown, consistent with our prior results for DJ-MC, DJ-MC- im’s learning and accounting for transition preferences when selecting the next song remains advantageous over the GREEDY approach, which selects the next song based on the *song* reward exclusively. The comparison between DJ-MC and DJ-MC-IM also sheds light on the benefits of the standard DJ-MC’s exploration of the listener’s preferences via the RANDOM rollouts, as well as DJ-MC’s considerations of not only the immediate song and transition reward (namely, the transition from songs played thus far onto the next song), but how the choice of the next song may affect the enjoyment from future songs in the playlist. Specifically, via exploration, DJ-MC aims to select songs that improve the listener reward model and the selection of future songs. By evaluating the rewards from future trajectories of songs, the standard DJ-MC is also enabling the possibility of not selecting a song with the best immediate song and transition reward, in order to select songs that will yield a more enjoyable sequence. As shown in Figure A2, DJ-MC yields better rewards already after 10 steps relative to the myopic variant, DJ-MC-IM.

Notes on Incorporating User Feedback in the State Space

One may consider representing user feedback in the state explicitly so as to enable DJ-MC to learn different models of behavior in the context of different listener feedback. This strategy can improve DJ-MC’s performance by, for example, being more conservative and avoiding explorations of the listener’s preferences when the listener is indicating dissatisfaction. However, feedback is already being reflected implicitly in DJ-MC‘s recommendations. For example, as reflected in Algorithm 3, the update weight extracted from the reward at step k, is given by log($v_k/\bar{v}_k$), where $\bar{v}$ is the running average of reward, reflecting recently observed feedback. Because after a sequence of negative rewards, the average, $\bar{v}$, decreases, a positive reward translates into a significant increase in the reward computed for songs the listener enjoys—more so than it would have been in a less negative context. Consequently, after a “bad run,” the model is quicker to adapt to any evidence of more favorable song preferences and, similarly, after a “good run,” it is slower to do so, requiring more favorable feedback in order to make positive adjustments. Therefore, this property of DJ-MC results in a higher likelihood in such a context to recommend songs the listener is likely to enjoy and, similarly, rendering DJ-MC less likely to explore a listener's taste by recommending less enjoyable songs.

Appendix B

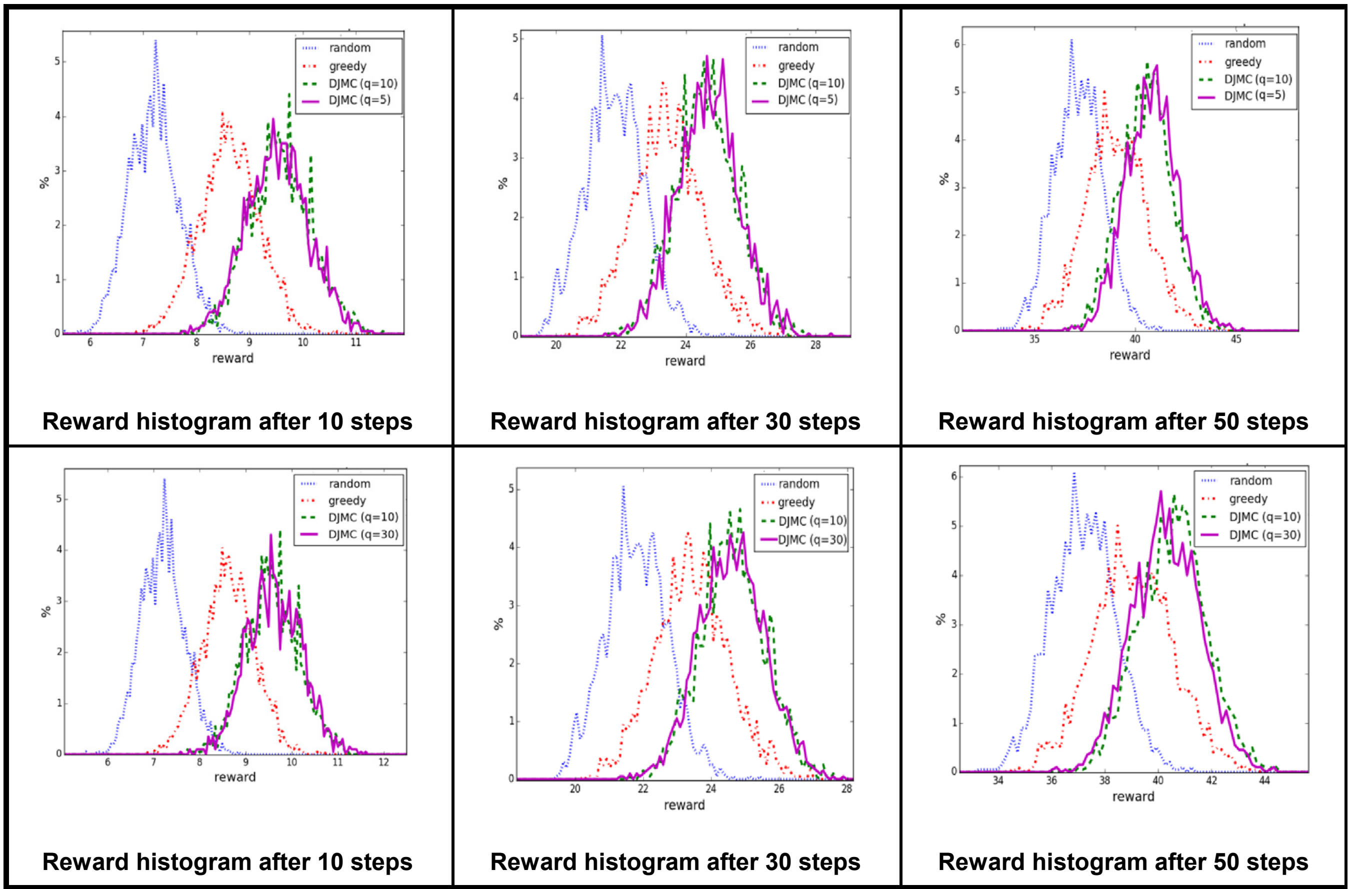
Additional Analyses

In this appendix, we report additional studies we have conducted to assess its performance under different parameter settings.

Planning Horizons

Our DJ-MC implementation uses a planning horizon, *q*, of 10 songs, while in practice, the true playlist horizon at any time can be arbitrarily longer. We therefore compared DJ-MC's performance with a planning horizon of 10 songs, to its performance with planning horizons of 5 or 30 songs.

As shown in Figure B1, we find that planning horizons of 5, 10 (as before), and 30 songs yield comparable results. Note that all the variants evaluated here perform stochastic exploration and an assessment of the likely effect of the next song on enjoyment from future songs in the playlist. Our earlier results show that not performing explorations, and myopically selecting the next song that yields the best immediate song and transition reward, yields worse rewards.



Reward histogram after 10 steps

Reward histogram after 30 steps

Reward histogram after 50 steps

Reward histogram after 10 steps

Reward histogram after 30 steps

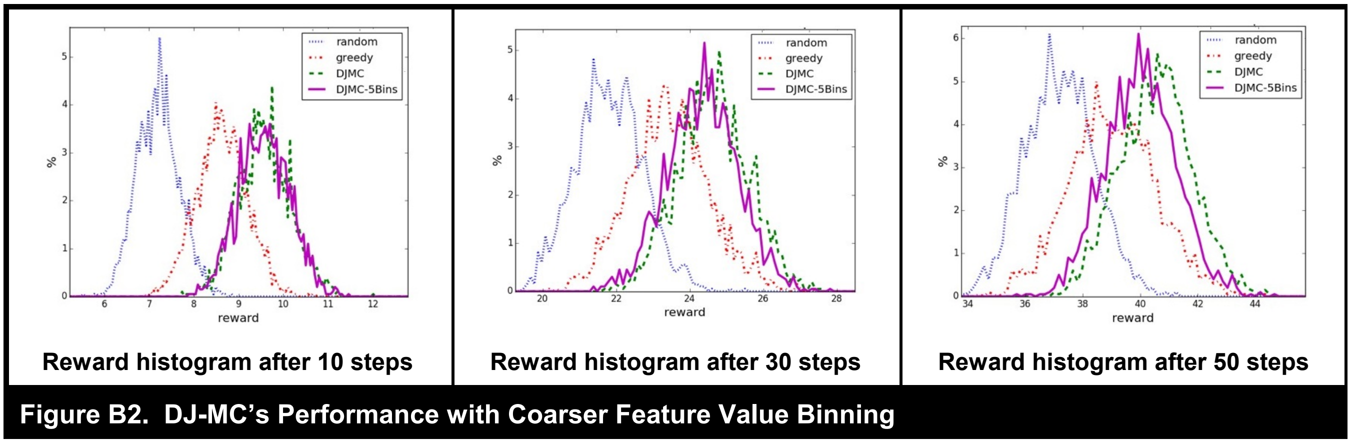
Reward histogram after 50 steps

Figure B1. DJ-MC's Performance Using Different Planning Horizons During Planning (First row: Comparison of DJ-MC with planning horizons (*q*) of 10 and 5 songs. Second row: comparison of DJ-MC with planning horizons (*q*) of 10 and 30 songs.)

Using Coarser Binning in the Song Representation

When considering the complexity of the listener reward model, we discuss the tradeoff between the model's bias and variance given the amount of *experiential* data, namely the amount of actual experiences with the listener. Fitting a reward model in our setting is constrained by the number of actual experiences with the listener during a listening session, namely between 10 and 50 songs. In principle, given unlimited training experiences, having a more complex (higher variance) reward model allows us to fit the listener's preferences better; however, if the number of training experiences is insufficient, this flexibility becomes a liability, and the reward model can over-fit the limited training experiences. Our study of a feature-dependent reward model, with the ability to model how a very large number of interactions between all possible pairs of song features maps onto preferences, shows that, given the number of experiences with the listener in our setting, such a model over-fits these training experiences. Thus, we find in our experiments that using such a listener reward model undermines DJ-MC's performance, even when the model reflects the true patterns underlying listeners' preferences.

The song representation adds another dimension by which we can control the reward model's complexity. The first element is the binning itself, so that we represent songs by the corresponding percentile bin value for each feature, rather than the corresponding feature's precise value. In addition, the choice of number of bins can vary as well. As shown in Figure B2, a comparison between DJ-MC with 5 bins and the standard DJ-MC with 10 bins, shows that the latter does not over-fit relative to the former. Thus, we find that having fewer bins does not introduce quite as dramatic a change to the variance of the model.



Reward histogram after 10 steps

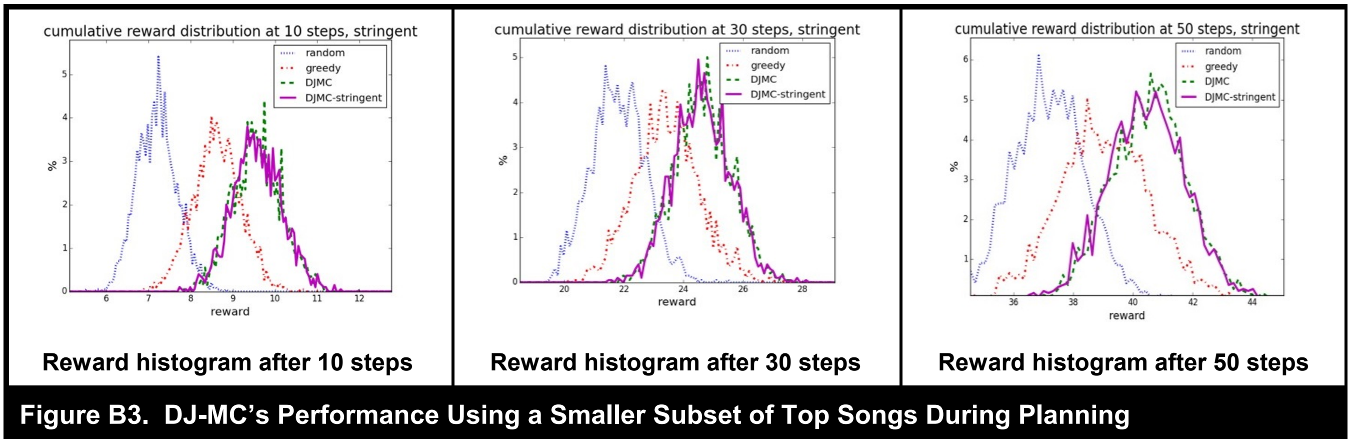
Reward histogram after 30 steps

Reward histogram after 50 steps

Figure B2. DJ-MC's Performance with Coarser Feature Value Binning

Using Smaller Subset, B, of Top Songs During Planning

During planning, DJ-MC produces rollouts, namely RANDOM sequences from the top B = 50% of the listener's most favorable songs (as estimated by DJ-MC) so as to assess the likely effect of the first song in the sequence on enjoyment from subsequent songs. We explored the sensitivity of our results to using a smaller subset of the top 25% most enjoyable songs in M. This reduction in the song space primarily increases the likelihood of selecting an enjoyable song; however, it might not improve the likelihood of identifying an enjoyable sequence and corresponding transitions. As show in Figure B3, our results suggest that reducing the set to 25% yields comparable performance for DJ-MC.



Reward histogram after 10 steps

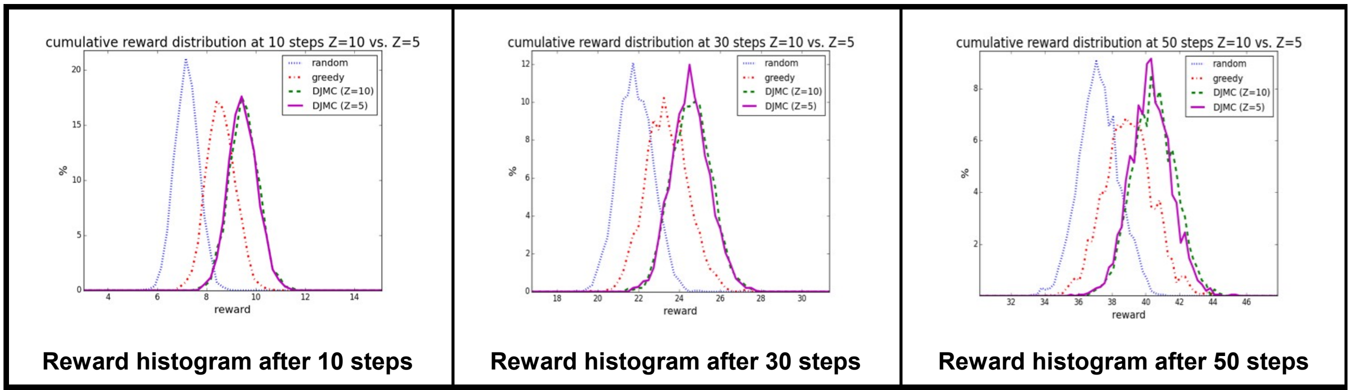
Reward histogram after 30 steps

Reward histogram after 50 steps

Figure B3. DJ-MC's Performance Using a Smaller Subset of Top Songs During Planning

Polling Listeners on Fewer Songs and Transitions During Initialization in Simulation Studies

In the simulation studies, a listener's reward model is first initialized with uniform weights so as to reflect that all songs and transitions are equally desirable. The initialization proceeds to poll the listener for songs she prefers and then asking the listener select a short sequence from which transition preferences are initialized. In the experiments we report in the paper, the (simulated) listener is polled for 10 songs and transitions. Figure B4 shows that when the number of songs and transitions the listener is polled for is five, the results are only slightly worse initially, but not meaningfully different from the results we report in the body of the paper. In the conclusions section, we discuss future work to explore the use of prior knowledge during initialization, so as to allow production of better playlists earlier.



Reward histogram after 10 steps

Reward histogram after 30 steps

Reward histogram after 50 steps

Figure B4. DJ-MC's Performance When Listeners are Polled for 5 and 10 Songs and Transitions During Initialization