# Playlist Prediction via Metric Embedding

October 24, 2017

## 1 Problem being addressed

Prediction of a music playlist involves generating a sequence of songs that has *meaningful transition* from one song to the next. The desirable aspect of such generative modelling is that the algorithm learns the representation without having to rely on a priori set of features. (auto learn the music embeddings.)

## 2 Key ideas and contributions

- Introduces **Logistic Markov Embedding (LME)** to learn to represent each song in a single-point / dual-point model space. Uses Euclidean distance as a metric of similarity. Uses metric embeddings to generalize the sequences, where playlists are paths in this latent space. The transition probabilities follow a logistic model. Optimization is by maximizing the log-likelihood.

- The *single-point model* considers a $d$-dimensional Euclidean space. But this is a symmetric space, and asymmetric directionality is important in transition (song $s_i$ to $s_j$ is meaningful, but not necessarily $s_j$ to $s_i$ ). Also faces coherences issues due to transitions to other genres.

- The *dual-point model* uses 2 vectors (entry *U(s)*, exit *V(s)*) to represent song $s$. Asymmetric song divergence $\|V(s) - U(s')\|_2$ for $s \to s'$ is considered. Without careful regularization, this model overfits with increasing dimensionality of the embedded space. Needs an extra local path regularizer $\Delta(s_a, s_b)$.

- Final model:
$$P(p_i|p_{i-1}) = \frac{e^{-\alpha\Delta(p_i,p_{i-1})^2 + \beta b_i - \gamma\Delta(p_i,p_0)^2}}{Z(p_{i-1}, p_0, \alpha, \beta, \gamma)}$$
$\alpha$ controls whether the model takes large/short steps. $\beta$ controls influence of popular songs. $\gamma$ controls how much the playlist stays close the seed location.

- Uses SGD for optimization of the non-convex objective which is $O(|S|^2)$ using every song pair transition matrix. Not all pairs are meaningful transitions. So, use a nearest neighbor *landmark heuristic* that works on a subset of songs $C_a$ $\forall a$

- Use of an extra local path regularizer turns the problem into a convex combination of local, non-convex problems that can hopefully be solved tractably. Global log-likelihood can be represented in terms of sum of local log likelihoods of transitions of form $s_a \to s_b$

- LME performs better than unigram and bigram (aka n-gram) sequence modelling. n-gram models from NLP comparatively fare poorly because the playlists have more uniform transition probabilities between 2 songs, as well as less data to train the NLP models (in comparison to training with text).

- **Training data** - Radio Playlists from Yes.com
  **Evaluation metric** - Recall, Avg log-likelihood.
  **Baselines** Uniform, Unigram, Bigram, models.

# 3 Strong points

- Successfully implements LME model that has enough expressive power to generate coherent sequences. Intuitive dual-point model with asymmetric divergence. Also uses a local path regularizer that turns the non-convex problem into convex sum of solvable local non-convex problems

- N-hop transitions are evaluated to discuss coherence. Small hops have higher avg. log likelihood. As N increases, this metric value reduces indicating meaningful transition pair dynamics.

- The LME model can easily be extended to incorporate user preferences, semantic tags, observable features as well as popularity. The model also allows control over the extent of popularity to consider over the similarity in the metric space.

- Intuitive generation of playlists using seed location (single song/ artist, user choice)

# 4 Weak Points

- Dual point model does not fare well over single point model. Does this mean the directionality is not important? Is this the nature of the playlist dataset?

- Directionality : The paper results clearly indicate no significant improvement in incorporating the directionality in the given radio playlist. Additionally, the strict asymmetric directionality assumption can be argued against.

- Long range dependencies : The model considers an iid Markov chain of order 1. The authors do touch upon long-range dependencies, but did not provide any results/methods (like time decay) to incorporate this.

- Subjectivity : The interpretation of coherence can be different for an amateur to a music aficionado to a connoisseur. The sensitivity control per user would add more parameters.

# 5 Extensions

- Inclusion of temporal dynamics. Covered by Moore in *Taste Over Time: the Temporal Dynamics of User Preferences*

- Handling Directionality better.

- Online, feedback receptive playlist generation. Incorporate thumbs-up/down to transition smoothly to a better, but not similar song as current.

# 6 Questions

1. Why does LME perform well on previously unseen transactions?

2. To what *extent* is the directionality and asymmetry important?

3. How logical is to embed the semantic tags and the songs in the same embedding space? What problems occur?

4. How do you impute ”some probability estimate” for a new song while testing? A prior value?

# 7 Misc

Easy, quick read. Creative. Second read unravels the underlying complexity and ingenuity. Beautiful!
ACM ref:
Shuo Chen, Josh L. Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist prediction via metric embedding. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD ’12). ACM, New York, NY, USA, 714-722. DOI: https://doi.org/10.1145/2339530.2339643

Here is an extremely good dissection of this paper by Charles H Martin: LME-Calculated Contents