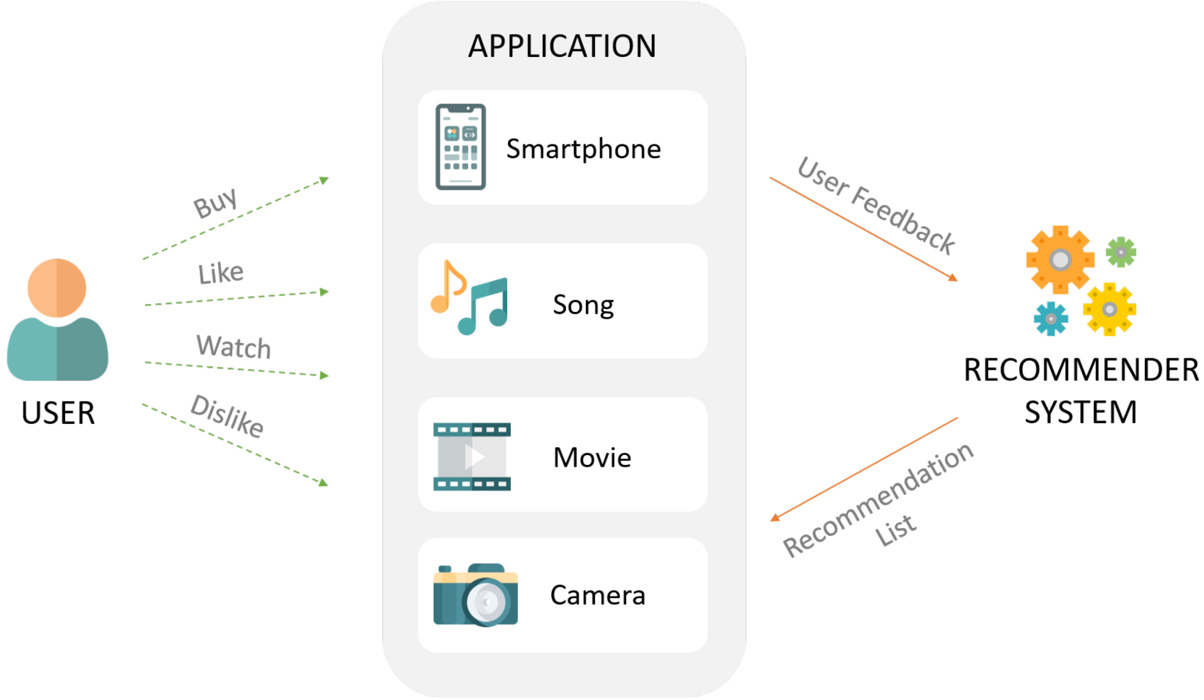
Project:
Hierarchical Poisson Factorization

Building Recommender Systems With Markovian Inference

Problem: Recommender Systems

- Recommender systems are algorithms aimed at suggesting relevant items to users.
- Critical in many industries, their applications include:
  - News, shopping and streaming websites/ apps.
  - Social Media platforms.
  - Stock Trading support systems.
  - Developing and/or expanding knowledge bases (e.g. data lakes).

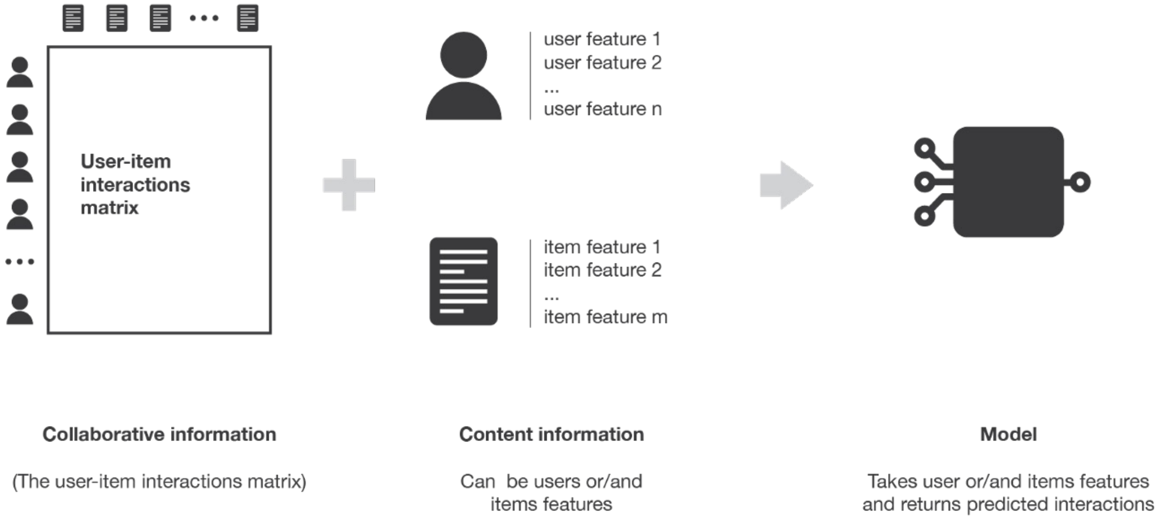


Types of Recommender Systems

Collaborative Filtering Methods

Users	User-item interactions matrix	Items
suscribers	rating given by a user to a movie (integer)	movies
readers	time spent by a reader on an article (float)	articles
buyers	product clicked or not when suggested (boolean)	products
	...	

Content-Based Systems



Types of Recommender Systems

Collaborative Filtering Methods

- Based solely on the past interactions between users and items, stored in the a user-item interaction matrix.
- Pros:
  - Main advantage is that they require no information about users or items.
  - The more users interact with items the more new recommendations become accurate
- Cons:
  - Main disadvantage: "cold start problem": impossible to recommend anything to new users or users with few interactions.

Content-Based Systems

- Use additional information about users and/or items on top of past user interactions.
- Build a model, based on the available "features", that explain the observed user-item interactions.
- Content based methods suffer far less from the cold start problem than collaborative approaches: new users or items can be described by their characteristics (content) and so relevant suggestions can be done for these new entities.

Hierarchical Poisson Factorizaton

1. For each user u:
   (a) Sample activity $\xi_u \sim \text{Gamma}(a', a'/b')$.
   (b) For each component k, sample preference

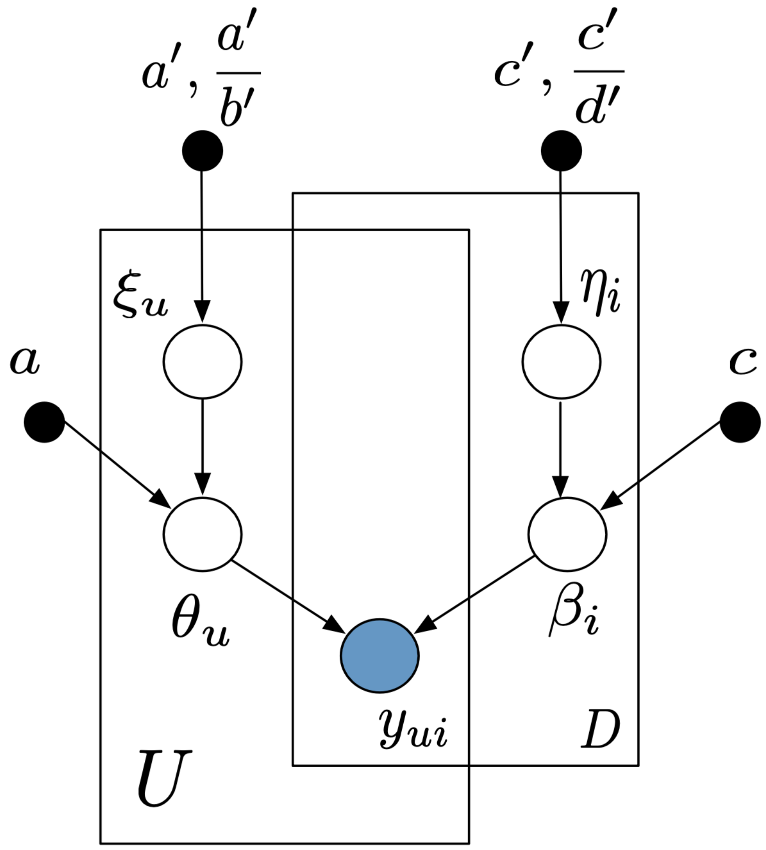
$$\theta_{uk} \sim \text{Gamma}(a, \xi_u).$$

2. For each item i:
   (a) Sample popularity $\eta_i \sim \text{Gamma}(c', c'/d')$.
   (b) For each component k, sample attribute

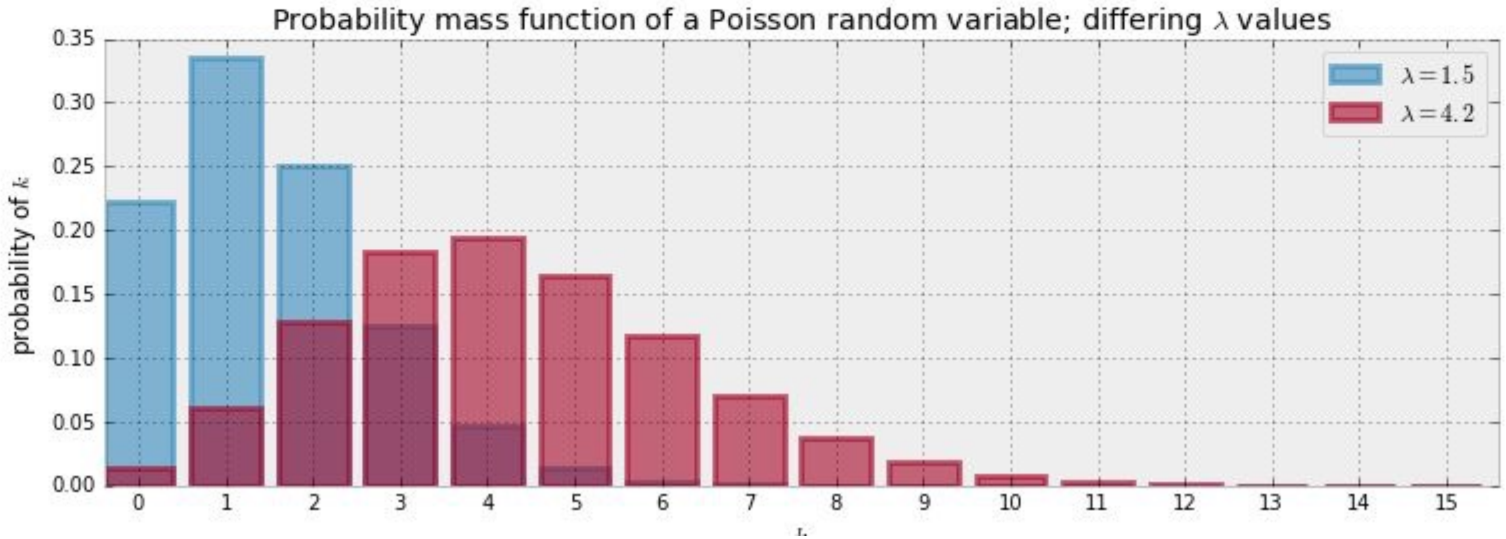
$$\beta_{ik} \sim \text{Gamma}(c, \eta_i).$$

3. For each user u and item i, sample rating

$$y_{ui} \sim \text{Poisson}(\theta_u^\top \beta_i).$$



Lecture 3 Reminder: Poisson Distribution



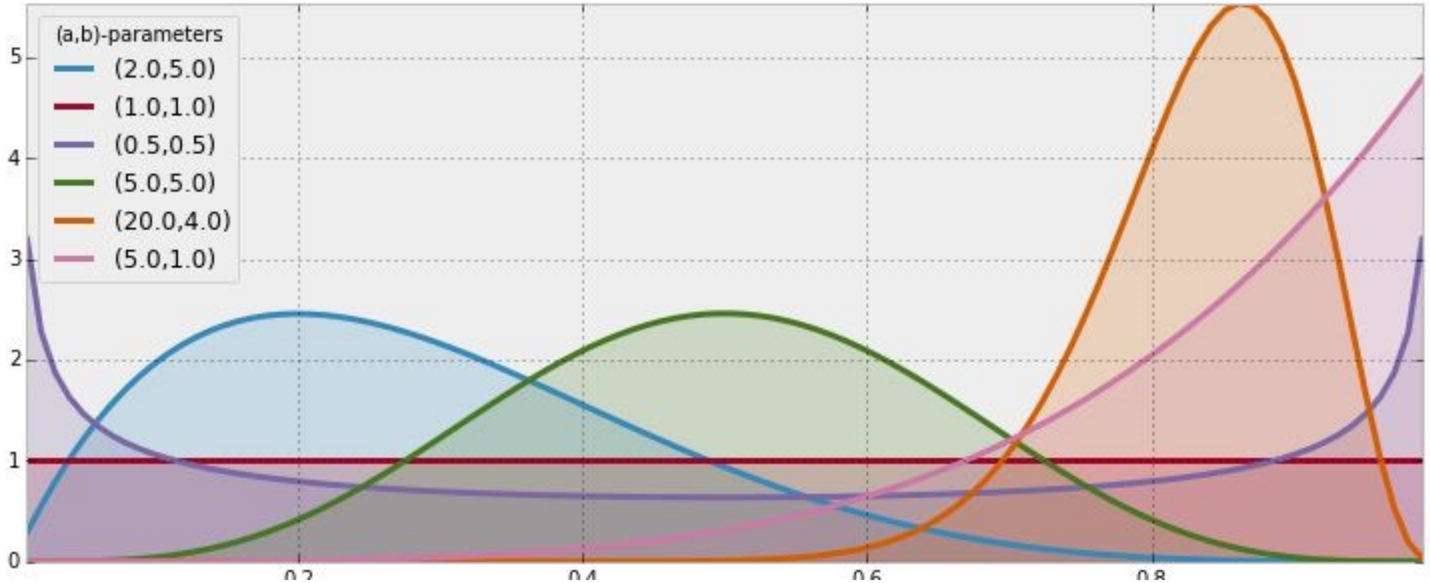
$$Z \sim \text{Poi}(\lambda)$$

$$P(Z = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \qquad k = 0,1,2,\ldots, \qquad \lambda \in \mathbb{R}_{>0}$$

$$E(Z|\lambda) = \sum_{k=0}^{\infty} kP(Z = k) = \lambda$$

```
Z ~ poisson(lambda)
```

Lecture 3 Reminder: Beta Distribution



$$Z \sim \text{Beta}(\alpha, \beta)$$

$$f_Z(z|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} z^{\alpha-1}(1-z)^{\beta-1},\ \alpha > 0, \beta > 0, 0 < z < 1$$

$$E(Z|\alpha,\beta) = \int_{-\infty}^{\infty} z f_Z(z|\alpha,\beta)dz = \frac{\alpha}{\alpha+\beta}$$

```
Z ~ beta(alpha, beta)
```

Task: Implement & Train HPF using MCMC

- Implement HPF using PyMC and MCMC as presented in Scalable Recommendation with Hierarchical Poisson Factorization.
  - The authors use Variational Inference to scale the model, but we want you to use MCMC in your implementation.
  - Use a default value of 0.3 for all Gamma distribution hyperparams.
  - After a successful training on a comprehensive enough dataset (definition in the next slide), you can attempt to perform hyperparameter tuning on your model for your specific data dsitribution.

HPF & MCMC's Pitfall!

- Since MCMC is less scalable than VI, your should first choose a recommender system dataset off Kaggle, then reduce the number of samples such that its dimensionality is feasible for MCMC.
  - **Example:** Use a combination of 10 users and 10 items as a sanity check. After this, increase dimensionality to the limit that MCMC allows.
  - **Nota Bene:** Pay attention to the type of users and items you choose for the sanity check dataset! If one user is heavily into high heels and another is interested in car engines (see stratified sampling), then even a linear classifier will be able to differ between the two.

Friendly Suggestions

- Choose different recommender system subsets out of common datasets.
  - It would be really suspicious  if all of you would pick the same X users and Y items from the same dataset out of the hundreds available on Kaggle.
  - We do not discourage repetition among some of you.
- Allocate more than a couple of hours for training your final models. MCMC may take up to 12 or more hours to converge!
  - You can do multiple quick training sessions on your sanity check dataset (this can be the same amongst you all).

Project Grading Criteria

- You should think of yourselves as ML/DL consultants and of us as your clients.
- We are not only interested in a cleanly-written, working implementation of HPF, but also in a thorough understanding of what you have accomplished.
- Whatever **Python code** format you chose to submit your project in (.ipynb or .py), we expect an acompanying documentation, containing…

Project Grading Criteria: Expectations

- How and why have you curated your dataset the way you have? Why did you pick these X users are Y items? What are the relationships between them? Does the model replicate the relationships that you have envisioned?
- Technical approach details: Why did you implement HPF your way? How did you choose your hyperparameters and MCMC training steps? How did you evaluate your model? What metrics did you use? How does the test distribution differ from the train and validation distribution?
- Relevant (sub)dataset statistics (presented in table format) and plots, as well as statistics and plots of your resulting MCMC outputs, comparisons between sanity check and the extended subdataset.

Project Grading Criteria: Extra

- **Extra:** Tables representing comparisons between multiple runs (several hyperparameter sets, different MCMC steps, or both), as well as the accompanying plots.
- **Extra:** Comparison with other models: you can modify LDA (example project presented soon™ at the lab, so stay tuned!) to work as a recommender system.
- **Extra:** Comparison with a nonparametric model (which means writing your own custom distributions), as presented in Bayesian Nonparametric Poisson Factorization for Recommendation Systems.
- **Extra:** An HPF implementation using Variational Inference.
- **Extra:** Custom MCMC sampling schema proposal.