Expectation-Maximization for Learning Determinantal Point Processes

Jennifer Gillenwater — jengi@cis.upenn.edu
Alex Kulesza — kulesza@umich.edu
Emily Fox — ebfox@stat.washington.edu
Ben Taskar — taskar@cs.washington.edu

TASK: SUBSET SELECTION

Example: product recommendation
Two goals: relevance and diversity

Relevance only:

Relevance + diversity:

SIMILARITY KERNEL

$$K = \begin{bmatrix} 0.5 & 0.38 & 0.0 \\ 0.38 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.3 \end{bmatrix}$$

Constraints: $0 \preceq K \preceq 1$

DETERMINANTAL POINT PROCESS

$$\boldsymbol{Y} \sim \mathrm{DPP}(K) \implies \mathcal{P}(Y \subseteq \boldsymbol{Y}) = \det(K_Y)$$

$\mathcal{P}(\quad \subseteq \boldsymbol{Y}) = \det(K_{\{1\}}) = K_{11} = 0.5$

$\mathcal{P}(\quad \subseteq \boldsymbol{Y}) = \det(K_{\{1,2\}}) = \det\begin{pmatrix} 0.5 & 0.38 \\ 0.38 & 0.4 \end{pmatrix} = 0.05$

$\mathcal{P}(\quad \subseteq \boldsymbol{Y}) = \det(K_{\{1,3\}}) = \det\begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.3 \end{pmatrix} = 0.15$

DPP INFERENCE

Exact and efficient inference: $O(N^3)$ or faster

- marginalization: $\mathcal{P}(Y \subseteq \boldsymbol{Y}) = \det(K_Y)$
- normalized probabilities: $\mathcal{P}(\boldsymbol{Y} = Y) = |\det(K - I_{\overline{Y}})|$
- conditioning: $\mathcal{P}(\boldsymbol{Y} = Y \mid A \subseteq \boldsymbol{Y}), \mathcal{P}(\boldsymbol{Y} = Y \mid A \cap \boldsymbol{Y} = \emptyset)$
- sampling: $Y \sim \mathrm{DPP}(K)$

But: DPP max likelihood learning is (conjectured) NP-hard

DPP LOG-LIKELIHOOD

Ground set of N items:

Training data (example subsets):

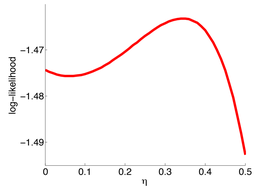
Y_1
Y_2
Y_3
$\vdots$
Y_T

$$\mathcal{L}(K) = \sum_{t=1}^{T} \log\left(|\det(K - I_{\overline{Y}_t})|\right)$$

LOG-LIKELIHOOD IS NON-CONCAVE

Training data:

$$K = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.4 \end{bmatrix} + \eta \begin{bmatrix} 0.1 & 0.6 \\ 0.6 & -0.1 \end{bmatrix}$$



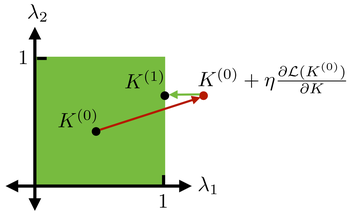
PRIOR WORK

Quality learning: learn weight for each row of K *Kulesza and Taskar (UAI 2011)*	**Sum-of-experts model**: weighted sum of fixed DPPs *Kulesza and Taskar (ICML 2011)*
Parametric kernel: K assumed Gaussian, poly, etc. *Affandi et al. (ICML 2014)*	**This work:** no fixed values or restrictive parameterizations for K

BASELINE: PROJECTED GRADIENT

Gradient: $\dfrac{\partial \mathcal{L}(K)}{\partial K} = \sum_{t=1}^{T} (K - I_{\overline{Y}_t})^{-1}$

Project to satisfy constraints on eigenvalues of K: $\mathbf{0} \le \boldsymbol{\lambda} \le \mathbf{1}$

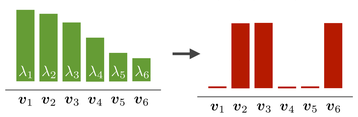


GENERATIVE DPP SAMPLING

Eigendecompose $K = V \Lambda V^\top$

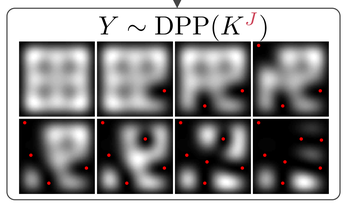
Sample "hidden" eigenvectors J

$$K^J = V^J \mathrm{diag}(\mathbf{1})(V^J)^\top$$



Sample subset based on J

$$Y \sim \mathrm{DPP}(K^J)$$



Main idea: Exploit hidden variable J to develop EM-style optimization

EXPECTATION-MAXIMIZATION

Main idea: Exploit hidden variable J to develop EM-style optimization

$$\begin{aligned} \mathcal{P}(Y) &= |\det(K - I_{\overline{Y}})| \\ &= \sum_{J: J \subseteq \{1,\dots,N\}} \mathcal{P}_{K^J}(Y) \prod_{j: j \in J} \lambda_j \prod_{j: j \notin J} (1 - \lambda_j) \\ &= \sum_{J: J \subseteq \{1,\dots,N\}} p_K(Y \mid J) p_K(J) \end{aligned}$$

LOWER-BOUNDING

$$\begin{aligned} \mathcal{L}(K) = \mathcal{L}(V, \Lambda) &= \sum_{t=1}^{T} \log p_K(Y_t) = \sum_{t=1}^{T} \log \left(\sum_J p_K(J, Y_t) \right) \\ &= \sum_{t=1}^{T} \log \left(\sum_J q(J \mid Y_t) \frac{p_K(J, Y_t)}{q(J \mid Y_t)} \right) \\ &\ge \sum_{t=1}^{T} \sum_J q(J \mid Y_t) \log \left(\frac{p_K(J, Y_t)}{q(J \mid Y_t)} \right) \equiv F(q, V, \Lambda) \end{aligned}$$

COORDINATE ASCENT

E-step: $\max_q F = \min_q \sum_{t=1}^{T} \mathbf{KL}(q(J \mid Y_t) \parallel p_K(J \mid Y_t))$ — concave in q

M-step: $\max_{V,\Lambda} F = \max_{V,\Lambda} \sum_{t=1}^{T} \mathbb{E}_q[\log p_K(J, Y_t)]$ — concave in Λ

s.t. $\mathbf{0} \le \boldsymbol{\lambda} \le \mathbf{1}, V^\top V = I$ — not concave in V

E-STEP

Set distributions equal to minimize KL:

$$q(J \mid Y_t) = p_K(J \mid Y_t)$$

Theorem: If $p_K(Y \mid J)$ is a DPP distribution compactly parameterized by K, then $p_K(J \mid Y)$ is also a DPP distribution with a related compact parameterization computable in at most $O(N^3)$ time.

$$K_Y = [V \Lambda V^\top]_Y = V_Y \Lambda (V_Y)^\top \implies$$

$$q(J \mid Y) \propto \det\left(\left[V_Y^\top V_Y \mathrm{diag}\left(\frac{\boldsymbol{\lambda}}{\mathbf{1} - \boldsymbol{\lambda}} \right) \right]_J \right)$$

M-STEP

EIGENVALUES

The eigenvalue update is closed-form. Constraints are implicitly enforced, since marginals of q are in [0,1]. (No need for a projection step.)

$$\lambda_j = \frac{1}{T} \sum_{t=1}^{T} \sum_{J: j \in J} q(J \mid Y_t) = \frac{1}{T} \sum_{t=1}^{T} q(j \in J \mid Y_i)$$

Because q is a DPP, the marginalization required for this update is efficient. $O(TNk^2)$, for $k = \max_t |Y_t|$

EIGENVECTORS

The eigenvector update is more complicated, but still efficient. The constraints correspond to optimization over Stiefel manifold, and Edelman et al. (SIMAX 1998) show that the below update stays on the manifold.

$$V' \leftarrow V \exp\left[\eta \left(V^\top \frac{\partial \mathcal{L}}{\partial V} - \left(\frac{\partial \mathcal{L}}{\partial V} \right)^\top V \right) \right]$$

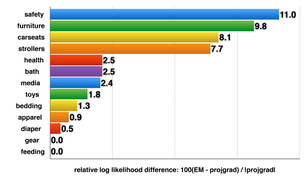
PRODUCT RECOMMENDATION DATASET

~30,000 Amazon baby registries, 13 product categories

Goal: learn a DPP for each category

LOG-LIKELIHOOD RESULTS

NAIVE INITIALIZATION



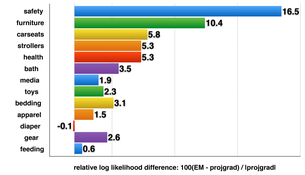
Draw starting point from a Wishart with N degrees of freedom

MOMENTS-MATCHING INITIALIZATION

Data-dependent starting point

$$m_i = \frac{1}{T} \sum_{t=1}^{T} 1(i \in Y_t), \quad m_{ij} = \frac{1}{T} \sum_{t=1}^{T} 1(i \in Y_t \wedge j \in Y_t)$$

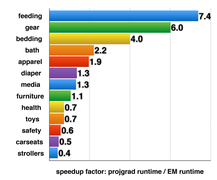
$$\tilde{K}_{ii} = m_i, \quad \tilde{K}_{ij} = \sqrt{\max\left(\tilde{K}_{ii} \tilde{K}_{jj} - m_{ij}, 0 \right)}$$



Using all data: EM and baseline are comparable

Low-data setting $T = 2N$

RUNTIME RESULTS



$$\text{EM}: O(T_1 T_2 N^3 + T_1 T N k^2)$$
$$\text{Proj grad}: O(T_1 T_2 N^3 + T_1 T N^3)$$

T_1 = iterations to converge
T_2 = iterations to find a step size
T = # of example sets
N = ground set size

EX: "SAFETY" CATEGORY

Left: 10-item registry for the "safety" category under the model learned by projected gradient. Right: The EM model replaces two redundant items (crossed out) with two more diverse items.

Code + Data: https://code.google.com/p/em-for-dpps