Supplementary Materials for Learning the Parameters of Determinantal Point Process Kernels

Raja Hafiz Affandi RAJARA@WHARTON.UPENN.EDU
Department of Statistics, University of Pennsylvania

Emily B. Fox EBFOX@STAT.WASHINGTON.EDU
Department of Statistics, University of Washington

Ryan P. Adams RPA@SEAS.HARVARD.EDU
Department of Statistics, Harvard Univerity

Ben Taskar TASKAR@CS.WASHINGTON.EDU
Department of Computer Science & Engineering, University of Washington

1. Gradient for Discrete DPP

Gradient ascent and stochastic gradient ascent provide attractive approaches in learning parameters, Θ of DPP kernel $L(\Theta)$ because of their theoretical guarantees, but require knowledge of the gradient of the log-likelihood $\mathcal{L}(\Theta)$. In the discrete DPP setting, this gradient can be computed straightforwardly and we provide examples for discrete Gaussian and polynomial kernels here.

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log(\det(L_{A^t}(\Theta))) - T\log(\det(L(\Theta)+I)) \ ,$$

$$\begin{aligned}\frac{d\mathcal{L}(\Theta)}{d\Theta} = \ & \textstyle\sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(\Theta)^{-1}\frac{dL_{A^t}(\Theta)}{d\Theta}\right) \\ & -T\mathrm{tr}\left((L(\Theta)+I)^{-1}\frac{dL(\Theta)}{d\Theta}\right) \ .\end{aligned}$$

To find the MLE, we can perform gradient ascent

$$\Theta_i = \Theta_{i-1} + \eta\frac{dl(\Theta)}{d\Theta} \ .$$

In the following examples, we denote $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d)})$, where d is the number of dimension.

1.1. Example I: Gaussian Similarity with Uniform Quality

$L(\Sigma) = \exp\{-(\mathbf{x}-\mathbf{y})^\top\Sigma^{-1}(\mathbf{x}-\mathbf{y})\}$.

Denote $G_{ij}^{(lm)} = L_{ij}\frac{(x_i^{(l)}-x_j^{(l)})(x_i^{(m)}-x_j^{(m)})}{2\Sigma_{lm}^2}$.

Then,

$$\begin{aligned}\frac{d\mathcal{L}(\Sigma)}{d\Sigma_{lm}} = \ & \textstyle\sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(\Sigma)^{-1}G_{A^t}^{(lm)}\right) \\ & -T\mathrm{tr}\left((L(\Sigma)+I)^{-1}G^{(lm)}\right) \ .\end{aligned}$$

1.2. Example II: Gaussian Similarity with Gaussian Quality

$L(\Gamma,\Sigma)$
$= \exp\{-\mathbf{x}^\top\Gamma^{-1}\mathbf{x} - (\mathbf{x}-\mathbf{y})^\top\Sigma^{-1}(\mathbf{x}-\mathbf{y}) - \mathbf{y}^\top\Gamma^{-1}\mathbf{y}\}$.
Denote $C_{ij}^{(lm)} = L_{ij}\frac{(x_i^{(l)}x_i^{(m)}+x_j^{(l)}x_j^{(m)})}{2\Gamma_{lm}^2}$ and $G_{ij}^{(lm)}$ as in previous example.

Then,

$$\begin{aligned}\frac{d\mathcal{L}(\Gamma,\Sigma)}{d\Gamma_{lm}} = \ & \textstyle\sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(\Sigma)^{-1}C_{A^t}^{(lm)}\right) \\ & -T\mathrm{tr}\left((L(\Sigma)+I)^{-1}C^{(lm)}\right)\end{aligned}$$

and $\frac{dl(\Gamma,\Sigma)}{d\Sigma_{lm}}$ is the same as the previous example.

1.3. Example III: Polynomial Similarity with Uniform Quality

$L(p,q) = \left(\mathbf{x}^\top\mathbf{y}+p\right)^q$.

Denote $R_{ij} = qL_{ij}^{\frac{q-1}{q}}$ and $U_{ij} = L_{ij}\log(L_{ij}^{\frac{1}{q}})$.

Then,

$$\begin{aligned}\frac{d\mathcal{L}(p,q)}{dp} = \ & \textstyle\sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(p,q)^{-1}R_{A^t}\right) \\ & -T\mathrm{tr}\left((L(p,q)+I)^{-1}R)\right) \ ,\end{aligned}$$

Algorithm 1 Random-Walk Metropolis-Hastings

Input: Dimension: D, Starting point: Θ_0, Prior distribution: $\mathcal{P}(\Theta)$, Proposal distribution $f(\hat{\Theta}|\Theta)$ with mean Θ, Samples: $A^1, \ldots, A^T$.
$\Theta = \Theta_0$
for $i = 0 : (\tau - 1)$ **do**
  $\hat{\Theta} \sim f(\hat{\Theta}|\Theta_i)$
  $r = \left(\frac{\mathcal{P}(\hat{\Theta}|A^1,\ldots,A^T)}{\mathcal{P}(\Theta_i|A^1,\ldots,A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)} \right)$
  $u \sim$ Uniform[0,1]
  if $u < \min\{1, r\}$ **then**
    $\Theta_{i+1} = \hat{\Theta}$
Output: $\Theta_{0:\tau}$

$$\frac{d\mathcal{L}(p,q)}{dq} = \sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(p,q)^{-1} U_{A^t}\right) - T\mathrm{tr}\left((L(p,q)+I)^{-1}U\right) .$$

2. Bayesian Learning

In the main paper, we highlight two techniques: random-walk Metropolis-Hastings (MH) and slice sampling to sample from the posterior distribution. We present the pseudo-code for these algorithms here (Alg. 1 and Alg. 2).

In Alg. 3, we also present the pseudo-code for the random-walk MH algorithm for handling large-scale discrete and continuous DPPs using posterior bounds. Finally, we present an illustration of slice sampling using posterior bounds in Figure 1.

3. Proof of DPP/kDPP Denominator Bounds

In the main paper, we show that the lower and upper posterior probability bounds for the DPP/kDPP can be incorporated in many MCMC algorithms, and provide an effective means of garnering posterior samples assuming the bounds can be efficiently tightened. Below, we provide the proofs to Propositions 3.1 and 3.2, which bound the denominator of the posterior probability using eigenvalue truncations.

Proposition 3.1 *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L. Then*

$$\prod_{n=1}^{M}(1+\lambda_n) \leq \prod_{n=1}^{\infty}(1+\lambda_n) \tag{1}$$

and

$$\prod_{n=1}^{\infty}(1+\lambda_n) \leq \exp\left\{\mathrm{tr}(L) - \sum_{n=1}^{M}\lambda_n\right\}\left[\prod_{n=1}^{M}(1+\lambda_n)\right] . \tag{2}$$

Proof: The first inequality is trivial since the eigenvalues $\lambda_{1:\infty}$ are all nonnegative.

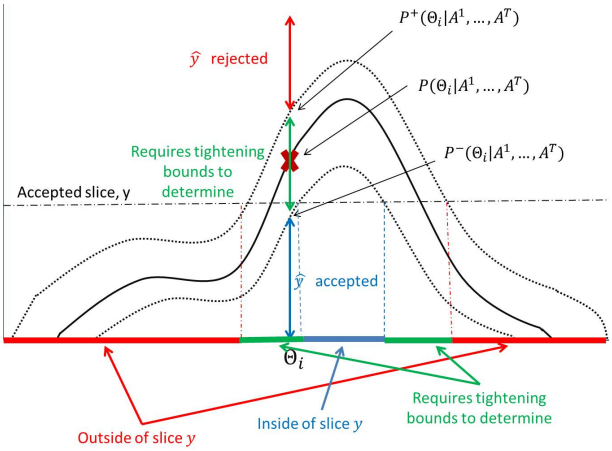

Figure 1. Illustration of slice sampling algorithm using posterior bounds. In the first step, a candidate slice $\hat{y}$ is generated. $\hat{y}$ is rejected if it is above the upper posterior bound and rejected if it is below the lower posterior bound. If $\hat{y}$ is in between the bounds, then the bounds are tightened until a decision can be made. Once a slice, y is accepted, we need to sample new parameters inside the slice. To determine whether the endpoints of the interval or the new parameters are in the slice we: (i) decide that they are in the slice if the upper bound of posterior probability evaluated at the points are higher than the slice value, or (ii) decide that they are outside of the slice if the lower bound of the posterior probability is lower than the slice value. Otherwise, we tighten the bounds until a decision can be made.

Algorithm 2 Univariate Slice Sampling

Input: Starting point: Θ_0, Initial width: w, Prior distribution: $\mathcal{P}(\Theta)$, Samples: $A = [A^1, \ldots, A^T]$.
$\Theta = \Theta_0$
for $i = 0 : (\tau - 1)$ **do**
  $y \sim \text{Uniform}[0, \mathcal{P}(\Theta_i|A^1, \ldots, A^T)]$
  $z \sim \text{Uniform}[0, 1]$
  $L = \Theta_i - z * \frac{w}{2}$
  $R = L + \frac{w}{2}$
  while $y > \mathcal{P}(L|A^1, \ldots, A^T)$ **do**
    $L = L - \frac{w}{2}$
  while $y > \mathcal{P}(R|A^1, \ldots, A^T)$ **do**
    $R = R + \frac{w}{2}$
  $\hat{\Theta} \sim \text{Uniform}[L, R]$
  if $\mathcal{P}(\hat{\Theta}|A^1, \ldots, A^T) < y$ **then**
    while $\mathcal{P}(\hat{\Theta}|A^1, \ldots, A^T) < y$ **do**
      if $\hat{\Theta} > \Theta$ **then**
        $R = \hat{\Theta}$
      else
        $L = \hat{\Theta}$
      $\hat{\Theta} \sim \text{Uniform}[L, R]$
  $\Theta_{i+1} = \hat{\Theta}$
Output: $\Theta_{0:\tau}$

To prove the second inequality, we use the AM-GM inequality: For any non-negative numbers, $\gamma_1, ..., \gamma_M, (\prod_{n=1}^{M} \gamma_n)^{\frac{1}{M}} \leq \sum_{n=1}^{M} \frac{\gamma_n}{M}$.

Let $\Lambda_M = \sum_{n=M+1}^{\infty} \lambda_n$ and $\gamma_n = 1 + \lambda_n$. Then,

$$\begin{aligned}
\prod_{n=1}^{\infty}(1+\lambda_n) &= \prod_{n=1}^{\infty}\gamma_n = (\prod_{n=1}^{M}\gamma_n)(\prod_{n=M+1}^{\infty}\gamma_n) \\
&= (\prod_{n=1}^{M}\gamma_n)(\lim_{l\to\infty}\prod_{n=M+1}^{M+l}\gamma_n) \\
&\leq (\prod_{n=1}^{M}\gamma_n)(\lim_{l\to\infty}(\sum_{n=M+1}^{M+l}\frac{\gamma_n}{l})^l) \\
&\leq (\prod_{n=1}^{M}(1+\lambda_n))\exp(\Lambda_M)\,.
\end{aligned}$$

□

Proposition 3.2 *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L. Then*

$$e_k(\lambda_{1:M}) \leq e_k(\lambda_{1:\infty}) \tag{3}$$

and

$$e_k(\lambda_{1:\infty}) \leq \sum_{j=0}^{k} \frac{(\text{tr}(L) - \sum_{n=1}^{M}\lambda_n)^j}{j!} e_{k-j}(\lambda_{1:M})\,. \tag{4}$$

Proof: Let $e_k(\lambda_{1:m})$ be the kth elementary symmetric function: $e_k(\lambda_{1:m}) = \sum_{J\subseteq\{1,\ldots,m\},|J|=k} \prod_{j\in J}\lambda_j$.

Algorithm 3 Random-Walk Metropolis-Hastings with Posterior Bounds

Input: Dimension: D, , Starting point: Θ_0, Prior distribution: $\mathcal{P}(\Theta)$, Proposal distribution $f(\hat{\Theta}|\Theta)$ with mean Θ, samples: $A = [A^1, \ldots, A^T]$.
$\Theta = \Theta_0$
for $i = 0 : \tau$ **do**
  $\hat{\Theta} \sim f(\hat{\Theta}|\Theta_i)$
  $r_+ = \infty, r_- = -\infty$
  $u \sim \text{Uniform}[0,1]$
  while $u \in [r_-, r_+]$ **do**
    $r^+ = \left(\frac{\mathcal{P}^+(\hat{\Theta}|A^1,\ldots,A^T)}{\mathcal{P}^-(\Theta_i|A^1,\ldots,A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right)$
    $r^- = \left(\frac{\mathcal{P}^-(\hat{\Theta}|A^1,\ldots,A^T)}{\mathcal{P}^+(\Theta_i|A^1,\ldots,A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right)$
    Increase tightness on $\mathcal{P}^+$ and $\mathcal{P}^-$
  if $u < \min\{1, r^-\}$ **then**
    $\Theta_t = \hat{\Theta}$
Output: $\Theta_{0:\tau}$

Trivially, we have a lower bound since the eigenvalues $\lambda_{1:\infty}$ are non-negative: $e_k(\lambda_{1:m}) \leq e_k(\lambda_{1:n}) \quad \text{for } m \leq n$.

For the upper bound we can use the Schur-concavity of elementary symmetric functions for non-negative arguments (Guan, 2006). Thus, for $\bar{\lambda}_{1:N} \prec \lambda_{1:N}$:

$$\sum_{i=1}^{k}\bar{\lambda}_n \leq \sum_{n=1}^{k}\lambda_n \quad \text{for } k = 1, \ldots, N-1 \tag{5}$$

and

$$\sum_{n=1}^{N}\bar{\lambda}_n = \sum_{n=1}^{N}\lambda_n\,, \tag{6}$$

we have $e_k(\bar{\lambda}_{1:N}) \geq e_k(\lambda_{1:N})$.

Now let $\Lambda_M = \sum_{n=M+1}^{\infty}$ and $\Lambda_M^N = \sum_{n=M+1}^{N}$. We consider

$$\bar{\lambda}_{1:N}^{(M)} = (\lambda_1, \ldots, \lambda_M, \frac{\Lambda_M^N}{N-M}, \ldots, \frac{\Lambda_M^N}{N-M}). \tag{7}$$

Note that $\bar{\lambda}_{1:N}^{(M)} \prec \lambda_{1:N}$ and so $e_k(\bar{\lambda}_{1:N}^{(M)}) \geq e_k(\lambda_{1:N})$ for $M < N$.

We now compute $e_k(\bar{\lambda}_{1:N}^{(M)})$. Note that for $e_k(\bar{\lambda}_{1:N}^{(M)})$, the terms in the sum are products of k factors, each containing some of the $\lambda_{1:M}$ factors and some of the $\frac{\Lambda_M^N}{N-M}$ factors. The sum of the terms that have j factors of type $\frac{\Lambda_M^N}{N-M}$ is $\binom{N-M}{j}\left(\frac{\Lambda_M^N}{N-M}\right)^j e_{k-j}(\Lambda(m))$, so we have:

$$e_k(\bar{\lambda}_{1:N}^{(M)}) = \sum_{j=0}^{k}\binom{N-M}{j}\left(\frac{\Lambda_M^N}{N-M}\right)^j e_{k-j}(\lambda_{1:M})\,.$$

Using $\binom{N-M}{j} \leq \frac{(N-M)^j}{j!}$, we get

$$e_k(\bar{\lambda}^{(M)}_{1:N}) = \sum_{j=0}^{k} \left(\frac{(\Lambda^N_M)^j}{j!} \right) e_{k-j}(\lambda_{1:M}) \, .$$

Letting $N \to \infty$, we get out upper bound

$$e_k(\lambda_{1:\infty}) \leq \sum_{j=0}^{k} \left(\frac{(\Lambda_M)^j}{j!} \right) e_{k-j}(\lambda_{1:M}) \quad \text{for } m \leq n.$$

4. DPP/kDPP Numerator Bounds

In the main paper, we develop a Bayesian method that only requires an upper and lower bound on the likelihood. There, we focus on the large N challenge by bounding the denominator of the posterior probability. An analogous method can be used for handling large observation sets, A_t, by bounding the numerator term, $\det(L_{A_t}(\Theta))$. Assume that the size of a particular observation A_t is K. Let $\xi_1, \xi_2 \ldots \xi_K$ denote the eigenvalues of L_{A_t}. We can then express $\det(L_{A_t}(\Theta))$ as

$$\det(L_{A_t}(\Theta)) = \prod_{n=1}^{K} \xi_n \, . \tag{8}$$

We can find a truncation of the eigenvalues, $\xi_1, \ldots, \xi_M$ $(M < K)$, using methods such as power iteration. Below we provide the numerator bounds based on the eigenvalue truncation that can be arbitrarily tightened by including more eigenvalue terms.

Proposition 4.1 *Let $\xi_{1:K}$ be the eigenvalues of kernel L_{A_t}. Then for $M < K$,*

$$\prod_{n=1}^{M} \xi_n \leq \prod_{n=1}^{K} \xi_n \tag{9}$$

and

$$\prod_{n=1}^{K} \xi_n \leq \left[\prod_{n=1}^{M} \xi_n \right] \left[\left(\frac{1}{K-M} \right) \left(\text{tr}(L_{A_t}) - \sum_{n=1}^{M} \xi_n \right) \right]^{K-M} . \tag{10}$$

Proof: The first inequality is trivial since the eigenvalues $\xi_{1:K}$ are all nonnegative.

To prove the second inequality, we use the AM-GM inequality: For any non-negative numbers, $\gamma_1, ..., \gamma_M$, $(\prod_{n=1}^{M} \gamma_n)^{\frac{1}{M}} \leq \sum_{n=1}^{M} \frac{\gamma_n}{M}$.

Then,

$$\begin{aligned} \prod_{n=1}^{K} \xi_n &= (\prod_{n=1}^{M} \xi_n)(\prod_{n=M+1}^{K} \xi_n) \\ &\leq \left[\prod_{n=1}^{M} \xi_n \right] \left[\left(\frac{1}{K-M} \right) \sum_{n=M+1}^{K} \xi_n \right]^{K-M} \\ &= \left[\prod_{n=1}^{M} \xi_n \right] \left[\left(\frac{1}{K-M} \right) \left(\text{tr}(L_{A_t}) - \sum_{n=1}^{M} \xi_n \right) \right]^{K-M} . \end{aligned}$$

□

5. Moments for Continuous DPP with Gaussian Quality and Similarity

In the continuous case, given the eigendecomposition of the kernel operator, $L(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{\infty} \lambda_n \phi_n(\mathbf{x})^* \phi_n(\mathbf{y})$ (where $\phi_n(\mathbf{x})^*$ denotes the complex conjugate of the nth eigenfunction), the mth moment can be evaluated as

$$E[\mathbf{x}^m] = \int_{\Omega} \sum_{n=1}^{\infty} \frac{\lambda_n}{\lambda_n + 1} \mathbf{x}^m \phi_n(\mathbf{x})^2 d\mathbf{x} \, . \tag{11}$$

Here we present the derivation for moments when

$$q(\mathbf{x}) = \sqrt{\alpha} \prod_{d=1}^{D} \frac{1}{\sqrt{\pi \rho_d}} \exp\left\{ -\frac{x_d^2}{2\rho_d} \right\} \tag{12}$$

and

$$k(\mathbf{x}, \mathbf{y}) = \prod_{d=1}^{D} \exp\left\{ -\frac{(x_d - y_d)^2}{2\sigma_d} \right\}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^D. \tag{13}$$

In this case, the eigenvalues and eigenvectors of the operator L are given by (Fasshauer & McCourt, 2012),

$$\lambda_{\mathbf{n}} = \alpha \prod_{d=1}^{D} \sqrt{\frac{1}{\frac{\beta_d^2+1}{2} + \frac{1}{2\gamma_d}}} \left(\frac{1}{\gamma_d(\beta_d^2+1)+1} \right)^{n_d - 1} , \tag{14}$$

and

$$\phi_{\mathbf{n}}(\mathbf{x}) = \prod_{d=1}^{D} \left(\frac{1}{\pi \rho_d^2} \right)^{\frac{1}{4}} \sqrt{\frac{\beta_d}{2^{n_d-1}\Gamma(n_d)}} \exp\left\{ -\frac{\beta_d^2 x^2}{2\rho_d^2} \right\} H_{n_d - 1}\left(\frac{\beta_d x_d}{\sqrt{\rho_d^2}} \right) , \tag{15}$$

where $\gamma_d = \frac{\sigma_d}{\rho_d}$, $\beta_d = (1 + \frac{2}{\gamma_d})^{\frac{1}{4}}$ and $\mathbf{n} = (n_1, n_2, \ldots, n_D)$ is a multi-index.

In the case of DPPs (as opposed to k-DPPs), we can use the number of items as an estimate of the 0th moment. The 0th moment is given by $\sum_{\mathbf{n}=1} \frac{\lambda_{\mathbf{n}}}{1+\lambda_{\mathbf{n}}}$. Denote

$\mathbf{x} = (x_1, x_2, \ldots, x_d)$. The for higher moments, note that

$$\begin{aligned} E[x_j^m] &= \int_{\mathbb{R}} \sum_{n=1}^{\infty} \frac{\lambda_n}{\lambda_n+1} x_j^m \phi_n(\mathbf{x})^2 dx_j \\ &= \sum_{n=1}^{\infty} \frac{\lambda_n}{\lambda_n+1} \int_{\mathbb{R}} x_j^m \phi_n(\mathbf{x})^2 dx_j \ . \end{aligned}$$

Using the results of moment integrals involving a product of two Hermite polynomials (Paris, 2010), we get that

$$E[x_j^m] = \int_{\mathbb{R}^d} \sum_{\mathbf{n}}^{\infty} \frac{\lambda_{\mathbf{n}}}{\lambda_{\mathbf{n}}+1} (\frac{\sqrt{\rho_j}}{\sqrt{2}\beta_j})^m \wp_{\frac{m}{2}}(n_j - 1) \quad (16)$$

for m even and 0 otherwise. The polynomial $\wp_{\frac{m}{2}}(n_j - 1)$ is given in Eq. (4.8) in Paris (2010). For example, the second and fourth moments are given by

(i) $E[x_j^2] = \sum_{\mathbf{n}}^{\infty} \frac{\lambda_{\mathbf{n}}}{\lambda_{\mathbf{n}}+1} (\frac{\sqrt{\rho_j}}{\sqrt{2}\beta_j})^2 (2n_j - 1)$,

(ii) $E[x_j^4] = \sum_{\mathbf{n}}^{\infty} \frac{\lambda_{\mathbf{n}}}{\lambda_{\mathbf{n}}+1} (\frac{\sqrt{\rho_j}}{\sqrt{2}\beta_j})^4 3(2n_j^2 - 2n_j + 1)$.

For a low dimensional setting, we can learn the parameters by using grid search such that the moments agree.

6. Details on Simulation

In the main paper, we use our Bayesian learning algorithms to learn parameters from (i) simulated data generated from a 2-dimensional isotropic discrete kernel ($\sigma_d = \sigma$, $\rho_d = \rho$ for $d = 1, 2$), (ii) nerve fiber data using 2-dimensional isotropic continuous kernel ($\sigma_d = \sigma$, $\rho_d = \rho$ for $d = 1, 2$) and (ii) image diversity data using 3600-dimensional discrete kernel with Gaussian similarity. In all of these experiments, we use weakly informative inverse gamma priors on σ, ρ and α. In particular, for all three parameters, we used the same priors for all three parameters

$$\mathcal{P}(\alpha) = \mathcal{P}(\rho) = \mathcal{P}(\sigma) = \text{Inv-Gamma}(0.001, 0.001) \ .$$

We then learn the parameters using hyperrectangle slice sampling.

7. Details on Image Diversity

In studying the diversity in images, we extracted 3 different types of features from the images—color features, SIFT-descriptors (Lowe, 1999; Vedaldi & Fulkerson, 2010) and GIST-descriptors (Oliva & Torralba, 2006). We describe these features below.

Color: Each pixel is assigned a coordinate in three-dimensional Lab color space. The colors are then sorted into axis-aligned bins, producing a histogram of either 8 (denoted color8) or 64 (denoted color64) dimensions.

SIFT: The images are processed to obtain sets of 128-dimensional SIFT descriptors. These descriptors are commonly used in object recognition to identify objects in images and are invariant to scaling, orientation and minor distortions. The descriptors for a given category are combined, subsampled to set of 25,000, and then clustered using k-means into either 256 (denoted SIFT256) or 512 (denoted SIFT512) clusters. The feature vector for an image is the normalized histogram of the nearest clusters to the descriptors in the image.

GIST: The images are processed to obtain 960-dimensional GIST feature vectors that are commonly used to describe scene structure.

We also extracted the features above from the center of the images, defined as the centered rectangle with dimensions half those of the original image. This yields a total of 10 different feature vectors. Since we are only concerned with the diversity of the images, we ensure that the quality across the images are uniform by normalizing each feature vector such that their L_2 norm equals to 1. We then combine the feature vectors into 3 types of features—color, SIFT and GIST.

For the Google Top 6 images, we model the samples, A^t_{Top6} as though they are generated from a 6-DPP with kernel $L^{subcat}(A^t)$. To highlight the effect of the human annotation in the partial results sets, we model the samples as though they are generated from a conditional 6-DPP.

In general, given a partial set of observations A and k-DPP kernel L, we can define the conditional k-DPP probability of choosing a set B given the inclusion of set A (with $|A| + |B| = k$) as

$$\mathcal{P}^k_L(Y = A \cup B | A \in Y) \propto \det(L^A_B) \ , \quad (17)$$

with

$$L^A = \left(\left[((L + I_{A^c})^{-1} \right]_{A^c} \right)^{-1} - I \ , \quad (18)$$

where I_{A^c} denotes the identity matrix with 0 along the diagonal corresponding to elements in A. Here, following the $N \times N$ inversion, the matrix is restricted to rows and columns indexed by elements not in A, then inverted again. The normalizer is given by (Kulesza & Taskar, 2012).

$$\sum_{|Y'|=k-|A|} \det(L^A_{Y'}) \ . \quad (19)$$

In our experiment, our samples can be seperated into the partial result sets and human annotations,

$$A^t_{\text{DPP+human}} = (A^t, b^t) \ , \quad (20)$$

where A^t is the partial result sets and b^t is the human annotated result, we model the data from the conditional 6-DPP $L^{subcat}(b^t | A^t)$. In this case, the likelihood is given by

$$L^i(\Theta^{cat}) = \frac{\det(L^{i\ A^t}_{b^t_t}(\Theta^{cat}))}{\sum_{i=1}^N L^{i\ A^t}_{x^i_i}(\Theta^{cat})} \quad (21)$$

for each subcategory, i. That is, for each subcategory, i, we compute $L^i(\Theta^{cat})$ and use Eq. (18) to compute the conditional kernel.

References

Fasshauer, G.E. and McCourt, M.J. Stable evaluation of Gaussian radial basis function interpolants. *SIAM Journal on Scientific Computing*, 34(2):737–762, 2012.

Guan, K. Schur-convexity of the complete elementary symmetric function. *Journal of Inequalities and Applications*, 2006(1):67624, 2006.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3), 2012.

Lowe, D. G. Object recognition from local scale-invariant features. In *Proc. IEEE International Conference on Computer Vision*, 1999.

Oliva, A. and Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36, 2006.

Paris, R.B. Asymptotics of integrals of hermite polynomials. *Appl. Math. Sci. 4*, pp. 3043–3056, 2010.

Vedaldi, A. and Fulkerson, B. Vlfeat: An open and portable library of computer vision algorithms. In *Proc. International Conference on Multimedia*, 2010.