Determinantal Point Processes

Anđelka Zečević
andjelkaz@matf.bg.ac.rs

Determinantal Point Processes (DPPs)

Problem of interest: How can we select a subset of a given set that is of a good quality and diversity at the same time?

Sources:

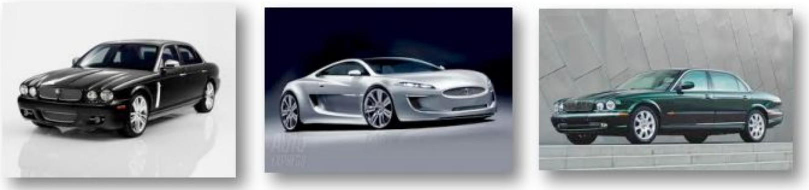
Alex Kulesza & Ben Taskar, 2013:
Determinantal Point Processes for Machine Learning

Alexei Borodin, 2009:
Determinantal Point Processes

Motivation - Information Retrieval

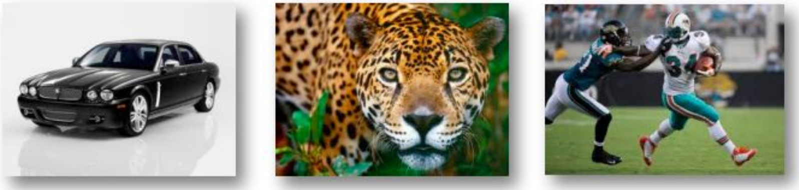
Image search: "jaguar"

Relevance only:



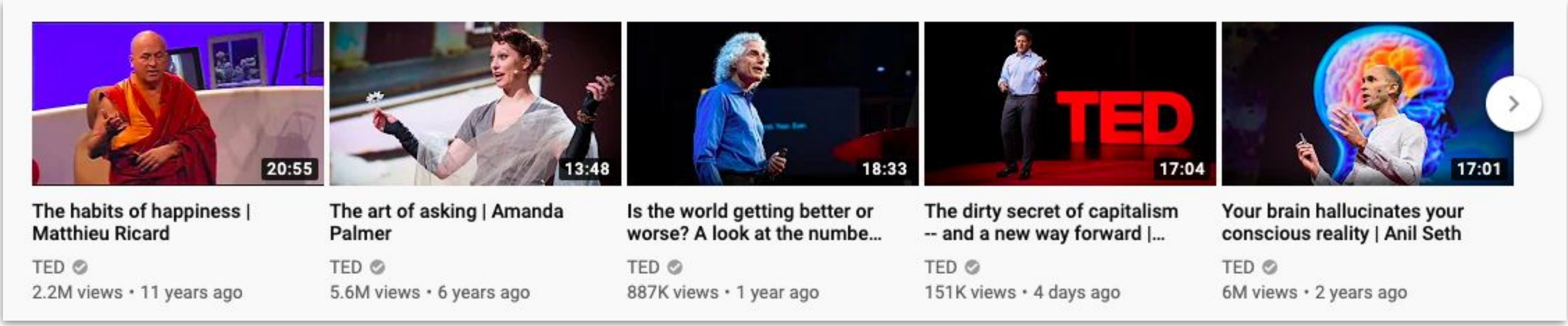
...

Relevance + diversity:



...

Motivation - Recommender Systems



The habits of happiness | Matthieu Ricard
TED
2.2M views • 11 years ago

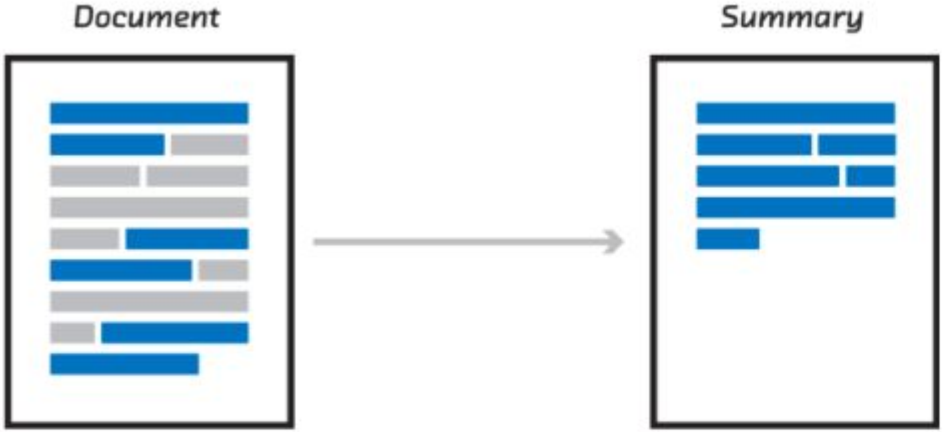
The art of asking | Amanda Palmer
TED
5.6M views • 6 years ago

Is the world getting better or worse? A look at the numbe...
TED
887K views • 1 year ago

The dirty secret of capitalism -- and a new way forward |...
TED
151K views • 4 days ago

Your brain hallucinates your conscious reality | Anil Seth
TED
6M views • 2 years ago

Motivation - Extractive Text Summarization



Discrete Point Processes

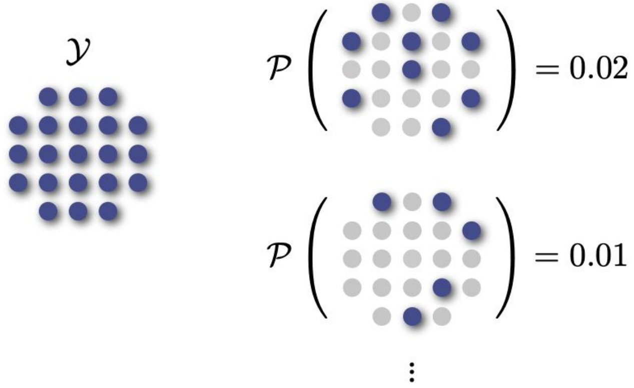
- Number of items (videos, sentences, …): $\boldsymbol{N}$
- Ground set: set of indexes

$$\mathcal{Y} = \{1, 2, ..., N\}$$

- Number of subsets: 2^N

- Point process:

Probability measure $\mathcal{P}$ over subsets $Y \subseteq \mathcal{Y}$



Discrete Point Processes

Independent Point Process

Each element i included with probability p_i

$$\mathcal{P}(Y) = \prod_{i \in Y} p_i \prod_{i \notin Y} (1 - p_i)$$

DPP - marginal kernel

Let $\mathbf{K} = [\mathbf{K}_{\mathbf{ij}}]$ be and $\mathbf{N} \times \mathbf{N}$ real, symmetric matrix with properties:

- $\mathbf{K}$ is a positive semidefinite matrix $x^T K x \geq 0$ for all $x \in R^N$

  all principal minors of are non-negative

  all eigenvalues are non-negative

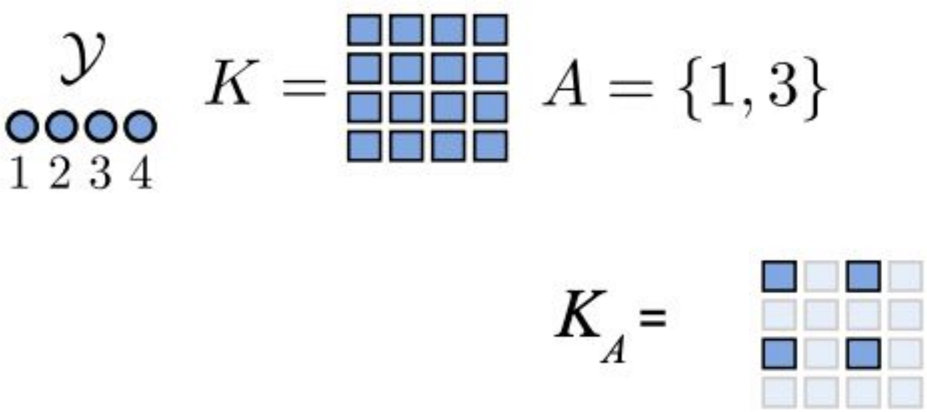
- all eigenvalues are bounded above by 1

Matrix **K** is so called marginal kernel.

Intuition: K_{ij} represents similarity of the items i and j

DPP - marginal kernel

If $\mathbf{A} \subseteq \mathcal{Y}$ than $\mathbf{K}_\mathbf{A} = [\mathbf{K}_{\mathbf{ij}}]_{\mathbf{i,j \in A}}$ is the matrix with rows and columns indexed by the elements of the set $\mathbf{A}$.



DPP - definition

Point process $\mathcal{P}$ is called determinantal point process if for a random subset $\boldsymbol{Y}$ drawn according to $\mathcal{P}$ for every $A \subseteq \mathcal{Y}$ holds

$$\mathcal{P}(A \subseteq \boldsymbol{Y}) = \det(K_A)$$

Adaptation: $\det(K_\emptyset) = 1$

Notation: Y ~ DPP(K)

DPP - diversification

For $A = \{i\}$:

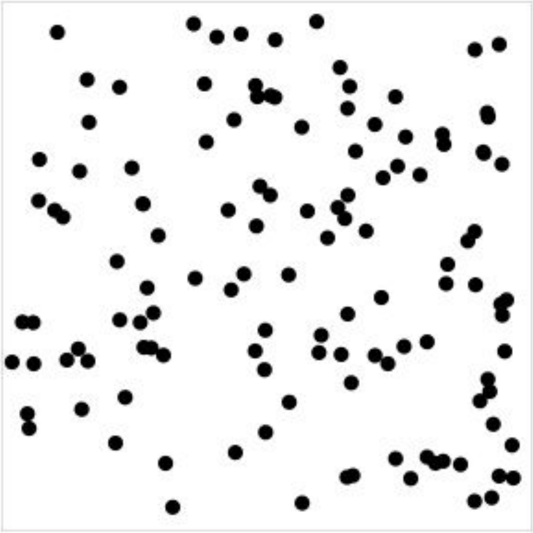
$$\mathcal{P}(\{i\} \subseteq Y) = K_{ii}$$

For $A = \{i, j\}$:

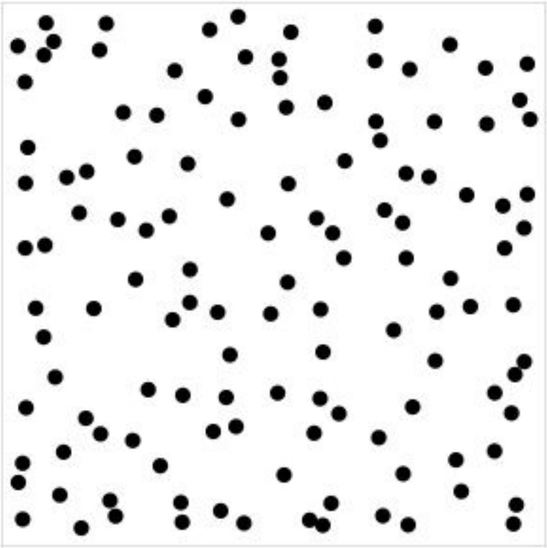
$$\mathcal{P}(\{i,j\} \subseteq Y) = \begin{vmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{vmatrix} = \mathcal{P}(\{i\} \subseteq Y)\mathcal{P}(\{j\} \subseteq Y) - K_{ij}^2$$

DPPs have ability to **diversify!**

DPP - diversification

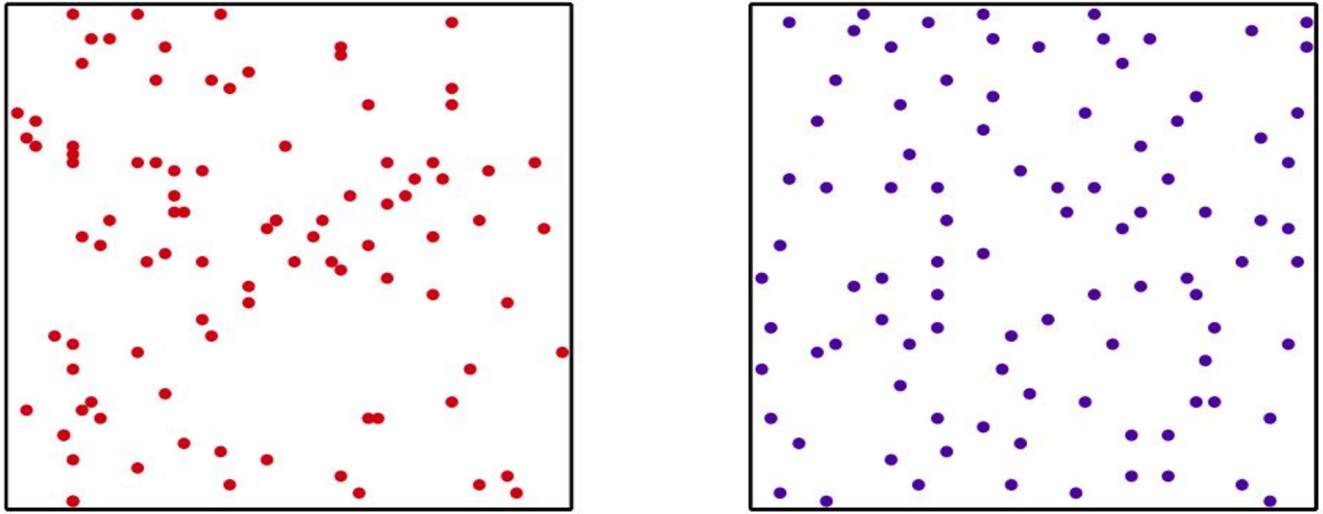


Independent Point Process



Determinantal Point Process

DPP - diversification



Independent

DPP

DPP properties

Restriction

If Y is distributed as a DPP with marginal kernel K and $A \subseteq \mathcal{Y}$ then $Y \cap A$ is DPP with marginal kernel K_A

Complement

If Y is distributed as DPP with marginal kernel K, $\mathcal{Y}$ - Y is distributed as a DPP with marginal kernel I-K.

Scaling

If $K = \gamma K'$ for some $0 \leq \gamma < 1$, then for all $A \subseteq \mathcal{Y}$ we have $\det(K_A) = \gamma^{|A|}\det(K'_A)$

L-ensemble

Let $\mathbf{L} = [\mathbf{L}_{ij}]$ be and $\mathbf{N} \times \mathbf{N}$ real, symmetric positive semidefinite matrix.

An L-ensemble is a point process that satisfies $\mathcal{P}_L(Y) \propto det(L_Y)$

Normalization constant: $\sum_{Y \subseteq \mathcal{Y}} \det(L_Y) = \det(L + I)$

For $Y \subseteq \mathcal{Y}$ we have $\mathcal{P}_L(Y) = \frac{det(L_Y)}{det(L+I)}$

Borodin & Rains, 2005

DPP and L-ensemble

- An L-ensemble is a DPP with marginal kernel K given by

$$K = L(L+I)^{-1} = I - (L+I)^{-1}$$

- Not all DPPs are L-ensembles!
  When any eigenvalue of K achieves the upper bound of 1, the DPP is not an L-ensemble!

$$L = K(I-K)^{-1}$$

DPP and L-ensemble

If

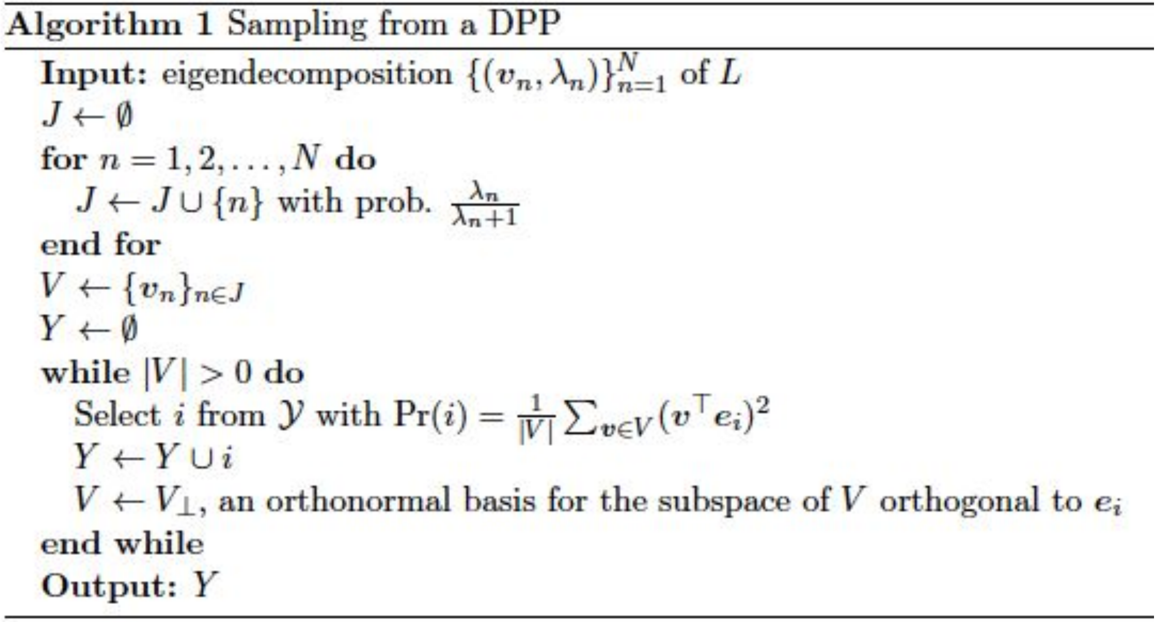
$$L = \sum_{n=1}^{N} \lambda_n \boldsymbol{v}_n \boldsymbol{v}_n^\top$$

is an eigendecomposition of $\mathbf{L}$ then

$$K = \sum_{n=1}^{N} \frac{\lambda_n}{\lambda_n + 1} \boldsymbol{v}_n \boldsymbol{v}_n^\top .$$

is eigendecomposition of $\mathbf{K}$.

Sampling Algorithm



Algorithm 1 Sampling from a DPP

```
**Input:** eigendecomposition $\{(\boldsymbol{v}_n, \lambda_n)\}_{n=1}^N$ of $L$
$J \leftarrow \emptyset$
**for** $n = 1, 2, \ldots, N$ **do**
  $J \leftarrow J \cup \{n\}$ with prob. $\frac{\lambda_n}{\lambda_n+1}$
**end for**
$V \leftarrow \{\boldsymbol{v}_n\}_{n \in J}$
$Y \leftarrow \emptyset$
**while** $|V| > 0$ **do**
  Select $i$ from $\mathcal{Y}$ with $\Pr(i) = \frac{1}{|V|}\sum_{\boldsymbol{v} \in V}(\boldsymbol{v}^\top \boldsymbol{e}_i)^2$
  $Y \leftarrow Y \cup i$
  $V \leftarrow V_\perp$, an orthonormal basis for the subspace of $V$ orthogonal to $\boldsymbol{e}_i$
**end while**
**Output:** $Y$
```

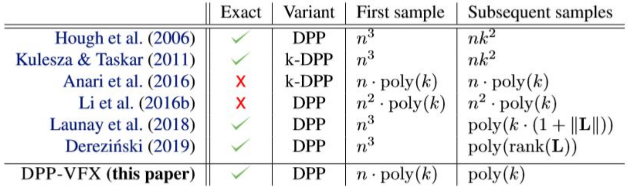
Sampling Algorithm Complexity

Algorithm complexity: $O(Nk^3)$ k = |V|

Gram-Schmidt orthonormalization: $O(Nk^2)$

Bottleneck: eigendecomposition of L with complexity $O(N^3)$

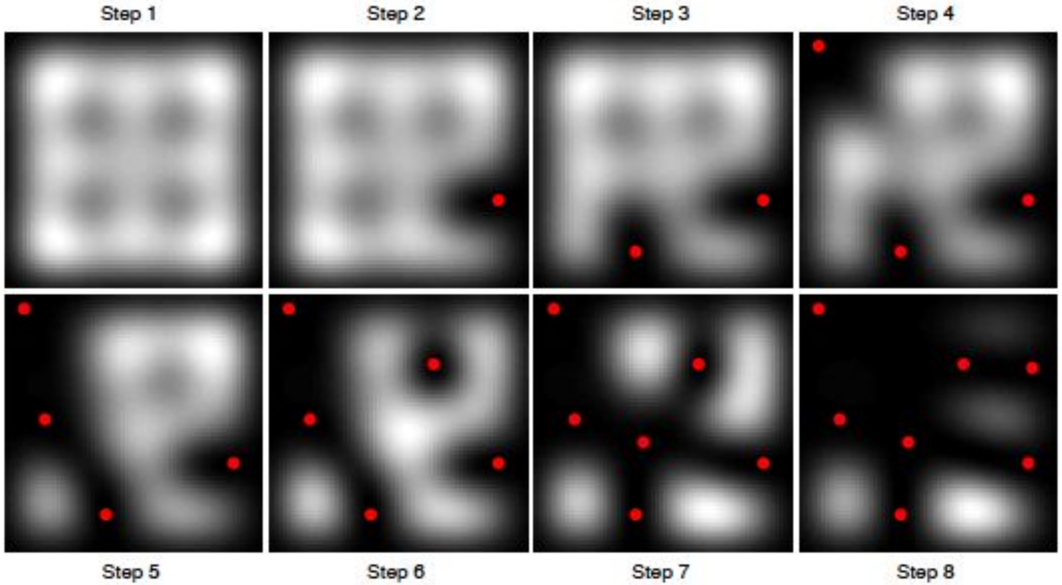
There exists exact and approximating DPP samplings with lower complexity.



	Exact	Variant	First sample	Subsequent samples
Hough et al. (2006)	✓	DPP	n^3	nk^2
Kulesza & Taskar (2011)	✓	k-DPP	n^3	nk^2
Anari et al. (2016)	✗	k-DPP	$n \cdot \mathrm{poly}(k)$	$n \cdot \mathrm{poly}(k)$
Li et al. (2016b)	✗	DPP	$n^2 \cdot \mathrm{poly}(k)$	$n^2 \cdot \mathrm{poly}(k)$
Launay et al. (2018)	✓	DPP	n^3	$\mathrm{poly}(k \cdot (1 + \lVert \mathbf{L} \rVert))$
Dereziński (2019)	✓	DPP	n^3	$\mathrm{poly}(\mathrm{rank}(\mathbf{L}))$
DPP-VFX **(this paper)**	✓	DPP	$n \cdot \mathrm{poly}(k)$	$\mathrm{poly}(k)$

Derezinski et. al, 2019

Sampling Algorithm - Visualization



NP Hardness

Ko et. al (1995): Finding the set Y that maximizes $\mathcal{P}_L(Y)$ is NP-hard.

$\mathcal{P}_L(Y)$ is a log-submodular function and can be optimized in polynomial time.

$$\log \mathcal{P}_L(Y \cup \{i\}) - \log \mathcal{P}_L(Y) \geq \log \mathcal{P}_L(Y' \cup \{i\}) - \log \mathcal{P}_L(Y') \quad \text{for} \quad Y \subseteq Y' \subseteq \mathcal{Y} - \{i\}$$

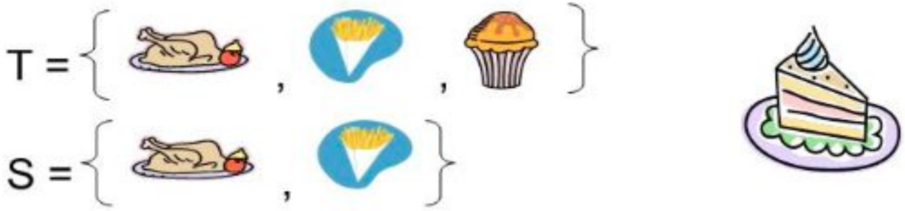
Submodularity

$X = \{x_1, ..., x_n\}$ a ground set of elements

$f : 2^X \to R_+$ a score function on subsets of X

Marginal value: $f_S(x) = f(S \cup \{x\}) - f(S)$

$$\boxed{S \subseteq T \Rightarrow f_S(\mathrm{x}) \geq f_T(\mathrm{x})}$$



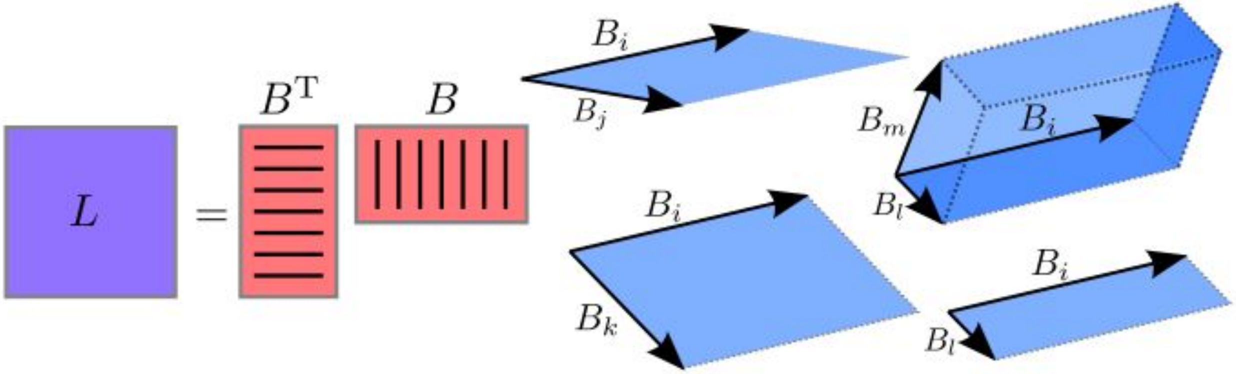
$$f_S(cake) \geq f_T(cake)$$

Question of Diversity

$\mathbf{L}$ can be decomposed as $\mathbf{L} = \mathbf{B}^\mathrm{T}\mathbf{B}$ for some $\mathbf{D}$ x $\mathbf{N}$, $\mathbf{D} \leq \mathbf{N}$

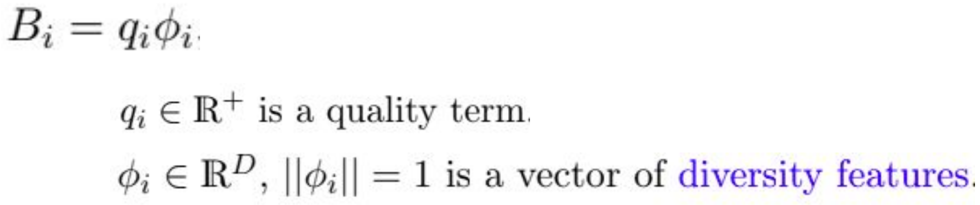
Let B_i be the i-th column of $\mathbf{B}$

$$\det(L_Y) = \mathrm{Vol}^2(\{B_i\}_{i \in Y})$$



Question of Quality

Columns of B are vectors that represent items in the set $\mathcal{Y}$.



$B_i = q_i \phi_i$.

$q_i \in \mathbb{R}^+$ is a quality term.

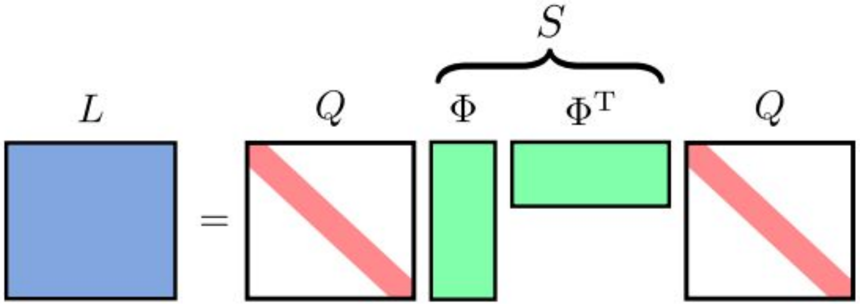
$\phi_i \in \mathbb{R}^D$, $||\phi_i|| = 1$ is a vector of diversity features.

Question of Quality

$$L_{ij} = q_i \phi_i^\top \phi_j q_j$$

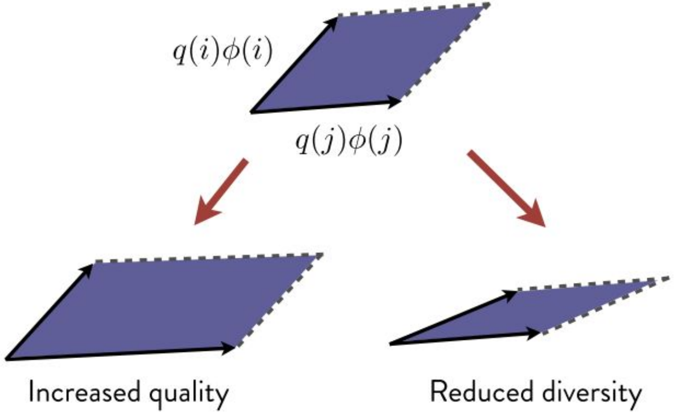
Similarity matrix $\mathbf{S}$:

$$S_{ij} \equiv \phi_i^\top \phi_j = \frac{L_{ij}}{\sqrt{L_{ii} L_{jj}}}$$



Question of Quality and Diversity

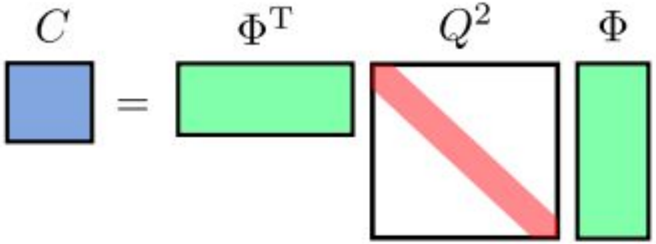
$$\mathcal{P}_L(Y) \propto \left(\prod_{i \in Y} q_i^2\right) \det(S_Y)$$



Decomposition for large sets

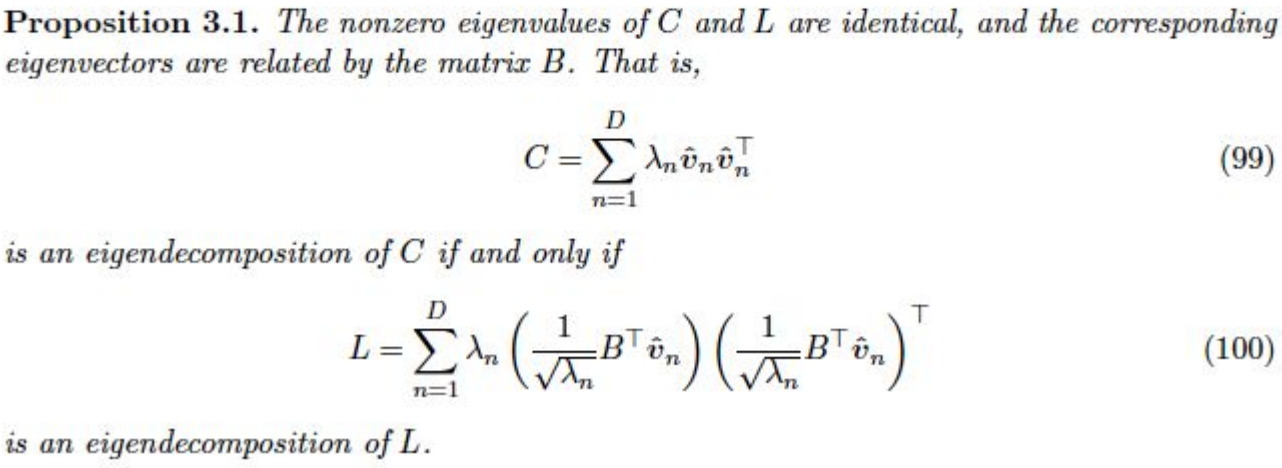
Dual representation:

C = BB^T is DxD real symmetric positive-semidefinite matrix



Decomposition for large sets

Dual representation:



Proposition 3.1. *The nonzero eigenvalues of C and L are identical, and the corresponding eigenvectors are related by the matrix B. That is,*

$$C = \sum_{n=1}^{D} \lambda_n \hat{v}_n \hat{v}_n^\top \tag{99}$$

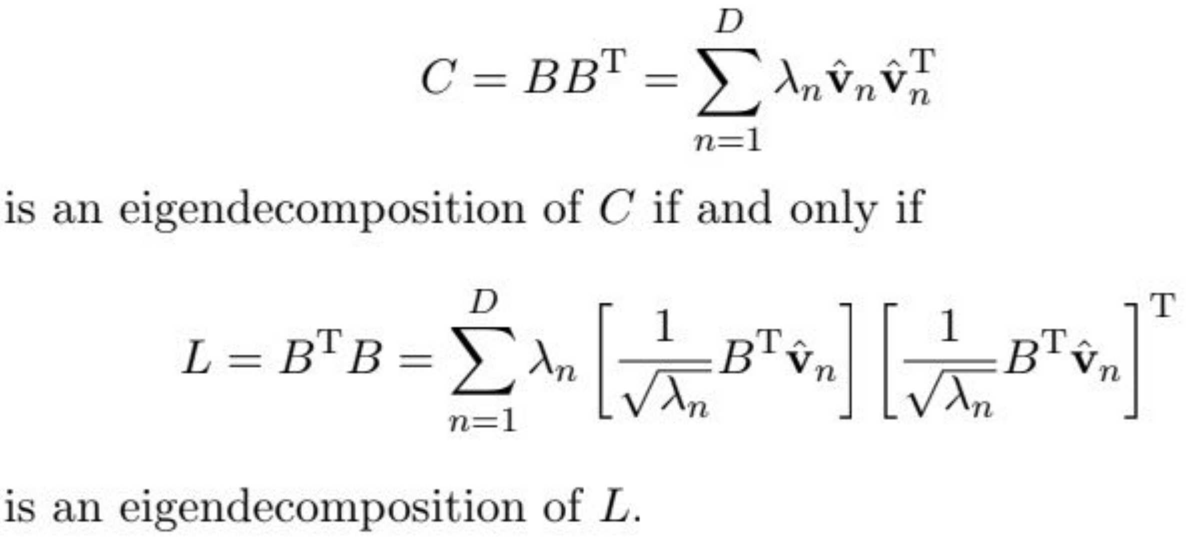
is an eigendecomposition of C if and only if

$$L = \sum_{n=1}^{D} \lambda_n \left(\frac{1}{\sqrt{\lambda_n}} B^\top \hat{v}_n \right) \left(\frac{1}{\sqrt{\lambda_n}} B^\top \hat{v}_n \right)^\top \tag{100}$$

is an eigendecomposition of L.

Decomposition for large sets

Dual representation:



$$C = BB^{\mathrm{T}} = \sum_{n=1}^{D} \lambda_n \hat{\mathbf{v}}_n \hat{\mathbf{v}}_n^{\mathrm{T}}$$

is an eigendecomposition of C if and only if

$$L = B^{\mathrm{T}}B = \sum_{n=1}^{D} \lambda_n \left[\frac{1}{\sqrt{\lambda_n}} B^{\mathrm{T}} \hat{\mathbf{v}}_n\right] \left[\frac{1}{\sqrt{\lambda_n}} B^{\mathrm{T}} \hat{\mathbf{v}}_n\right]^{\mathrm{T}}$$

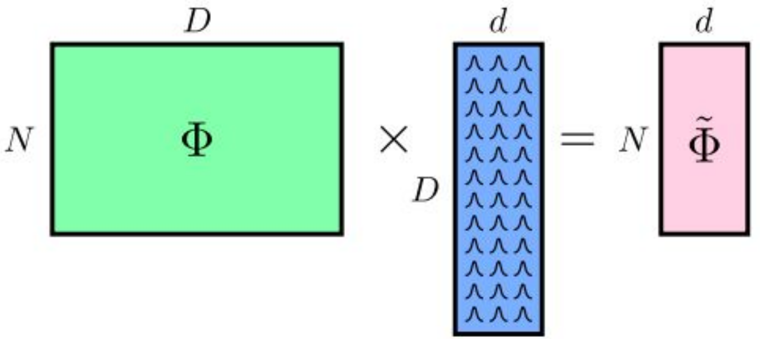
is an eigendecomposition of L.

Decomposition for large sets

Random projection:

Dimension D of diversity features can be large.
Idea: Project diversity vectors to a space of a low dimension d.

There are theoretical guarantees that random projection approximately preserves distances.



Learning

Training data

$$\{(X^{(t)}, Y^{(t)})\}, t = 1, 2, \ldots, T$$

$$(X^{(t)}, Y^{(t)}) \in \mathcal{X} \times 2^{\mathcal{Y}(X)}$$

$\mathcal{X}$ is an iput space

$\mathcal{Y}(X)$ is the associated ground set of input X

We assume that DPP kernel L(X; θ) is parametrized in terms of generic θ that reflect quality and/or diversity properties.

Learning

A conditional DPP $\mathcal{P}(Y|X)$ is a conditional probabilistic model which assigns a probability to every possible subset $Y \subseteq \mathcal{Y}(X)$. The model takes the form of an L-ensemble

$$\mathcal{P}_L(Y|X) \propto det(L_Y(X))$$

when L(X) is a positive semidefinite $|\mathcal{Y}(X)| \times |\mathcal{Y}(X)|$ matrix that depends on the input X

Learning

Goal:

choose parameter θ to maximize the conditional log-likelihood of the training data (so called Maximum Likelihood Estimation, MLE)

$$\mathcal{L}(\theta) = \log \prod_{t=1}^{T} \mathcal{P}_{\theta}(Y^{(t)} | X^{(t)})$$

with conditional probability of an output Y given input X under parameter θ

$$\mathcal{P}_{\theta}(Y|X) = \frac{\det(L_Y(X;\theta))}{\det(L(X;\theta) + I)}$$

Learning the Parameters of DPP Kernels

- It is conjecture that MLE is NP-hard to compute (Kulesza, 2012)
  - non-convex optimization
  - non-convexity holds even under various simplified assumptions on the form of L
  - approximating the mode of size k of a DPP to within a c^k (c>1) factor is known to be NP-hard

- Special cases of quality or similarity functions:
  - Gaussian similarity with uniform quality
  - Gaussian similarity with Gaussian quality
  - Polynomial similarity with uniform quality

- Nelder-Mead simplex algorithm:

  does not require explicit knowledge of derivates of a log-likelihood function, there are no theoretical guarantees about convergence to a stationary point.

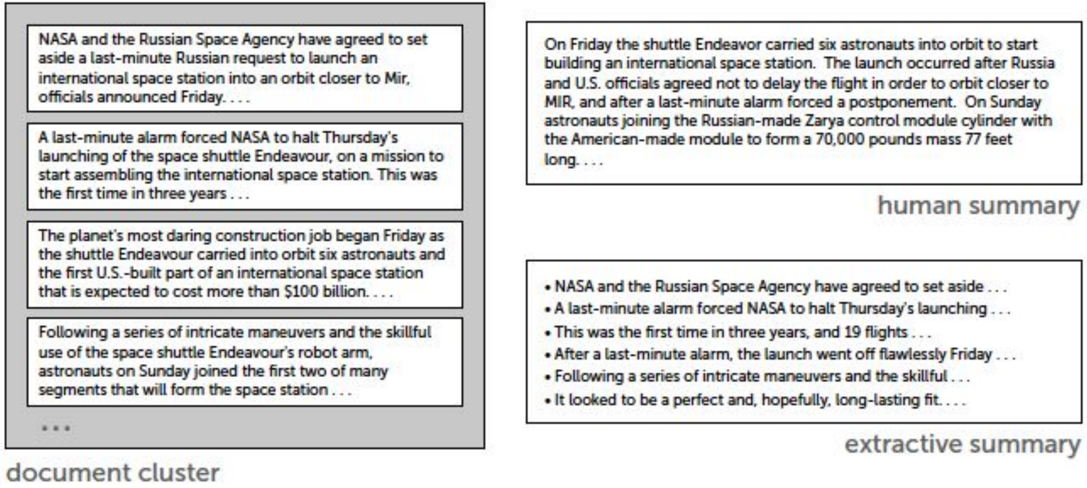
Learning the parameters of DPP Kernels

Heuristics

- Expectation-Maximization (Gillenwater et al., 2014)
- MCMC (Affandi et al., 2014)
- Fixed point algorithms (Mariet & Sra, 2015)

Extractive Document Summarization

Goal: Learn a DPP to model good summaries Y for a given input X



NASA and the Russian Space Agency have agreed to set aside a last-minute Russian request to launch an international space station into an orbit closer to Mir, officials announced Friday. . . .

A last-minute alarm forced NASA to halt Thursday's launching of the space shuttle Endeavour, on a mission to start assembling the international space station. This was the first time in three years . . .

The planet's most daring construction job began Friday as the shuttle Endeavour carried into orbit six astronauts and the first U.S.-built part of an international space station that is expected to cost more than $100 billion. . . .

Following a series of intricate maneuvers and the skillful use of the space shuttle Endeavour's robot arm, astronauts on Sunday joined the first two of many segments that will form the space station . . .

. . .

document cluster

On Friday the shuttle Endeavor carried six astronauts into orbit to start building an international space station. The launch occurred after Russia and U.S. officials agreed not to delay the flight in order to orbit closer to MIR, and after a last-minute alarm forced a postponement. On Sunday astronauts joining the Russian-made Zarya control module cylinder with the American-made module to form a 70,000 pounds mass 77 feet long. . . .

human summary

- NASA and the Russian Space Agency have agreed to set aside . . .
- A last-minute alarm forced NASA to halt Thursday's launching . . .
- This was the first time in three years, and 19 flights . . .
- After a last-minute alarm, the launch went off flawlessly Friday . . .
- Following a series of intricate maneuvers and the skillful . . .
- It looked to be a perfect and, hopefully, long-lasting fit. . . .

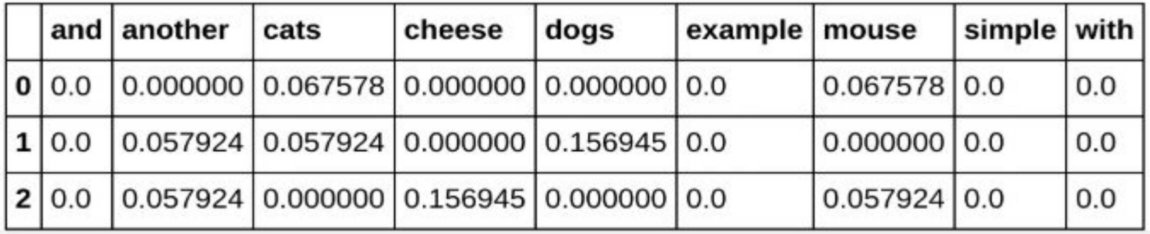
extractive summary

Kuleza & Taskar, 2012

Extractive Document Summarization

Similarity features

sentences are represented as normalized **tf-idf** vectors



	and	another	cats	cheese	dogs	example	mouse	simple	with
0	0.0	0.000000	0.067578	0.000000	0.000000	0.0	0.067578	0.0	0.0
1	0.0	0.057924	0.057924	0.000000	0.156945	0.0	0.000000	0.0	0.0
2	0.0	0.057924	0.000000	0.156945	0.000000	0.0	0.057924	0.0	0.0

TF-IDF bag of words

Similarity function

similarity among sentences

$$S_{ij} = \frac{\sum_w \mathrm{tf}_i(w)\mathrm{tf}_j(w)\mathrm{idf}^2(w)}{\sqrt{\sum_w \mathrm{tf}_i^2(w)\mathrm{idf}^2(w)}\sqrt{\sum_w \mathrm{tf}_j^2(w)\mathrm{idf}^2(w)}} \in [0,1]$$

Extractive Document Summarization

Quality scores

$$q_i(X;\theta) = \exp\left(\frac{1}{2}\theta^\top \boldsymbol{f}_i(X)\right)$$

$\boldsymbol{f}_i(X) \in \mathbb{R}^m$ feature vector for the sentence i

$\boldsymbol{\theta} \in \boldsymbol{R}^m$ parameter vector

Quality features

length, position of the sentence in its original document, mean cluster similarity, personal pronouns, LexRank score, …

Extractive Document Summarization

Quality and similarity combined in the form of conditional DPP probability:

$$\mathcal{P}_\theta(Y|X) = \frac{\prod_{i\in Y}\left[\exp\left(\theta^\top \boldsymbol{f}_i(X)\right)\right]\det(S_Y(X))}{\sum_{Y'\subseteq\mathcal{Y}(X)}\prod_{i\in Y'}\left[\exp\left(\theta^\top \boldsymbol{f}_i(X)\right)\right]\det(S_{Y'}(X))}.$$

Holds $\mathcal{L}(\theta) = \log\prod_{t=1}^{T}\mathcal{P}_\theta(Y^{(t)}|X^{(t)})$ is concave in θ

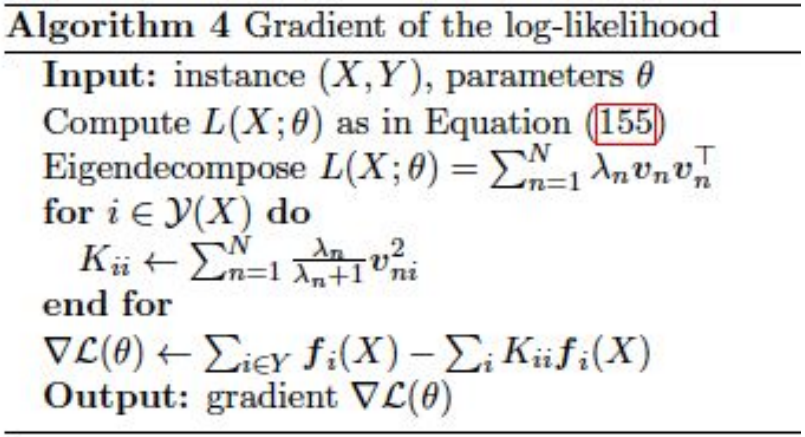
Gradient $\nabla\mathcal{L}(\theta)$

$$= \sum_{i\in Y}\boldsymbol{f}_i(X) - \sum_{Y'\subseteq\mathcal{Y}(X)}\mathcal{P}_\theta(Y'|X)\sum_{i\in Y'}\boldsymbol{f}_i(X)$$

$$\sum_{Y'\subseteq\mathcal{Y}(X)}\mathcal{P}_\theta(Y'|X)\sum_{i\in Y'}\boldsymbol{f}_i(X) = \sum_{i}\boldsymbol{f}_i(X)\sum_{Y'\supseteq\{i\}}\mathcal{P}_\theta(Y'|X)$$

Extractive Document Summarization

Learning quality parameters by gradient descent



```
Algorithm 4 Gradient of the log-likelihood
Input: instance $(X, Y)$, parameters $\theta$
Compute $L(X;\theta)$ as in Equation (155)
Eigendecompose $L(X;\theta) = \sum_{n=1}^{N} \lambda_n \boldsymbol{v}_n \boldsymbol{v}_n^\top$
for $i \in \mathcal{Y}(X)$ do
    $K_{ii} \leftarrow \sum_{n=1}^{N} \frac{\lambda_n}{\lambda_n+1} \boldsymbol{v}_{ni}^2$
end for
$\nabla\mathcal{L}(\theta) \leftarrow \sum_{i \in Y} \boldsymbol{f}_i(X) - \sum_i K_{ii} \boldsymbol{f}_i(X)$
Output: gradient $\nabla\mathcal{L}(\theta)$
```

Extractive Document Summarization

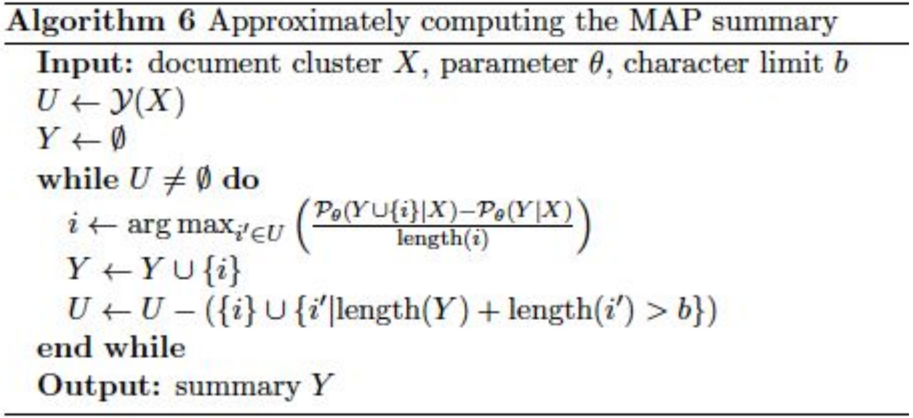
Inference: for a given X and learned parameter θ find Y with at most b characters

Maximum a posteriori:

$$Y^{\mathrm{MAP}} = \arg\max_{Y} \; \mathcal{P}_\theta(Y|X)$$

$$\text{s.t.} \; \sum_{i \in Y} \mathrm{length}(i) \leq b \,,$$

Y is submodular and we can approximate it through simple greedy algorithm



Algorithm 6 Approximately computing the MAP summary

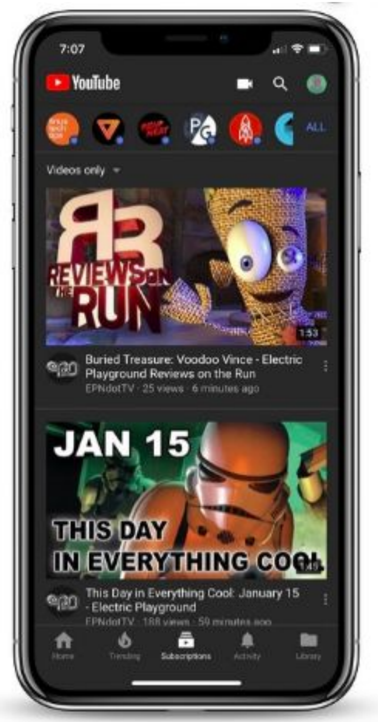
```
Input: document cluster $X$, parameter $\theta$, character limit $b$
$U \leftarrow \mathcal{Y}(X)$
$Y \leftarrow \emptyset$
while $U \neq \emptyset$ do
    $i \leftarrow \arg\max_{i' \in U} \left( \frac{\mathcal{P}_\theta(Y \cup \{i\}|X) - \mathcal{P}_\theta(Y|X)}{\mathrm{length}(i)} \right)$
    $Y \leftarrow Y \cup \{i\}$
    $U \leftarrow U - (\{i\} \cup \{i' | \mathrm{length}(Y) + \mathrm{length}(i') > b\})$
end while
Output: summary $Y$
```

Diversified Recommendation

- Content is presented in the form of a feed:
  an ordered list of items through the user browses

- The goodness of the recommendation is measured via utility function

- Goal:
  select and order a set of k items such that the utility of the set is maximized



Buried Treasure: Voodoo Vince - Electric Playground Reviews on the Run
EPNdotTV · 25 views · 6 minutes ago

This Day in Everything Cool: January 15 - Electric Playground
EPNdotTV · 188 views · 59 minutes ago

Gillenwater et al, 2018

k-DPP

A k-DPP on a discrete set is distribution over all subsets with cardinality k.

$$\mathcal{P}^k(Y) = \frac{\det(L_Y)}{\sum_{|Y'|=k} \det(L_{Y'})}$$

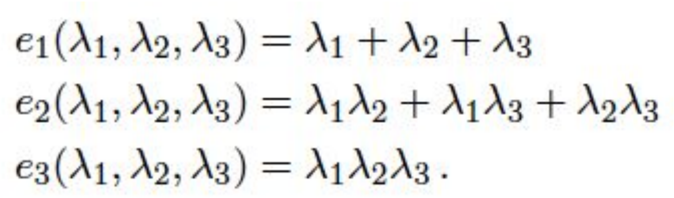
Normalization constant is $Z_k = \sum_{|Y'|=k} \det(L_{Y'}) = e_k(\lambda_1, \lambda_2, \ldots, \lambda_N)$ where e_k is k-th

elementary symmetric polynomial $e_k(\lambda_1, \lambda_2, \ldots, \lambda_N) = \sum_{\substack{J \subseteq \{1,2,\ldots,N\} \\ |J|=k}} \prod_{n \in J} \lambda_n$

over $\lambda_1, \lambda_2 \ldots, \lambda_N$ eigenvalues of the matrix L.

k-DPP

For N = 3 elementary symmetric polynomials are:

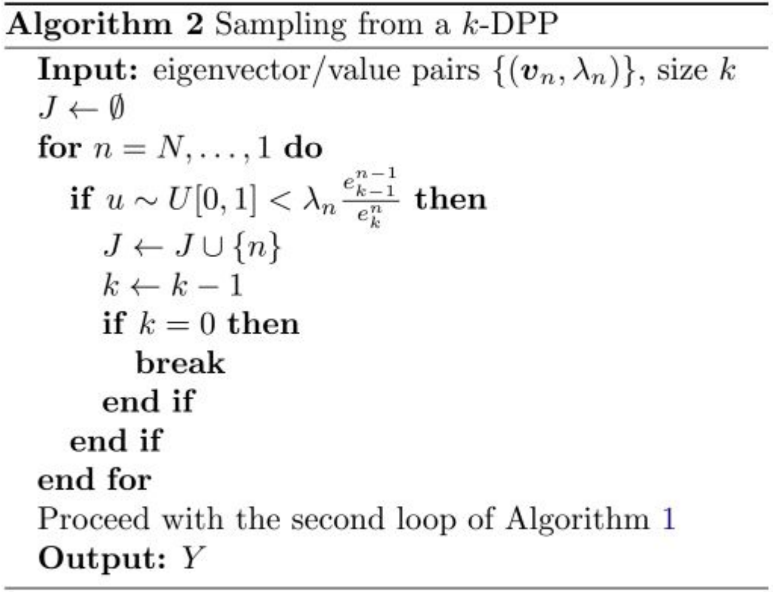


$$
\begin{aligned}
e_1(\lambda_1, \lambda_2, \lambda_3) &= \lambda_1 + \lambda_2 + \lambda_3 \\
e_2(\lambda_1, \lambda_2, \lambda_3) &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 \\
e_3(\lambda_1, \lambda_2, \lambda_3) &= \lambda_1\lambda_2\lambda_3 \, .
\end{aligned}
$$

There is an effective recursive algorithm for elementary symmetric polynomials calculations.

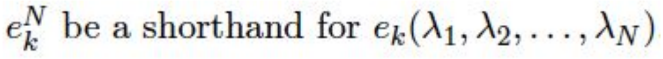
k-DPP

Sampling:



Algorithm 2 Sampling from a k-DPP

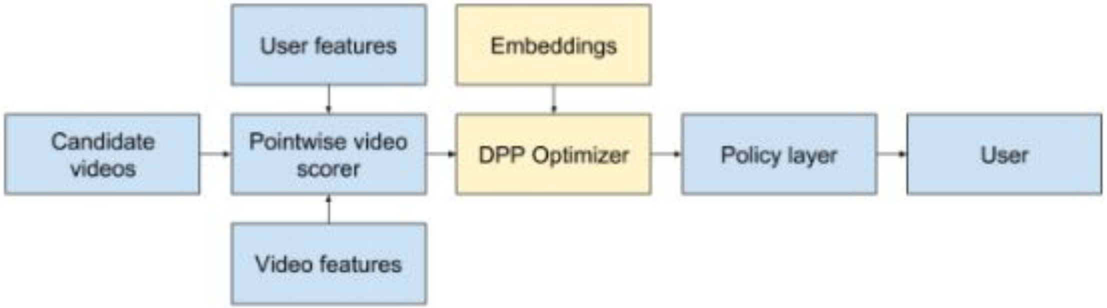
```
Input: eigenvector/value pairs ${(\boldsymbol{v}_n, \lambda_n)}$, size $k$
$J \leftarrow \emptyset$
for $n = N, \ldots, 1$ do
  if $u \sim U[0,1] < \lambda_n \frac{e_{k-1}^{n-1}}{e_k^n}$ then
    $J \leftarrow J \cup \{n\}$
    $k \leftarrow k - 1$
    if $k = 0$ then
      break
    end if
  end if
end for
Proceed with the second loop of Algorithm 1
Output: $Y$
```



e_k^N be a shorthand for $e_k(\lambda_1, \lambda_2, \ldots, \lambda_N)$

Diversified Recommendation

DPP inputs: personalized quality scores and pointwise item distances



Diversified Recommendation

Observed interaction of user i with the feed list of length N is given as a binary vector: y_u= [0, 1, 0, 1, 1, …, 0]

Goal:

maximize the total number of interactions

$$G' = \sum_{u \sim \text{Users}} \sum_{i \sim \text{Items}} y_{ui} .$$

Slight modification in order to train models from records of previous interactions: maximize the cumulative gain by reranking the feed items

$$G = \sum_{u \sim \text{Users}} \sum_{i \sim \text{Items}} \frac{y_{ui}}{j}$$

j is the new rank that the model assigns to an item.

Diversified Recommendation

The interaction y_u = [0, 1, 0, 1, 1, ….0] can be written as Y = {2, 4, 5}

Assumption:

Y represents a drawn from the probability distribution defined by a user-specific DPP.

Diversified Recommendation

q_i - personalized quality score
D_{ij} - Jaccard distance built on video descriptions

$$L_{ii} = q_i^2$$
$$L_{ij} = \alpha q_i q_j \exp\left(-\frac{D_{ij}}{2\sigma^2}\right) \text{ , for } i \neq j$$

α∈ [0, 1) and σ are learning parameters

Run grid search to find the values of the parameters that maximize the cumulative gain.

Diversified Recommendation

f - parametrized quality function

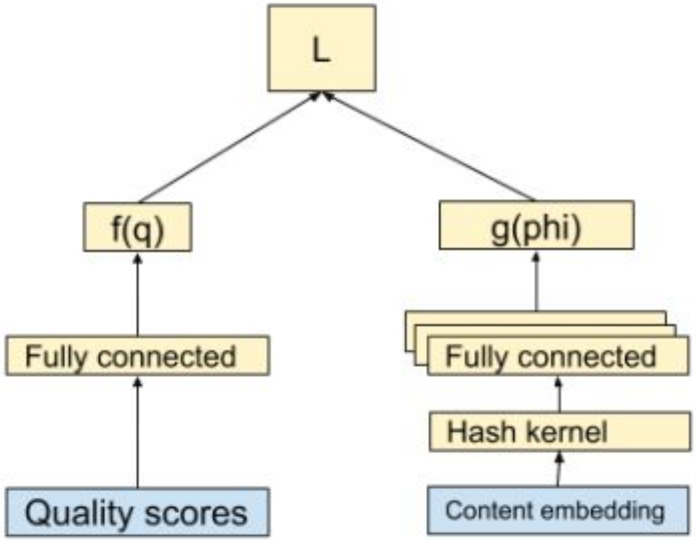
g - parametrized content re-embedding function

□ - fixed regularization parameter

w - the parameters of L

$$L_{ij} = f(\boldsymbol{q}_i)\boldsymbol{g}(\phi_i)^T \boldsymbol{g}(\phi_j) f(\boldsymbol{q}_j) + \delta \mathbb{1}_{i=j}$$

$$\begin{aligned}\mathrm{LogLike}(\boldsymbol{w}) &= \sum_{j=1}^{M} \log(\mathcal{P}_{L(\boldsymbol{w})}(Y_j)) \\ &= \sum_{j=1}^{M} \left[\log(\det(L(\boldsymbol{w})_{Y_j})) - \log(\det(L(\boldsymbol{w}) + I))\right]\end{aligned}$$



Diversified Recommendation

Inference

greedy algorithm for submodular maximization for size-k window

$$Y = \emptyset$$

Runs k iterations adding one video to Y on each iteration: $\max_{v \in \text{remaining videos}} \det(L_{Y \cup v})$

The greedy algorithm gives the natural order for videos in size-k window.

Than the algorithm is repeated for the unused N-k videos.

DPP classes

- Discrete DPP
- k-DPP

- Continual DPP
- Structured DPP

  translation, bioinformatics, ...
- Sequential DPP

  video summarization, …

Code

Matlab:

https://www.alexkulesza.com/code/dpp.tgz

Python:

https://github.com/guilgautier/DPPy

Thank you!