k-SDPP: Fixed-Size Video Summarization via Sequential Determinantal Point Processes

Jiping Zheng[1,2] and **Ganfeng Lu**[1]
[1]College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics
[2]Collaborative Innovation Center of Novel Software Technology and Industrialization
{jzh, luganf}@nuaa.edu.cn

Abstract

With the explosive growth of video data, video summarization which converts long-time videos to key frame sequences has become an important task in information retrieval and machine learning. Determinantal point processes (DPPs) which are elegant probabilistic models have been successfully applied to video summarization. However, existing DPP-based video summarization methods suffer from poor efficiency of outputting a specified size summary or neglecting inherent sequential nature of videos. In this paper, we propose a new model in the DPP lineage named k-SDPP in vein of sequential determinantal point processes but with fixed user specified size k. Our k-SDPP partitions sampled frames of a video into segments where each segment is with constant number of video frames. Moreover, an efficient branch and bound method (BB) considering sequential nature of the frames is provided to optimally select k frames delegating the summary from the divided segments. Experimental results show that our proposed BB method outperforms not only k-DPP and sequential DPP (seqDPP) but also the partition and Markovian assumption based methods.

1 Introduction

With the popularity of camera devices, a large number of videos are captured every day. Especially after entering the 5G era, each of us equipped with a mobile phone is a potential photographer who can produce video data anytime and anywhere. In 2019, it was estimated that there are 38 EB video data produced by mobile applications per month and will increase to 160 EB per month in 2025 [Eri, 2019]. It is a huge problem to store and manage these video data and no individual or company can afford to store all the video data. Meanwhile, browsing these huge video data is time-consuming. Fortunately, an efficient way to handle the explosive data is automatical summarization of the videos which converts long videos to key frame sequences.

In the literature of video summarization [Vivekraj *et al.*, 2019], models of the DPP lineage have shown great success to select informative subsets for users [Kulesza and Taskar, 2012]. Originating from quantum physics and random matrix theories, DPP is a powerful tool to balance two important properties of video summarization, importance and diversity. Compared to traditional independent sampling methods, DPP has more advantages in terms of diversity [Hough *et al.*, 2006] and has been applied to many data summarization applications, such as image search [Kulesza and Taskar, 2011a], document summarization [Kulesza and Taskar, 2011b], recommendation systems [Zhou *et al.*, 2010], sensor placement [Krause *et al.*, 2008], etc. Even DPP-based models work well in various scenarios for video summarization, they have some inherent drawbacks. For vanilla DPP-based [Kulesza and Taskar, 2012] and some other video summarization algorithms [Kulesza and Taskar, 2011a; Zhang *et al.*, 2016; Celis *et al.*, 2018], they usually ignore the sequences between the video frames. For example, if there is a video about a football match, there may be two frames about players who have scored in the same position at two different moments. Then the algorithms according to these DPPs will not keep these two frames simultaneously in the summary for violating the DPP's standard of diversity. But these two frames are very valuable frames both needed to be included in the summary. To address this issue, [Gong *et al.*, 2014] proposes seqDPP which fully considers the sequence correlation between video frames. However, seqDPPs [Gong *et al.*, 2014; Sharghi *et al.*, 2016] cannot control the outputting size for each summarizing task. But on many occasions when summarizing a video, we need to predetermine the size of the result (or no more than a certain value).

To output a summary with fixed size, efforts include k-DPPs [Kulesza and Taskar, 2011a], Partition-DPPs [Zhang *et al.*, 2016] and Generalized DPPs (GDPPs) [Sharghi *et al.*, 2018]. Different from k-DPPs which directly model sets of fixed size and further normalize and sample from these sets for the summary, Partition-DPPs divide the frames of a video into partitions and extract frames with certain size from each partition, then total length of the extracted frames equals to the user specified size. A GDPP can be considered as a mixture of k-DPPs, thus k-DPP including DPP is a special case of GDPP. However, large size of each partition (Partition-DPPs) or component (GDPPs) makes probabilistic inference expensive and leads to low efficiency to get the final summary. Moreover, these DPP-based models are based on the characteristic of diversity which do not take the users' supervisions into account and neglect the sequential nature of videos.

In this paper, we propose k-SDPP, a new probabilistic model in the DPP lineage. k-SDPP outputs a fixed-size, *e.g.,* k video frames via sequential determinantal point processes. Unlike k-DPPs, Partition-DPPs and GDPPs, our k-SDPP divides a long video sequence into disjointed consecutive short segments. Since final k frames are selected from these segments, we propose branch and bound method (BB) to optimally allocate the k frames to the disjointed consecutive segments. We show that seqDPP is a special case of our BB method which greedily outputs a summary. Besides seqDPP, we also compare our BB method with other typical methods including k-DPP, partition and Markovian assumption based solutions.

The rest of this paper is organized as follows. Section 2 surveys most related work. Section 3 introduces related DPP models. Our proposed DPP model, k-SDPP along with the branch and bound method is detailed in Section 4. Section 5 experimentally evaluates our method, and Section 6 concludes this paper.

2 Related Work

There are plenty of methods for video summarization and applying blooming DPP-based approaches to summarize videos is an important aspect for video summarization [Vivekraj *et al.*, 2019]. Two categories of video summarization approaches are emphasized in the literature, namely, property-based and DPP-based. For property-based approaches, a variety of properties, such as representativeness, diversity, importance, interestingness, storyness etc. are exploited to summarize a video. For DPP-based approaches, various DPP models are utilized to learn the criteria from humman-annotated summaries automatically in a supervised manner. We only survey most related DPP-based approaches due to space limitation.

To deal with the inherent sequences of videos, Markov DPP [Affandi *et al.*, 2012] defines conditional probabilities depending on the immediate past segment and helps to explore the frames of interest to users by providing sequential and diverse results. To avoid the disturbance of the frames far from each other of Markov DPP, [Gong *et al.*, 2014] proposes seqDPP which can fully consider the order correlation between video frames and faithfully represent the relationship between the video data. In addition, they also propose ways to teach the system to learn from human-created summaries, so that the final summary is closer to human-created results. SeqDPP calculates conditional probabilities depending only on the immediate past segment which is the same as Markov DPP [Affandi *et al.*, 2012]. The main difference is that seqDPP selects diverse sets from the present time instead of the whole video sequence of Markov DPP. Thus Markov DPP fails to select video frames following inherent temporal order and cannot model the sequential nature faithfully. Further, [Sharghi *et al.*, 2016] extends seqDPP to SH-DPP (Sequential and Hierarchical DPP) which aims to advance the user-oriented video summarization by modeling user queries in the summarization process. SH-DPP has two layers of random variables. The first layer is used to select the frames relevant to the user queries while the second layer models the importance of the frames in the context of the videos. SH-DPP is efficient in modeling extremely lengthy videos and capable of producing summaries on the fly. [Celis *et al.*, 2018] presents a framework to integrate fairness constraints into determinantal distributions and provides efficient algorithms to sample from these distributions for video summarization. Moreover, [Mirzasoleiman *et al.*, 2018] finds that the DPP probability is a log-submodular function and they use submodular techniques to seek a near-optimal function. They develop single pass streaming algorithm *streaming local search* by above three orders of magnitudes to the state-of-art algorithms with approximation theoretical guarantees for streaming video data.

To tackle the problem that outputting fixed-size summary for DPP-based models, [Kulesza and Taskar, 2011a; Li *et al.*, 2016] directly model sets of fixed size and further normalize and sample from these sets for summary. [Kathuria and Deshpande, 2016] proposes Partition-DPPs which divides a long video sequence into p partitions where each partition can be represented as a k_i-DPP ($i = 1, 2, ..., p$ and $\sum_{i=1}^{p} k_i = k$). The partition process makes probabilistic inference easy. However, Partition-DPPs do not consider the sequences of the videos. GDPP [Sharghi *et al.*, 2018] which is also in vein of seqDPP allows users to control the lengths of system-generated video summaries where an arbitrary prior distribution can be imposed over the sizes of the subsets of video frames. They show that vanillia DPPs and k-DPP are special instances of GDPP and DPPs in seqDPP can be substituted by GDPPs. But GDPPs also suffer from low efficiency of probabilistic inference which is performed in each component *i.e.,* a DPP or k-DPP of GDPP.

3 Preparation

In this section, we introduce DPP, k-DPP and seqDPP for video summarization and further detail our k-SDPP along with the BB method in the next section.

3.1 Determinantal Point Process

Determinantal point process (DPP) is a model which was first used to characterize Pauli exclusion where two identical particles cannot occupy the same quantum state simultaneously [Macchi and Odile, 1975]. The identity of exclusion makes DPP a tool for modeling diversity so that DPP is appropriate for video summarization. In this section, we first give basic concepts about DPP.

Let $G = \{1, 2, ..., N\}$ be a ground set of N frames of a video. Note that a video frame is usually represented by its feature vector, *e.g.*, Fisher vector [Perronnin and Dance, 2007] which can well cover the content of a frame. A determinantal point process (DPP) defines a discrete probability distribution over all 2^N subsets of G. Let X be the random variable of selecting a subset from the collection of these 2^N subsets and X is distributed according to

$$P(X = \boldsymbol{x}) = \frac{det(\boldsymbol{L}_{\boldsymbol{x}})}{det(\boldsymbol{L} + \boldsymbol{I})} \tag{1}$$

And we also have

$$det(\boldsymbol{L}+\boldsymbol{I}) = \sum_{X\subseteq G} det(\boldsymbol{L}_X) \tag{2}$$

where the kernel matrix $\boldsymbol{L} \in (\mathbb{R}_+^{N\times N})$ is the DPP's parameter constrained to be positive semidefinite. The rows and columns of $\boldsymbol{L}$ are indexed by the frames in G while $\boldsymbol{L_x}$ is the sub-matrix of the $\boldsymbol{L}$ whose rows and columns are selected according to the frames in subset $\boldsymbol{x}$. $\boldsymbol{I}$ is the $N \times N$ identity matrix. The $det(\cdot)$ is a determinant function which makes the model have the identity of pairwise repulsion. For example, if a subset X with only two elements i and j is selected, we have

$$P(X=\{i,j\}) = L_{ii}L_{jj} - L_{ij}^2 \tag{3}$$

From Equation 3, we find that the more i and j are similar to each other, the less probability they will be in the same subset. In extreme case $i = j$, we have $P = 0$ (because $L_{ii} = L_{jj} = L_{ij}$). Each value in matrix $\boldsymbol{L}$ is to measure the probability that the corresponding two elements could be in the same subset. Also, the most diverse subset of G will have the largest value of $det(\cdot)$, and it has the highest probability as shown in Equation 4.

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\subseteq \boldsymbol{G}}{\arg\max}\, P(X=\boldsymbol{x}) = \underset{\boldsymbol{x}\subseteq \boldsymbol{G}}{\arg\max}\, det(\boldsymbol{L_x}) \tag{4}$$

To find the best subset from all G's 2^N subsets is obviously an NP-hard problem, an efficient way is the sampling method. Eigen-decomposition on matrix $\boldsymbol{L}$ can be performed first, then the size of the summary is randomly determined by sampling based on its eigenvalue. Finally, according to the size of the summary, the frames are selected from the ground set according to the principle of diversity [Kulesza and Taskar, 2012].

For video summarization, we need to select a subset of all frames in a video over a DPP. And now we have some training frames in the form of videos and the ground-truth summaries. The most important thing for us is to find the underlying parameter matrix $\boldsymbol{L}$ which is exploited to generate summaries for this video. Because the video we need to summarize may have frames that have not been seen in the training samples, $\boldsymbol{L}$ needs to be reparameterized. [Kulesza and Taskar, 2011b] proposes quality/diversity decomposition to reparameterize $\boldsymbol{L}$.

$$L_{ij} = q_i\phi_i^T\phi_j q_j,\ q_i = exp(\frac{1}{2}\theta^T f_i) \tag{5}$$

where ϕ_i is the normalized image feature vector of frame i. The image feature vector *i.e.,* Fisher vector, can fully represent the importance of a frame. f_i is the quality feature vector (*e.g.,* contextual feature, saliency map) which encodes other information of frame i. θ is the parameter which is optimized with maximum likelihood estimation, so the target subsets will have the highest probabilities.

3.2 k-DPP

Vanilla DPPs cannot guarantee to output a summary of fixed size [Kulesza and Taskar, 2012]. To control the size of output summary, k-DPP [Kulesza and Taskar, 2011a; Li *et al.*, 2016] can be exploited. k-DPP is similar to DPP where a k-DPP defines a discrete probability distribution over all subsets $X \in G$ with cardinality constraint k. It can be obtained simply by taking the cardinality constraint k into a standard DPP, from Equation 2, we have:

$$P_{\boldsymbol{L}}^k(X) = \frac{det(\boldsymbol{L}_X)}{\sum_{|X'|=k} det(\boldsymbol{L}_{X'})} \tag{6}$$

If each eigenvalue of a DPP's marginal kernel is in $\{0,1\}$, we call the DPP an elementary DPP. We use P^V to denote an elementary DPP where V is a set of orthonormal vectors. For k-DPP, it needs to find the best subset in all G's C_N^k k-size subsets. It also takes exponential time to find the determinant with the highest probability.

Note that [Sharghi *et al.*, 2018] proposes generalized DPP (GDPP) where they rewrite DPP (Equation 1) as follows.

$$P(Y;\boldsymbol{L}) = \sum_{k=0}^{n}\pi_i \sum_{S\in 2^G, |S|=k} P(Y;S)\prod_{i\in S}\lambda_i \tag{7}$$

where λ is eigenvalue of matrix $\boldsymbol{L}$. GDPP as shown in Equation 7 entails DPP when π is a uniform distribution and k-DPP when π is a Dirac delta distribution.

3.3 Sequential DPP

It is no doubt that vanilla DPPs and k-DPPs are powerful tools for modeling diverse subset selection, but they both ignore the sequences among the video frames. Sequential DPP (seqDPP) is provided to capture the sequential nature of video frames. SeqDPP first partitions a video into m disjointed consecutive short segments and $\bigcup_{i=1}^m G_i = G$. At each segment i, let X_i be the variable of the subset selected from the segment i. Then seqDPP imposes a DPP over two neighboring segments where the current segment's ground set is $V_i = \boldsymbol{x}_{i-1} \cup G_i$. $\boldsymbol{x}_{i-1}$ is the subset selected from the segment $i-1$. The conditional distribution of X_i is

$$P(X_i = \boldsymbol{x}_i | X_{i-1} = \boldsymbol{x}_{i-1}) = \frac{det(\boldsymbol{L}_{\boldsymbol{x}_{i-1}\cup \boldsymbol{x}_i})}{det(\boldsymbol{L}_{\boldsymbol{x}_{i-1}\cup G_i} + \boldsymbol{I}_i)} \tag{8}$$

where $\boldsymbol{L}_{\boldsymbol{x}_{i-1}\cup G_i}$ is the sub-matrix of $\boldsymbol{L}$ whose rows and columns are selected according to the elements in $\boldsymbol{x}_{i-1}\cup G_i$. $\boldsymbol{I}_i$ is a diagonal matrix whose size is the same with the $\boldsymbol{L}_{\boldsymbol{x}_{i-1}\cup G_i}$. However, the elements corresponding to X_{i-1} are zeros and the elements corresponding to G_i are ones. Equation 8 means $\boldsymbol{x}_i$ should be as diverse as possible from the previous subset $\boldsymbol{x}_{i-1}$. In other words, the subset is not constrained by the subsets selected in the distant past. In this way, the sequential nature can be taken into consideration that the two goals in the football match example will not be missed.

$$\begin{aligned} &P(X_1 = \boldsymbol{x}_1, X_2 = \boldsymbol{x}_2, ..., X_m = \boldsymbol{x}_m) \\ &= P(X_1 = \boldsymbol{x}_1)\prod_{i=2}^{m} P(X_i = \boldsymbol{x}_i | X_{i-1} = \boldsymbol{x}_{i-1}) \end{aligned} \tag{9}$$

The sequential DPP for modeling sequential video data can also be drawn as a Bayesian network [Gong *et al.*, 2014] as

Equation 9 shows. Different from Markov DPP [Affandi *et al.*, 2012], each segment G_i in seqDPP is quite small, thus it is reasonable to do exhaustive searches for the most diverse subset within a segment.

4 Branch and Bound for k-SDPPs

In this section, we will introduce our new model k-SDPP and the branch and bound method to optimally allocate the size k to m disjointed consecutive video segments $G_1, G_2, ..., G_m$ where the ground set $G = \bigcup_{i=1}^{m} G_i$ and each segment G_i contains c frames except the last segment which may contain less than c frames.

We select a subset S_i from each segment G_i, $i \in \{1, 2, ..., m\}$ and there are $c_i = 2^{|G_i|}$ choices. Each subset S_i should be as different as possible from the previous selection S_{i-1} and the other segments' selections should have no influence on it. That means each segment's selection S_i should be much different from the selection S_{i-1}. But those selections which are far away from S_i chronologically speaking should have no influence on it. That is

$$\begin{aligned}&P(X_{i-1} = S_{i-1}, X_i = S_i)\\ &= P(X_{i-1} = S_{i-1})P(X_i = S_i|X_{i-1} = S_{i-1})\end{aligned} \tag{10}$$

Meanwhile, to fix the size of the summary, the selections of these segments should have the constraint as follows

$$|S_1| + |S_2| + ... + |S_m| \leq k \tag{11}$$

Let P_{ij} be the DPP probability of selecting S_i from G_i, w_{ij} be the number of frames in S_i and x_{ij} be the variable to indicate which subset of G_i is selected. As we know, there is only one subset of G_i to be included in the final result. Thus we define our problem in the following formula.

$$\begin{aligned}\text{maximize}\quad & \mathcal{P} = \sum_{i=1}^{m}\sum_{j=1}^{c_i} P_{ij}x_{ij}\\ \textit{s.t.}\quad & \sum_{i=1}^{m}\sum_{j=1}^{c_i} w_{ij}x_{ij} \leq k\\ & \sum_{j=1}^{c_i} x_{ij} = 1\\ & x_{ij} \in \{0, 1\}\end{aligned} \tag{12}$$

By Equation 12, we find that our problem is NP-hard which can be easily proved by reduction from the knapsack problem where the probability which we aim to maximize is the profit while the knapsack capacity is k. Specifically, our problem is essentially a multiple-choice knapsack problem (MCKP) [Kellerer *et al.*, 2004] and MCKP is equal to an Integer Programming problem which is NP-complete [Wolsey, 1998].

The NP-completeness of our problem lies in the integer constraint of the program in Equation 12. To make our problem tractable, we relax the constraint of x_{ij} from $x_{ij} \in \{0, 1\}$ to $x_{ij} \in [0, 1]$, thus the *branch and bound* method [Morrison *et al.*, 2016] can be adopted to optimally solve the program.

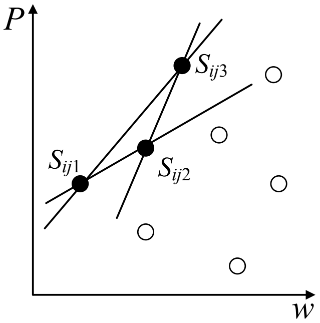


Figure 1: LP domination in segment G_i

4.1 Branch

Initially we branch the first segment. As we know that the ith segment has $c_i = 2^{|G_i|}$ choices, intuitively we need to access all c_i branches for segment i. Fortunately, the vast majority of the branches need not to be accessed due to the Pareto domination [Jin and Sendhoff, 2008] which is also popular for skyline computation in database area [Börzsöny *et al.*, 2001].

For two subsets S_{ij_1} and S_{ij_2} of G_i, if $w_{ij_1} = |S_{ij_1}| < |S_{ij_2}| = w_{ij_2}$ and $P_{ij_1} > P_{ij_2}$, we can see that subset S_{ij_2} will not be included in the final summary because the subset with larger probability and smaller size is preferred. By the Pareto domination, the total number of branches for segment G_i can be sharply decreased from $2^{|G_i|}$ to $|G_i|$.

Moreover, for 3 subsets S_{ij_1}, S_{ij_2} and S_{ij_3} of G_i, if following domination satisfies

$$\frac{P_{ij_3} - P_{ij_2}}{w_{ij_3} - w_{ij_2}} \geq \frac{P_{ij_2} - P_{ij_1}}{w_{ij_2} - w_{ij_1}} \tag{13}$$

then we say S_{ij_2} is linear programming (LP) dominated by S_{ij_1} and S_{ij_3}, then S_{ij_2} has no possibility to be included in the final result set.

Figure 1 shows the LP domination of three subsets S_{ij_1}, S_{ij_2} and S_{ij_3}. We know that these 3 subsets lie on the Pareto front or skyline of all the subsets of G_i, however S_{ij_2} is LP dominated by S_{ij_1} and S_{ij_3}, and it needs not to be expanded. Note that the determination of non-LP dominated points is closely related to the problem of finding the convex hull of a set of points in 2*d* space. Efficient algorithms *e.g.,* [Chan, 1996] can find the convex hull in time $O(|G_i| \log h_i)$ where h_i is the number of vertexes of the convex hull. If $|G_i|$ is small, we can directly compute the convex hull by Equation 13.

After these two kinds of branch pruning, there are few branches needed to be expanded for further processing which greatly improves the efficiency.

4.2 Bound

We propose the bounding technique to prune the branches which do not need to be expanded to find the final optimal solution. We keep the maximum value of the sum of the DPP probabilities for each having traversed path as the lower bound l.

For each branch which is on decision whether or not to be expanded, we calculate the upper bound u of it if we follow

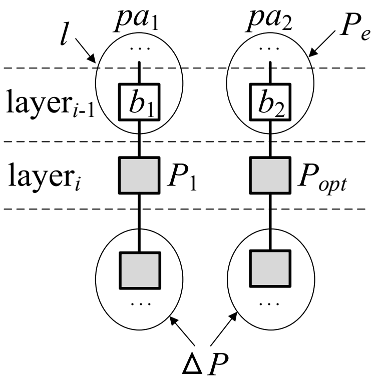


Figure 2: Safely pruning by bounding

this branch for video summarization. If $l > u$, the branch will be safely pruned. Compared to the lower bound, the calculation of the upper bound for each branch is the main focus of our BB method.

A simple way to calculate the upper bound is to follow the method in [Zemel, 1980; Kellerer *et al.*, 2004] which is determined by the sum of the optimal probabilities of sequential DPP for each segment. But the upper bound computed in this way may be large and also expensive resulting in that there are lots of branches are expanded which have no possibility to be included in the final optimal solution. Instead, we only compute the optimal ratio of the DPP probability value against its size P_{opt}/w_{opt} of the child branches to decide whether to prune the branch. Here, for we relax x_{ij} to be in the range $[0, 1]$, P_{opt}/w_{opt} can be obtained by linear programming thus Simplex, Interior Point method etc. [Bertsimas and Tsitsiklis, 1997] can be adopted to solve the LP problem.

Further, we find the branch can be safely pruned if following condition is satisfied

$$l > P_e + \frac{P_{opt}}{w_{opt}} k_r \tag{14}$$

where l is current maximum sum of DPP probabilities achieved, P_e is the sum of DPP probabilities achieved from root to current branch and k_r is the remaining size of k respectively. The reason lies in that the optimal solution corresponding to the path along with the pruned branch will not be better than the one corresponding to the path in which the lower bound obtained. Figure 2 illustrates the pruning idea of bounding. Assume at layer i (segment i), there are two branches b_1, b_2 and sum DPP probability values till them are lower bound l and P_e respectively. If $l > P_e + \frac{P_{opt}}{w_{opt}} k_r$ (Equation 14), for $k_r \geq w_{opt}$ (k_r is the remaining capacity of k if we consider k as knapsack capacity which is obviously larger than w_{opt}), we get $l > P_e + P_{opt}$. Assume the remaining of path pa_2, *i.e.,* descendants of branch b_1 achieves the optimal DPP probability value ΔP, the total DPP probabilities of path pa_2 is $P_e + P_{opt} + \Delta P$. Meanwhile, for path pa_1, from layer i, it expands the same descendant branches from b_1 as pa_2. Thus $P_1 \geq 0$ and the sum of the DPP probabilities for the remaining nodes of pa_1 is the same with pa_2 because it follows the Bayesian properties of seqDPP (Equations 8, 9, 10). We have that the total probability of path pa_1 is larger than that of path pa_2 as shown below

$$\mathcal{P}_{pa_1} = l + P_1 + \Delta P \geq P_e + P_{opt} + \Delta P = \mathcal{P}_{pa_2} \tag{15}$$

That means along with path pa_1, there exists a feasible solution which is better than the optimal solution along with path pa_2. Thus branch b_2 can be safely pruned.

Note that for seqDPP greedily selects child branches to expand to get the final solution, if we restrict the output size of seqDPP to be k, seqDPP can be regarded as a special case of our BB method.

5 Experiments

We evaluate our approach along with related DPP methods in the literature for video summarization.

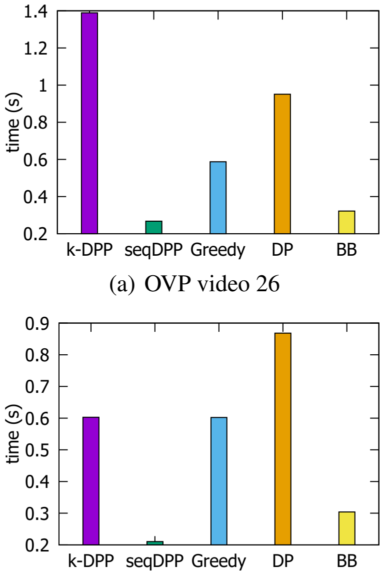
5.1 Setup

Datasets. We validate our method on two video datasets: the Open Video Project (OVP) dataset, and the YouTube dataset [de Avila *et al.*, 2011] with 50, 39 videos respectively (for YouTube dataset, there are total 50 videos, and we remove 11 cartoon movies for the quantity of these videos is not enough for training a DPP model). They both have 5 human-created summaries per video. We uniformly sample one frame per second for each video as their ground set. Same as [Gong *et al.*, 2014], for YouTube dataset, we randomly choose 31 videos of them for training and the rest 8 videos for testing. For OVP dataset, we randomly choose 40 videos of them for training and the rest 10 videos for testing.

Features. Each frame is encoded with an L_2-normalized 8192-dimensional Fisher vector [Perronnin and Dance, 2007], using SIFT features to compute it [Lowe, 2004]. The Fisher vector can well represent the content of a frame. In order to consider the correlation between frames, we also obtain 12-dimensional contextual features by adjusting the size of the window. We also add features computed from the frame saliency map [de Avila *et al.*, 2011] as [Gong *et al.*, 2014] does.

Evaluation metrics. In order to measure how much two summaries are in agreement with each other, we compute the pairwise distances between all frames across them. Two frames are "matched" if their visual distance is below a certain threshold. Each frame appears in the matched pairs at most once. We denote the summary generated by the method as A, the human-created summary as B and the matched frames as M. Following metrics are introduced: a) Precision, $P = |M|/|A|$; b) Recall, $R = |M|/|B|$; and c) F-score, $F = 2PR/(P + R)$.

Learning. To overcome the deficiency of vanilla DPPs and k-DPP, we follow seqDPP which has more flexible and powerful representations to reparameterize the $\boldsymbol{L}$ matrix with f_i where f_i is the feature representation for the frame i (by concatenating Fisher, contextual and saliency features of a frame). We use linear embedding which is introduced in [Gong *et al.*, 2014] as our learning method.



(a) OVP video 26

(b) YouTube video 90

Figure 3: Running time of the 5 methods

Compared methods. Total 5 methods including our proposed BB method are compared.

- k-DPP. The method restricts output set size to be k over standard DPPs [Kulesza and Taskar, 2011a].
- seqDPP. The sequential DPP method proposed in [Gong *et al.*, 2014].
- Greedy. Similar to the greedy approach of Partition-DPP [Zhang *et al.*, 2016] but along with the partition strategy in this paper.
- DP. With Markovian assumption for the relationship of the segments [Affandi *et al.*, 2012] and adopt the dynamic programming, specifically memoization technique [Michie, 1968] for avoiding repeated computations within recursions in dynamic programming, to allocate k to m segments.
- BB. Our proposed branch and bound method in this paper.

5.2 Results

Figure 3 shows the efficiency of the 5 methods on OVP video 26 and YouTube video 90 which are relatively longer than other videos. It can be observed that our BB method and seqDPP perform much better than other 3 methods. Further, seqDPP is a little better than BB because seqDPP is a greedy version of BB when they output same size summaries.

Table 1 and 2 show F-score, Recall and Precision values over OVP and YouTube datasets with different summary sizes. We set different k to show the performance of the 4 methods. For k-DPP stubbornly requires that the summary's size must be k, when the number of total frames of some datasets is less than k, we output all the frames as summary. For seqDPP which automatically outputs no fixed-size summary, we ignore it for comparison. From Table 1 and 2, we can see that our BB method is significantly better than other 3 methods. The results indicate the advantage of our BB method which considers the sequential nature of the frames and outputs fixed-size summaries. The lower efficiency of other 3 methods is caused by their specific assumptions. For the k-DPP and Greedy methods, they assume the frames in the video are independent while for the DP method, it follows Markov relationship of the frames which does not erase the influence of the frames far from each other. We also notice that when k is small, k-SDPP has high precision values but with relatively low recall values. This is because the summaries only satisfy specific users' tastes instead of all users'. When k becomes large, our BB method has large recall values but the precision values are relatively small. The reason is that the summaries reported by BB can satisfy most users.

Table 3 shows the performance of all the 5 methods when the segment size equals 10. The summary size of seqDPP is not determined until it outputs the results. We find that the summary size of seqDPP is usually close to T (the number of the video's segments). Thus for other 4 methods, we set their summary sizes in the range $[T-5, T+5]$ and compare their performance with seqDPP. From Table 3, we find our BB method achieves similar performance to seqDPP on both OVP and YouTube datasets. BB is better than the other 3 fixed-size methods, including k-DPP, Greedy and DP. Note that BB is not obviously better than seqDPP because the summary size output by seqDPP is not equal to that of BB. The experimental results indicate the summary sizes of seqDPP are larger than those of BB in most cases.

Figure 4 and 5 show one user summary and the summaries of the 5 methods on OVP video 21 and YouTube video 73 respectively. We first obtain the output size of seqDPP, then we set k to this size. We only illustrate F-score values because the values of F-score, Recall and Precision are the same due to the same summary size. From Figure 4 and 5 we can see that our BB method is much better than the k-DPP, Greedy and DP methods. Also, with same summary size, BB is better than seqDPP on both OVP video 21 and YouTube video 73.

6 Conclusion

In this paper, we propose a new DPP model named k-SDPP with efficient branch and bound method. Our k-SDPP model reserves sequential nature of video frames and outputs fixed-size video summaries. Moreover, the branch and bound method has the ability to provide optimal solution by branch procedure with LP-domination and bound procedure with efficient branch pruning. Experimental results show that our proposed method is prior to existing DPP-based methods for video summarization.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China and the Civil Aviation Administration of China under grant U1733112, the National Natural Science Foundation of China under grant 61702260, the Fundamental Research Funds for the Central Universities under grant NS2020068 and the Foundation of Graduate Innovation Center in NUAA under grant KFJJ20191602.

k	k-DPP			Greedy			DP			BB		
	F	R	P	F	R	P	F	R	P	F	R	P
5	57.3±0.3	47.8±0.4	72.1±0.6	53.4±0.3	42.4±0.4	72.8±0.5	53.7±0.2	44.8±0.4	71.0±0.5	**58.8**±0.3	49.8±0.5	75.4±0.4
10	62.0±0.4	69.1±0.6	56.5±0.5	62.5±0.4	63.2±0.6	62.2±0.4	59.3±0.4	67.9±0.6	56.1±0.5	71.1±0.3	77.4±0.4	67.7±0.4
15	58.2±0.4	81.0±0.5	45.8±0.5	65.7±0.3	74.9±0.4	58.8±0.4	60.6±0.4	83.5±0.4	47.6±0.5	72.1±0.3	83.1±0.4	65.4±0.4
20	52.3±0.3	88.3±0.3	37.3±0.3	66.3±0.3	79.4±0.5	57.1±0.4	57.7±0.4	91.3±0.4	42.4±0.4	72.4±0.3	84.7±0.4	64.9±0.4

Table 1: Performance of fixed size with segment size 10 on OVP dataset, measured by F-Score (F), Recall (R), and Precision (P)

k	k-DPP			Greedy			DP			BB		
	F	R	P	F	R	P	F	R	P	F	R	P
5	48.0±0.6	39.7±0.8	61.4±0.5	45.5±0.5	35.6±0.4	63.7±0.6	48.0±0.3	40.0±0.4	60.4±0.4	50.2±0.4	40.4±0.7	67.2±0.6
10	52.6±0.5	57.5±0.7	48.8±0.6	52.9±0.5	50.6±0.5	55.8±0.5	50.5±0.3	54.3±0.6	47.7±0.4	55.5±0.4	56.1±0.8	57.2±0.6
15	50.5±0.4	68.8±0.9	40.1±0.3	55.0±0.5	59.4±0.5	51.4±0.5	53.7±0.4	64.8±0.6	46.1±0.6	56.7±0.6	63.2±0.7	53.6±0.6
20	48.0±0.3	75.1±0.7	35.4±0.3	54.9±0.5	63.9±0.6	48.3±0.5	53.4±0.4	71.0±0.6	43.1±0.6	57.4±0.6	67.1±0.7	52.1±0.7

Table 2: Performance of fixed size with segment size 10 on YouTube dataset, measured by F-Score (F), Recall (R), and Precision (P)

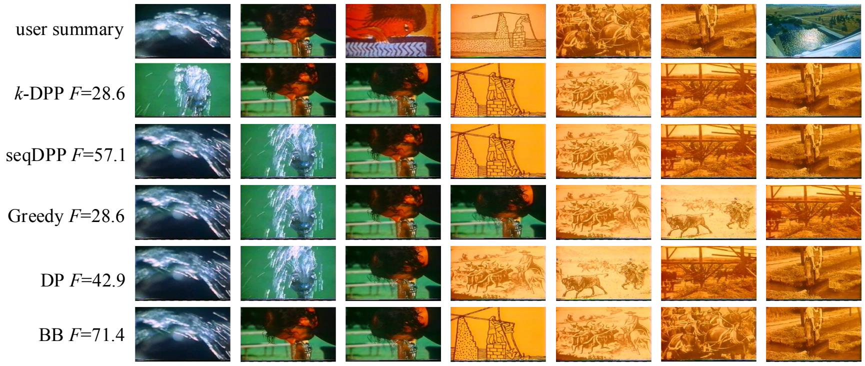


Figure 4: Summaries of different methods for OVP video 21

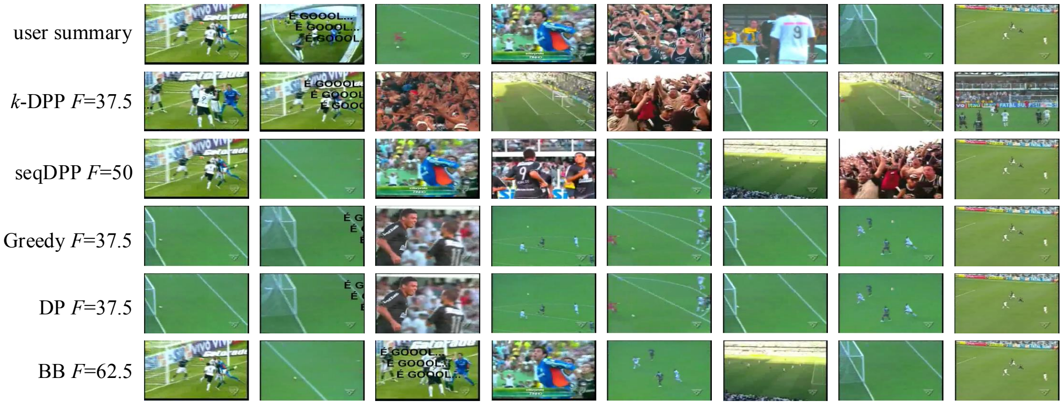


Figure 5: Summaries of different methods for YouTube video 73

Method	OVP			YouTube		
	F	R	P	F	R	P
k-DPP	61.3±0.4	70.3±0.4	54.2±0.5	52.8±0.4	66.9±0.5	44.6±0.5
seqDPP	75.3±0.7	80.4±0.9	77.8±1.0	57.8±0.5	69.8±0.5	54.2±0.7
Greedy	64.9±0.3	63.4±0.6	67.4±0.4	54.6±0.4	60.7±0.7	51.3±0.7
DP	66.0±0.4	75.2±0.5	60.0±0.6	55.9±0.4	68.1±0.5	49.9±0.5
BB	74.0±0.5	77.8±0.4	75.5±0.5	58.4±0.5	70.2±0.6	53.4±0.8

Table 3: Performance of various video summarization methods with segment size 10 on OVP and YouTube datasets

References

[Affandi *et al.*, 2012] Raja Hafiz Affandi, Alex Kulesza, and Emily B. Fox. Markov determinantal point processes. In *UAI*, pages 26–35, 2012.

[Bertsimas and Tsitsiklis, 1997] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[Börzsöny *et al.*, 2001] Stephan Börzsöny, Donald Kossmann, and Konrad Stocker. The skyline operator. In *ICDE*, pages 421–430, 2001.

[Celis *et al.*, 2018] Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and diverse DPP-based data summarization. In *ICML*, pages 716–725, 2018.

[Chan, 1996] Timothy M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.

[de Avila *et al.*, 2011] Sandra Eliza Fontes de Avila, Ana Paula Brandão Lopes, Antonio da Luz, Jr., and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[Eri, 2019] Ericsson mobility report. https://www.ericsson.com/en/mobility-report, November 2019. Accessed April 25, 2020.

[Gong *et al.*, 2014] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, pages 2069–2077, 2014.

[Hough *et al.*, 2006] J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and Bálint Virág. Determinantal processes and independence. *Probability Surveys*, 3(1):206–229, 2006.

[Jin and Sendhoff, 2008] Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(3):397–415, 2008.

[Kathuria and Deshpande, 2016] Tarun Kathuria and Amit Deshpande. On sampling and greedy MAP inference of constrained determinantal point processes. *CoRR*, abs/1607.01551, 2016.

[Kellerer *et al.*, 2004] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer, 2004.

[Krause *et al.*, 2008] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

[Kulesza and Taskar, 2011a] Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *ICML*, pages 1193–1200, 2011.

[Kulesza and Taskar, 2011b] Alex Kulesza and Ben Taskar. Learning determinantal point processes. In *UAI*, pages 419–427, 2011.

[Kulesza and Taskar, 2012] Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., Hanover, MA, USA, 2012.

[Li *et al.*, 2016] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Efficient sampling for k-determinantal point processes. In *AISTATS*, pages 1328–1337, 2016.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Macchi and Odile, 1975] Macchi and Odile. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

[Michie, 1968] Donald Michie. Memo functions and machine learning. *Nature*, 218:19–22, 1968.

[Mirzasoleiman *et al.*, 2018] Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming nonmonotone submodular maximization: Personalized video summarization on the fly. In *AAAI*, pages 1379–1386, 2018.

[Morrison *et al.*, 2016] David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102, 2016.

[Perronnin and Dance, 2007] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, pages 1–8, 2007.

[Sharghi *et al.*, 2016] Aidean Sharghi, Boqing Gong, and Mubarak Shah. Query-focused extractive video summarization. In *ECCV*, pages 3–19, 2016.

[Sharghi *et al.*, 2018] Aidean Sharghi, Ali Borji, Chengtao Li, Tianbao Yang, and Boqing Gong. Improving sequential determinantal point processes for supervised video summarization. In *ECCV*, pages 533–550, 2018.

[Vivekraj *et al.*, 2019] V.K. Vivekraj, Debashis Sen, and Balasubramanian Raman. Video skimming: Taxonomy and comprehensive survey. *ACM Computing Surveys*, 52(5), 2019.

[Wolsey, 1998] Laurence A. Wolsey. *Integer programming*. Wiley, 1998.

[Zemel, 1980] Eitan Zemel. The linear multiple choice knapsack problem. *Operations Research*, 28(6):1412–1423, 1980.

[Zhang *et al.*, 2016] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, pages 766–782, 2016.

[Zhou *et al.*, 2010] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences, PNAS*, 107(10):4511–4515, 2010.