Large-Margin Determinantal Point Processes

Wei-Lun Chao[*]
U. of Southern California
Los Angeles, CA 90089
weilunc@usc.edu

Boqing Gong[*]
U. of Southern California
Los Angeles, CA 90089
boqinggo@usc.edu

Kristen Grauman
U. of Texas at Austin
Austin, TX 78701
grauman@cs.utexas.edu

Fei Sha
U. of Southern California
Los Angeles, CA 90089
feisha@usc.edu

In this Supplementary Material, we provide extra details on the following:

- Sec. A: deriving the softmax, eq. (13) in the main text. We also show how to efficiently compute the marginal probability $P_{\{i\}}$ in eq. (13).
- Sec. B: subgradients of the objective function of our large-margin DPP (cf. eq. (14) in the main text).
- Sec. C: extra details on generating oracle video summaries and evaluating summarization results against user summaries.

A CALCULATING THE SOFTMAX

In the main text, we use softmax to deal with the exponential number of large-margin constraints and arrive at eq. (13) in the main text. Here we show how to calculate the right-hand side of eq. (13).

We first compute $\sum_{\boldsymbol{y}\subseteq\mathcal{Y}_n} \ell_\omega(\boldsymbol{y}_n,\boldsymbol{y})P(\boldsymbol{y};\boldsymbol{L}_n)$ as follows

$$\sum_{\boldsymbol{y}\subseteq\mathcal{Y}_n} \ell_\omega(\boldsymbol{y}_n,\boldsymbol{y})P(\boldsymbol{y};\boldsymbol{L}_n)$$

$$= \sum_{\boldsymbol{y}\subseteq\mathcal{Y}_n}\left[\sum_{i:i\in\boldsymbol{y}}\mathbb{I}(i\notin\boldsymbol{y}_n)+\omega\sum_{i:i\notin\boldsymbol{y}}\mathbb{I}(i\in\boldsymbol{y}_n)\right]P(\boldsymbol{y};\boldsymbol{L}_n) \tag{17}$$

$$= \sum_{i=1}^{\mathsf{M}}\left[\sum_{\boldsymbol{y}:i\in\boldsymbol{y}}\mathbb{I}(i\notin\boldsymbol{y}_n)P(\boldsymbol{y};\boldsymbol{L}_n) + \omega\sum_{\boldsymbol{y}:i\notin\boldsymbol{y}}\mathbb{I}(i\in\boldsymbol{y}_n)P(\boldsymbol{y};\boldsymbol{L}_n)\right] \tag{18}$$

$$= \sum_{i=1}^{\mathsf{M}}\left[\mathbb{I}(i\notin\boldsymbol{y}_n)P_{n_{\{i\}}}+\omega\mathbb{I}(i\in\boldsymbol{y}_n)\left(1-P_{n_{\{i\}}}\right)\right] \tag{19}$$

$$= \sum_{i:i\notin\boldsymbol{y}_n}P_{n_{\{i\}}}+\omega\sum_{i:i\in\boldsymbol{y}_n}(1-P_{n_{\{i\}}}) \tag{20}$$

$$= \sum_{i:i\notin\boldsymbol{y}_n}K_{n_{ii}}+\omega\sum_{i:i\in\boldsymbol{y}_n}(1-K_{n_{ii}}), \tag{21}$$

where $P_{n_{\{i\}}} = K_{n_{ii}}$ is the marginal probability of selecting item i. Now we are ready to see

$$\mathsf{softmax}_{\boldsymbol{y}\subseteq\mathcal{Y}_n}\log\ell_\omega(\boldsymbol{y}_n,\boldsymbol{y})+\log P(\boldsymbol{y};\boldsymbol{L}_n)$$

$$=\log\sum_{\boldsymbol{y}\subseteq\mathcal{Y}_n}\ell_\omega(\boldsymbol{y}_n,\boldsymbol{y})P(\boldsymbol{y};\boldsymbol{L}_n) \tag{22}$$

$$=\log\left(\sum_{i:i\notin\boldsymbol{y}_n}K_{n_{ii}}+\omega\sum_{i:i\in\boldsymbol{y}_n}(1-K_{n_{ii}})\right). \tag{23}$$

Moreover, recall that $\boldsymbol{K} = \boldsymbol{L}(\boldsymbol{L}+\boldsymbol{I})^{-1}$. Eigendecomposing $\boldsymbol{L}=\sum_m\lambda_m\boldsymbol{v}_m\boldsymbol{v}_m^T$, we have

$$\boldsymbol{K}=\boldsymbol{L}(\boldsymbol{L}+\boldsymbol{I})^{-1}=\sum_m\frac{\lambda_m}{\lambda_m+1}\boldsymbol{v}_m\boldsymbol{v}_m^T,$$

$$\text{and thus,}\quad K_{ii}=\sum_m\frac{\lambda_m}{\lambda_m+1}v_{mi}^2. \tag{24}$$

B SUBGRADIENTS OF THE OBJECTIVE FUNCTION

Recall that our objective function in eq. (14) of the main text actually consists of a likelihood term $\mathcal{L}(\cdot)$ and the other term of undesirable subsets. Denote them respectively by

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\alpha};\mathcal{Y}_n,\boldsymbol{y}_n)\triangleq\log P(\boldsymbol{y}_n;\boldsymbol{L}_n) = \log\det(\boldsymbol{L}_{n_{\boldsymbol{y}_n}})-\log\det(\boldsymbol{L}_n+\boldsymbol{I}), \tag{25}$$

$$\mathcal{A}(\boldsymbol{\theta},\boldsymbol{\alpha};\mathcal{Y}_n,\boldsymbol{y}_n)\triangleq\log\left(\sum_{i\notin\boldsymbol{y}_n}K_{n_{ii}}+\omega\sum_{i\in\boldsymbol{y}_n}(1-K_{n_{ii}})\right). \tag{26}$$

For brevity, we drop the subscript n of $\boldsymbol{L}_n$ and $K_{n_{ii}}$ and change $\boldsymbol{y}_n$ to $\boldsymbol{y}_\star$ in what follows.

[*] Equal contribution

To compute the overall subgradients, it is sufficient to compute the gradients of the above two terms, $\mathcal{L}$ and $\mathcal{A}$. Denoting by $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta\}$, we have

$$\frac{\partial\mathcal{L}}{\partial\Theta_k} = \sum_{i,j}\frac{\partial\mathcal{L}}{\partial L_{ij}}\frac{\partial L_{ij}}{\partial\Theta_k} = \mathbf{1}^T\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{L}}\circ\frac{\partial\boldsymbol{L}}{\partial\Theta_k}\right)\mathbf{1},$$
$$\frac{\partial\mathcal{A}}{\partial\Theta_k} = \mathbf{1}^T\left(\frac{\partial\mathcal{A}}{\partial\boldsymbol{L}}\circ\frac{\partial\boldsymbol{L}}{\partial\Theta_k}\right)\mathbf{1}, \tag{27}$$

where $\circ$ stands for the element-wise product between two matrices of the same size. We use the chain rule to decompose $\frac{\partial\boldsymbol{L}}{\partial\Theta_k}$ from the overall gradients on purpose. Therefore, if we change the way of parameterizing the DPP kernel $\boldsymbol{L}$, we only need care about $\frac{\partial\boldsymbol{L}}{\partial\Theta_k}$ when we compute the gradients for the new parameterization.

B.1 GRADIENTS OF THE QUALITY-DIVERSITY DECOMPOSITION

In terms of the quality-diversity decomposition (c.f. eq. (7) and (8) in the main text), we have

$$\frac{\partial\boldsymbol{L}}{\partial\alpha_k} = (\boldsymbol{q}\boldsymbol{q}^T)\circ S^k, \quad \frac{\partial L_{ij}}{\partial\theta_k} = L_{ij}(x_{ik}+x_{jk}),$$
$$\text{or} \quad \frac{\partial\boldsymbol{L}}{\partial\theta_k} = \boldsymbol{L}\circ(\boldsymbol{X}\boldsymbol{e}_k\mathbf{1}^T + \mathbf{1}\boldsymbol{e}_k^T\boldsymbol{X}^T) \tag{28}$$

where $\boldsymbol{q}$ is the vector concatenating the quality terms q_i, $\boldsymbol{X}$ is the design matrix concatenating $\boldsymbol{x}_i^T$ row by row, and $\boldsymbol{e}_k$ stands for the standard unit vector with 1 at the k-th entry and 0 elsewhere.

B.2 GRADIENTS WITH RESPECT TO THE DPP KERNEL

In what follows we calculate $\frac{\partial\mathcal{L}}{\partial\boldsymbol{L}}$ and $\frac{\partial\mathcal{A}}{\partial\boldsymbol{L}}$ in eq. (27). Noting that eq. (27) sums over all the (i,j) pairs, we therefore do not need bother taking special care of the symmetric structure in $\boldsymbol{L}$.

We will need map $\boldsymbol{L}_{\boldsymbol{y}_\star}$ "back" to a matrix $\boldsymbol{M}$ which is the same size as the original matrix $\boldsymbol{L}$, such that $\boldsymbol{M}_{\boldsymbol{y}_\star} = \boldsymbol{L}_{\boldsymbol{y}_\star}$ and all the other entries of $\boldsymbol{M}$ are zeros. We denote by $\langle\boldsymbol{L}_{\boldsymbol{y}_\star}\rangle$ such mapping, i.e., $\langle\boldsymbol{L}_{\boldsymbol{y}_\star}\rangle = \boldsymbol{M}$. Now we are ready to see,

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{L}} = \frac{\partial\log\det(\boldsymbol{L}_{\boldsymbol{y}_\star})}{\partial\boldsymbol{L}} - \frac{\partial\log\det(\boldsymbol{L}+\boldsymbol{I})}{\partial\boldsymbol{L}}$$
$$= \langle(\boldsymbol{L}_{\boldsymbol{y}_\star})^{-1}\rangle - (\boldsymbol{L}+\boldsymbol{I})^{-1}. \tag{29}$$

It is a little more involved to compute

$$\frac{\partial\mathcal{A}}{\partial\boldsymbol{L}} = \frac{1}{\sum_{i\notin\boldsymbol{y}_\star}K_{ii} + \omega\sum_{i\in\boldsymbol{y}_\star}(1-K_{ii})}$$
$$\times\left[\sum_{i\notin\boldsymbol{y}_\star}\frac{\partial K_{ii}}{\partial\boldsymbol{L}} - \omega\sum_{i\in\boldsymbol{y}_\star}\frac{\partial K_{ii}}{\partial\boldsymbol{L}}\right], \tag{30}$$

which involves $\frac{\partial K_{ii}}{\partial\boldsymbol{L}}$.

In order to calculate $\frac{\partial K_{ii}}{\partial\boldsymbol{L}}$, we start from the basic identity [Beyer, 1991] of

$$\frac{\partial\boldsymbol{A}^{-1}}{\partial t} = -\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial t}\boldsymbol{A}^{-1}, \tag{31}$$

followed by $\frac{\partial\boldsymbol{A}^{-1}}{\partial A_{mn}} = -\boldsymbol{A}^{-1}\boldsymbol{J}^{mn}\boldsymbol{A}^{-1}$, where $\boldsymbol{J}^{mn}$ is the same size as $\boldsymbol{A}$. The (m,n)-th entry of $\boldsymbol{J}^{mn}$ is 1 and all else are zeros.

Let $\boldsymbol{A} = (\boldsymbol{L}+\boldsymbol{I})$. Noting that $\boldsymbol{K} = \boldsymbol{L}(\boldsymbol{L}+\boldsymbol{I})^{-1} = \boldsymbol{I}-(\boldsymbol{L}+\boldsymbol{I})^{-1} = \boldsymbol{I}-\boldsymbol{A}^{-1}$ and thus $K_{ii} = 1-\left[\boldsymbol{A}^{-1}\right]_{ii}$, we have,

$$\frac{\partial K_{ii}}{\partial L_{mn}} = -\frac{\partial\left[\boldsymbol{A}^{-1}\right]_{ii}}{\partial L_{mn}} = -\frac{\partial\left[\boldsymbol{A}^{-1}\right]_{ii}}{\partial A_{mn}}$$
$$= \left[\boldsymbol{A}^{-1}\boldsymbol{J}^{mn}\boldsymbol{A}^{-1}\right]_{ii} = [\boldsymbol{A}^{-1}]_{mi}[\boldsymbol{A}^{-1}]_{ni}. \tag{32}$$

We can also write eq. (32) in the matrix form,

$$\frac{\partial K_{ii}}{\partial\boldsymbol{L}} = [\boldsymbol{A}^{-1}]_{\cdot i}[\boldsymbol{A}^{-1}]_{\cdot i}^T$$
$$= \boldsymbol{A}^{-1}\boldsymbol{e}_i\boldsymbol{e}_i^T\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{J}^{ii}\boldsymbol{A}^{-1}, \tag{33}$$

where $[\boldsymbol{A}^{-1}]_{\cdot i}$ is the i-th column of $\boldsymbol{A}^{-1}$.

Overall, we arrive at a concise form by writing out the right-hand-side of eq. (30) and merging some terms,

$$\sum_{i\notin\boldsymbol{y}_\star}\frac{\partial K_{ii}}{\partial\boldsymbol{L}} - \omega\sum_{i\in\boldsymbol{y}_\star}\frac{\partial K_{ii}}{\partial\boldsymbol{L}}$$
$$= \boldsymbol{A}^{-1}\boldsymbol{I}_\omega(\overline{\boldsymbol{y}_\star})\boldsymbol{A}^{-1} = (\boldsymbol{L}+\boldsymbol{I})^{-1}\boldsymbol{I}_\omega(\overline{\boldsymbol{y}_\star})(\boldsymbol{L}+\boldsymbol{I})^{-1} \tag{34}$$

where $\boldsymbol{I}_\omega(\overline{\boldsymbol{y}_\star})$ looks like an identity matrix except that its (i,i)-th entry is $-\omega$ for $i\in\boldsymbol{y}_\star$.

C VIDEO SUMMARIZATION

We provide details on 1) how to generate oracle summaries as the supervised information to learn DPPs and 2) how to evaluate system-generated summaries against user summaries. We also present more results on balancing the precision and recall through our large-margin DPP.

C.1 ORACLE SUMMARY

In the OVP dataset, each video comes along with five user summaries $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_5$ [de Avila et al., 2011]. Similar to document summarization [Kulesza and Taskar, 2011], we extract an "oracle" summary $\boldsymbol{y}_\star$ from the five user summaries using a greedy algorithm. Initialize $\boldsymbol{y}_\star = \emptyset$. From the frames not in $\boldsymbol{y}_\star$, we pick out the one i which contributes the most to the marginal gain,

$$\text{VSUMM}(\boldsymbol{y}_\star\cup\{i\}, \{\boldsymbol{y}_1,\cdots,\boldsymbol{y}_5\})$$
$$-\,\text{VSUMM}(\boldsymbol{y}_\star, \{\boldsymbol{y}_1,\cdots,\boldsymbol{y}_5\}), \tag{35}$$

where VSUMM is the package developed in [de Avila et al., 2011] to evaluate video summarization results. We postpone to Section C.2 for describing the evaluation scheme of VSUMM. Namely, we select the oracle frames greedily for each video and stop until the marginal gain becomes negative. We evaluate the oracle summaries against users' and find that they achieve high precision and recalls, 84.1% and 87.7% respectively, validating that the oracle summaries are able to serve as good supervised targets for training DPP models.

The above procedure allows a "user-independent" definition of a good oracle summary for learning. Of course if the application goal were to generate user-specific summaries catering to a particular user's taste, one would instead simply apply our framework with $\boldsymbol{y}_\star$ set to be that particular user's selection.

C.2 VSUMM: EVALUATING VIDEO SUMMARIZATION RESULTS

We evaluate video summarization results using the VSUMM package [de Avila et al., 2011]. Given two sets of summaries/frames, it searches for the maximum number of matched pairs of frames between them. Two images are viewed as a matched pair if their visual difference is below a certain threshold. VSUMM uses normalized color histograms to compute such difference. Besides, each frame of one set can be matched to at most one frame of the other set, and vice versa. After the matching procedure, one can hence develop different evaluation metrics based on the number of matched pairs. In our experiments, we define F-score, precision, and recall (cf. eq. (15) of the main text).

References

W. H. Beyer. *CRC standard mathematical tables and formulae*. CRC press, 1991.

S. E. F. de Avila, A. P. B. Lopes, et al. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

A. Kulesza and B. Taskar. Learning determinantal point processes. In *UAI*, 2011.