Dynamic Video-Stream Running Summarization

Chandrashekhar Lavania, Rishabh Iyer, Jeffrey A. Bilmes

http://terraswarm.org/

Motivation & roblem

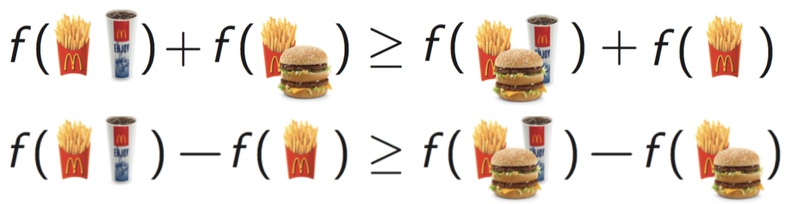
- Exists massive quantities of Videos and Live Video Streams
- Too large to store; too long to view (by humans); and too resource intensive to process in batch.
- Mostly uninteresting & ordinary; plagued with redundancy; surveillance is costly, tedious, and error-prone.

Aim: Running Video Summarization

- Automatically summarize incoming video stream online, in real time, using few compute resources (standard laptop).
- Video summarization: extract video "snippets" that together represent the video history relative to the current time
- Online algorithm: process data dynamically as it comes in, in a stream, rather than all at once in batch.
- Running summarization: summary of current history available and viewable at any time, and in real time.
- Useful for: surveillance, quick video "gisting", reducing storage requirements, reducing computational resources, reducing the deluge of big data streams.

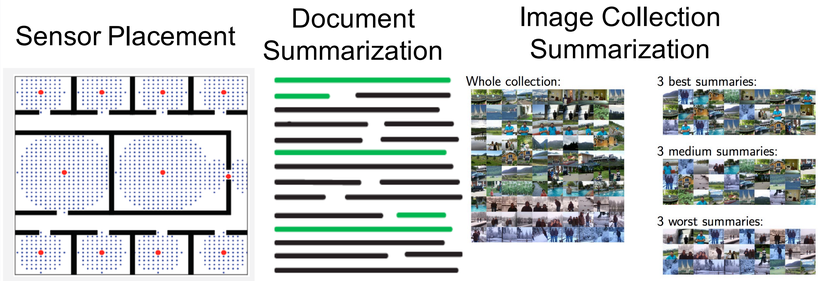
Submodular Functions

- We employ submodular functions to represent information.
- $f\colon 2^V \to \mathbb{R}, \forall X, Y \subseteq V, f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$
- Submodular functions have a natural diminishing returns property



- Submodular functions are natural combinatorial "information" functions, that apply to arbitrary data objects. Good at representing diversity, dispersion, span, and coverage.
- Important sub-area in machine learning. Some applications include sensor placement, document summarization, and image segmentation.

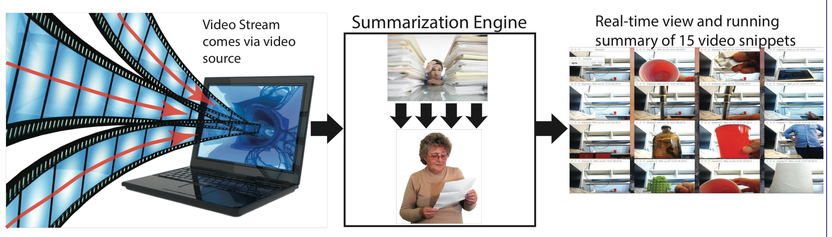
Submodular Applications in ML



- Other applications: image segmentation, graphical models, (image) denoising, data subset selection and active learning, auctions, diverse selection, signal encoding and sparse codes/structured convex norms, data partitioning, parallel computing, clustering, feature selection in pattern recognition, etc.

Our Approach: Real-Time Streaming Submodular Optimization

- Traditional submodular optimization requires storing all data in memory at the same time:
- $\underset{A \subseteq V : |A| = k}{\operatorname{argmax}} f(A)$ is solved via the greedy algorithm, and the greedy algo has a $1 - 1/e$ approximation guarantee, but it requires multiple sweeps through data, and data is stored in memory (best case compute is $O(k \log n)$.
- Streaming online real-time: summarize on the fly, as a stream of data comes in, decide at each data item if we keep it or not.
- Constraint: constant memory requirement --- if we choose not to keep data item, it is never again available for consid eration.

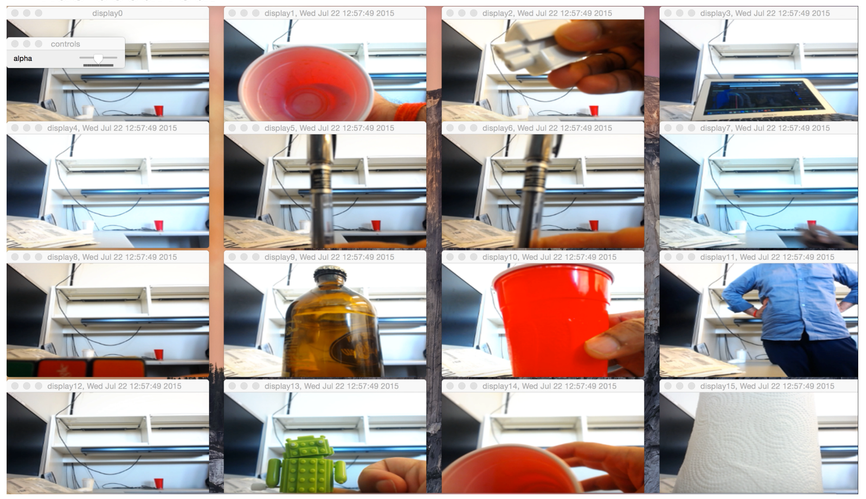


Streaming Summarization Process

- Divide incoming video stream into video snippets (snippet length is adjustable).
- As shown: a snippet consists of a 3-second temporal sequence of successive video frames.
- At the start, we add first k incoming snippets to the summary (k is the summary budget, and also can be adjustable, but for the demo we set k=15).
- On arrival of next snippet, decide to swap it with an existing snippet if submodular function gain of a swap is above a certain threshold.
- Once swapped out, old snippet is no longer available --- hence, constant memory requirements even for unboundedly long stream lengths.
- Overall memory cost is no more than the size of the summary (k in this case) since entire stream is not processed simultaneously.

Example Summary

- Live video feed is shown on upper left.
- The other 15 frames show the 15 snippets that are currently chosen --- algorithm may swap one of them out, and a new one in, when it decides that something representable/novel enough has occurred.



Sponsored by the TerraSwarm Research Center, one of six centers administered by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.