CS60021: Scalable Data Mining

Subset Selection

Sourangshu Bhattacharya

Submodular Subset Selection

Slides taken from IJCAI 2020 tutorial by
Rishabh Iyer and Ganesh Ramakrishnan

Combinatorial Subset Selection Problems

$V = \{$

$\}$

$f : 2^V \to \mathbb{R}$

$A = \{$

$\}$

Choose Subset $A \subseteq V$

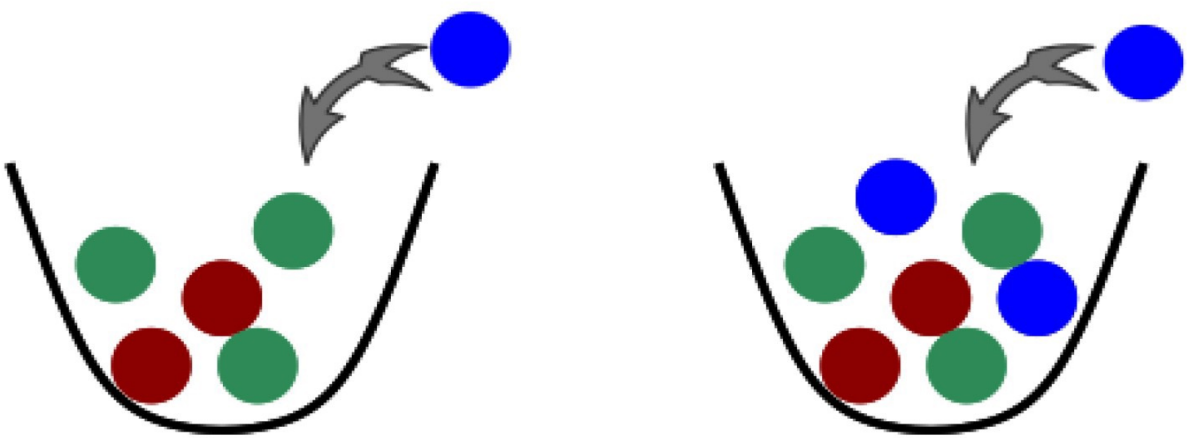
$f(A)$ is maximum

General Set function Optimization: very hard!

What if there is some special structure?

Submodular Functions

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B$$



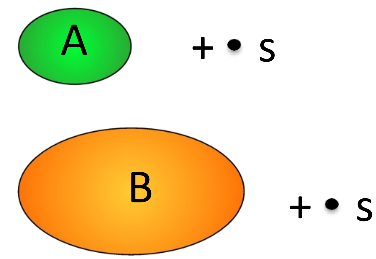
$f = \#$ of distinct colors of balls in the urn.

Negative of a Submodular Function is a Supermodular Function!

Equivalent Definitions of Submodularity

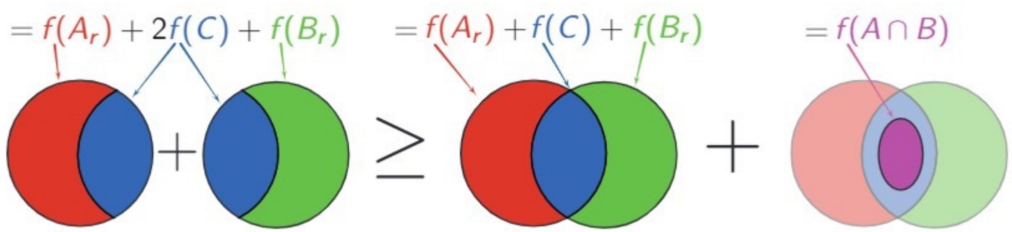
- Diminishing gains: for all $A, B \subseteq V$

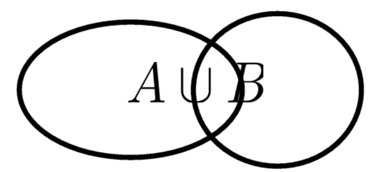
$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B$$



- Union-Intersection: for all $A, B \subseteq V$

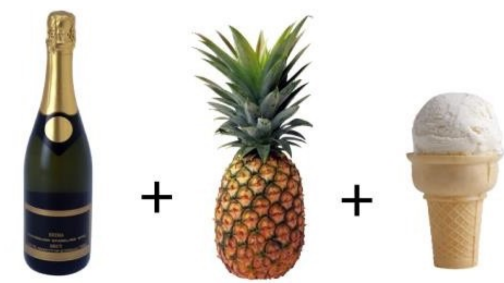
$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$





Modular Functions

- each element e has a weight $w(e)$



$$F(S) = \sum_{e \in S} w(e)$$

$$A \subset B$$

$$F(A \cup e) - F(A) = w(e) \quad = \quad F(B \cup e) - F(B) = w(e)$$

Modular Functions are both submodular and supermodular!

Monotone Submodular Functions

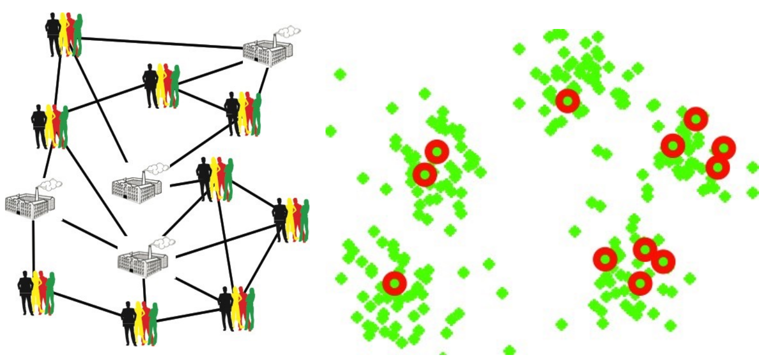
- A set function is called monotonic if
  $A \subseteq B \subseteq V \Rightarrow F(A) \leq F(B)$

- Examples:
  - Influence in social networks [Kempe et al KDD '03]
  - For discrete RVs, entropy $F(A) = H(X_A)$ is monotonic:
    Suppose $B = A \cup C$. Then
    $F(B) = H(X_A, X_C) = H(X_A) + H(X_C \mid X_A) \geq H(X_A) = F(A)$
  - Information gain: $F(A) = H(Y) - H(Y \mid X_A)$

Instantiations of Submodular Functions

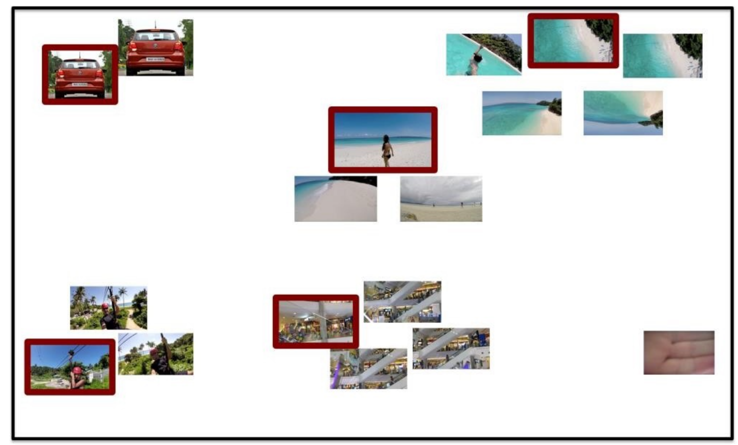
- **Representation Functions**
  - Facility Location Function (k-mediods clustering)
  - Graph Cut Family, Saturated Coverage
- **Diversity Functions**
  - Dispersion Functions (Min, Sum, Min-Sum)
  - Determinantal Point Processes
- **Coverage Functions**
  - Set Cover Function
  - Probabilistic Set Cover Function
  - Feature Based Functions
- **Importance Functions**
  - Modular Functions
- **Information Functions**
  - Mutual Information
  - Entropy
- **Discounted Cost Functions**
  - Clustered Concave over Modular Functions
  - Cooperative Costs and Saturations
- **Complexity Functions**
  - Bipartite Neighborhood Functions

Representation Functions



Facility Location	$\sum_{i \in V} \max_{k \in X} s_{ik}$
Saturated Coverage	$\sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \alpha_i\}$
Graph Cut	$\lambda \sum_{i \in V} \sum_{j \in X} s_{ij} - \sum_{i,j \in X} s_{ij}$

Similarity Kernel

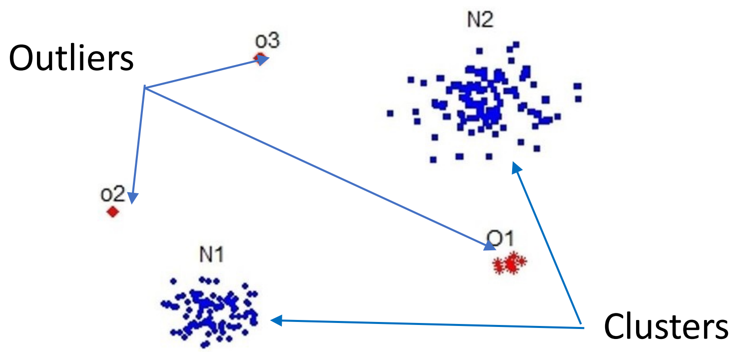


Representation Functions

Picks Centroids

Iyer 2015, Kaushal et al 2019, Tschiatchek et al2014, …

Diversity Functions: Dispersion



Dispersion Min	$\min_{k,l \in X, k \neq l} d_{kl}$
Dispersion Sum	$\sum_{k,l \in X} d_{kl}$
Dispersion Min-Sum	$\sum_{k \in X} \min_{l \in X} d_{kl}$

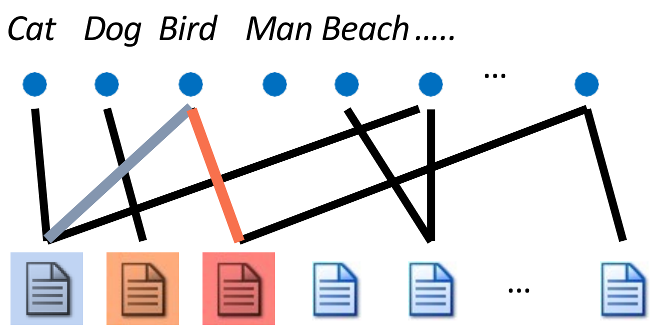
Dispersion Sum and Dispersion Min Not Submodular!

Diversity Functions

Picks items as different as possible!

Dasgupta et al 2013, Chakraborty et al 2015

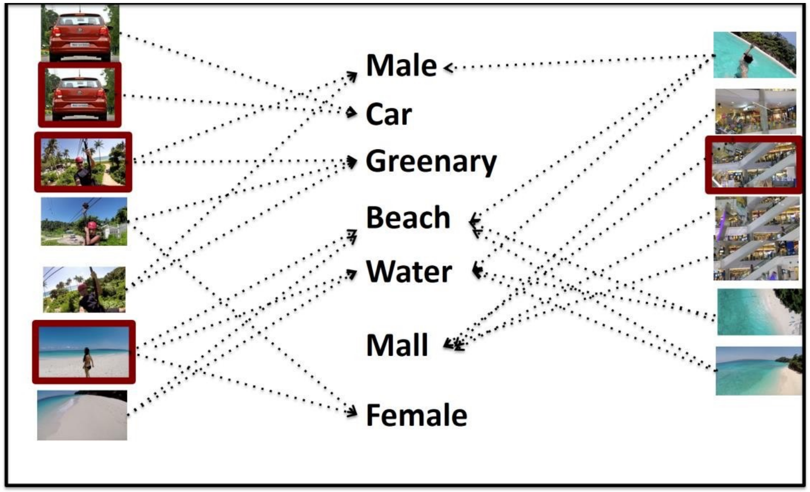
Coverage Functions



Set Cover Function

$$f(X) = w(\cup_{i \in X} U_i),$$

Concepts Covered by Instance i

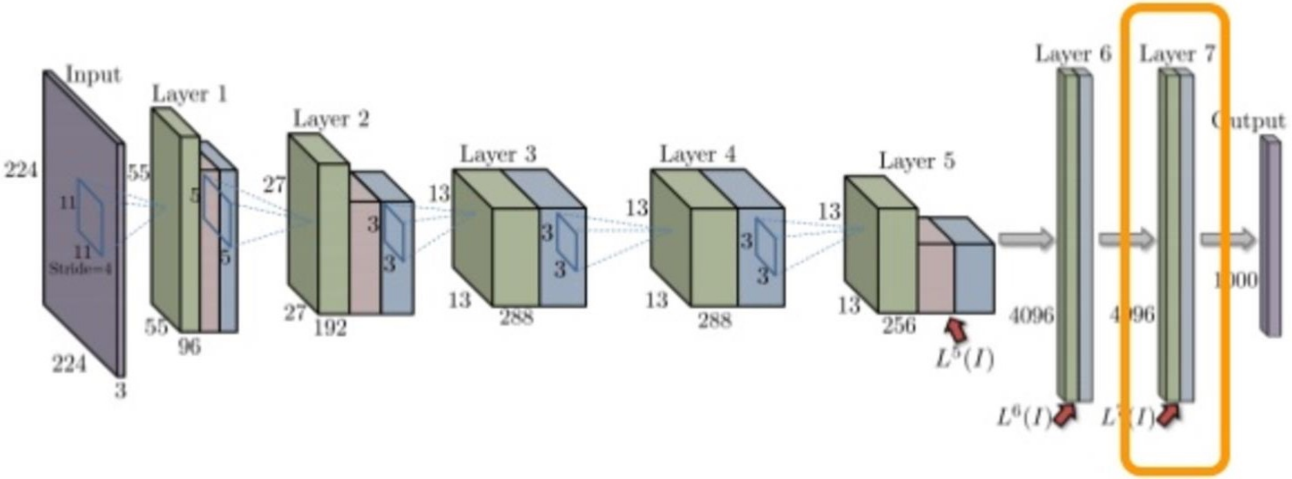


Coverage Functions

Select instances which "cover" all concepts

Wolsey et al 1982, ...

Feature Based Functions



Achieve Uniformity in Feature Coverage

Feature Based Functions

$$f_{\text{fea}}(S) = \sum_{u \in \mathcal{U}} g(m_u(S)).$$

Total Contribution of Feature u in the Set of Images S

Wei-Iyer et al 2014 …

Information Functions

$X_1, \ldots, X_n$ discrete random variables: $X_e \in \{1, \ldots, m\}$

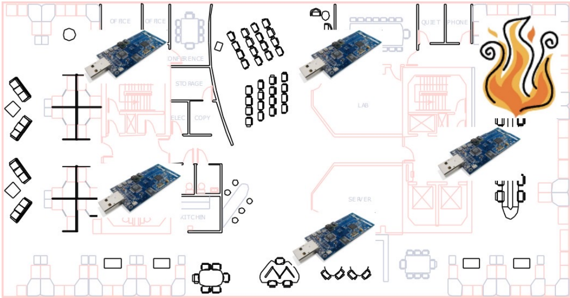
$$F(S) = H(X_S) = \text{ joint entropy of variables indexed by } S$$

$$H(X_e) = \sum_{x \in \{1,\ldots,m\}} P(X_e = x) \log P(X_e = x)$$

$$\boxed{A \subset B, e \notin B \qquad F(A \cup e) - F(A) \geq F(B \cup e) - F(B)??}$$

$$\begin{aligned} H(X_{A \cup e}) - H(X_A) \quad &= H(X_e | X_A) \\ &\leq H(X_e | X_B) \qquad \text{"information never hurts"} \\ &= H(X_{B \cup e}) - H(X_B) \end{aligned}$$

discrete entropy is submodular!



Entropy
Mutual Information
Information Gain
…

Krause et al 2008, …

Master Optimization Problem

Set Function

Selected set

$$\max_{\mathcal{A} \subseteq \mathcal{V}} F(\mathcal{A})$$

Selection cost

Budget

$$\text{subject to } C(\mathcal{A}) \leq B$$

F = Monotone Submodular,
Non Monotone Submodular,
Dispersion Functions,
....

F Models:

- Diversity
- Representation
- Coverage
- Information
- Importance
- ...

We shall study this and variants of this Master Optimization Problem!

Monotone Submodular Maximization

$$\max_{S} \; F(S) \;\; \text{s.t.} \;\; |S| \leq k$$

What is the Constraint?
C(S) = |S|

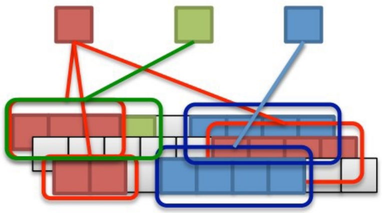
- greedy algorithm:

$S_0 = \emptyset$

for *i* = 0, ..., *k*-1

$$e^* = \arg \max_{e \in \mathcal{V} \setminus S_i} F(S_i \cup \{e\})$$

$$S_{i+1} = S_i \cup \{e^*\}$$

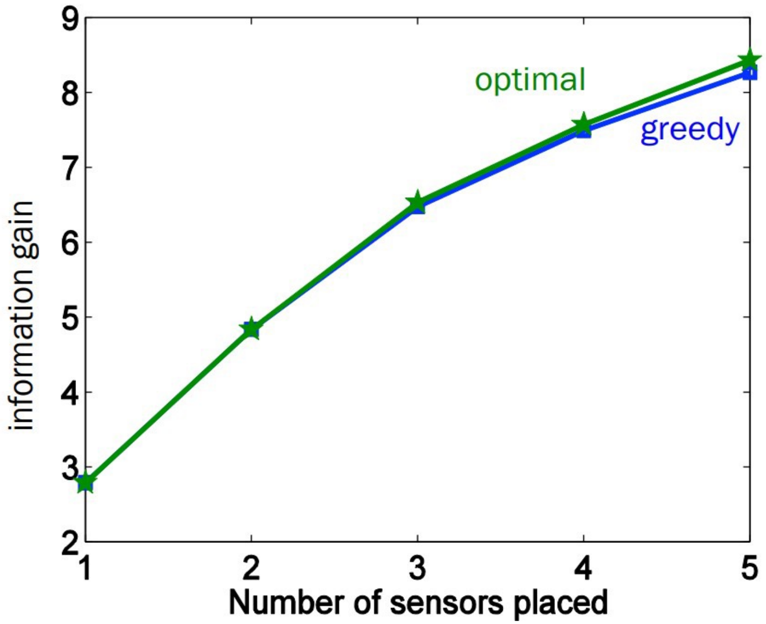


How "good" is S_k ?

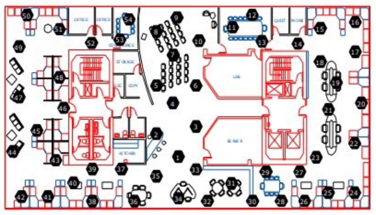
Approximation Guarantee!

How good is Greedy in Practice?

empirically:



sensor placement



How good is Greedy in Theory?

$$\max_S \; F(S) \;\; \text{s.t.} \;\; |S| \leq k$$

Theorem *(Nemhauser, Fisher, Wolsey `78)*

F monotone submodular, S_k solution of greedy. Then

$$F(S_k) \;\; \geq \;\; \left(1 - \frac{1}{e}\right) F(S^*)$$

optimal solution

No Poly-time algorithm can do better than this in the worst case!

Proof (Nemhauser et al 1978)

Let:

- $A_i = (v_1, v_2, \ldots, v_i)$ be the the chain formed by the greedy algorithm, as defined above
- $A^* = (v_1^*, v_2^*, \ldots, v_k^*)$ be the optimal solution, in an arbitrary order
- f be a monotone submodular function. Let $f \geq 0$ (*Update on 04/25/2019: I thought this was w.l.o.g., but Andrey Kolobov pointed out that we actually need* f *to be non negative*)
- $OPT = f(A^*)$, the value of the optimal solution.

We will prove that

$$f(A_k) \geq (1 - 1/e)OPT$$

Source: https://homes.cs.washington.edu/~marcotcr/blog/greedy-submodular/

Proof (Nemhauser et al 1978)

For all $i \leq k$, we have:

$$
\begin{aligned}
f(A^*) &\leq f(A^* \cup A_i) && \text{Monotonicity} \\
&= f(A_i) + \sum_{j=1}^{k} \Delta(v_j^* | A_i \cup \{v_1^*, v_2^*, \ldots, v_{j-1}^*\}) \\
&\leq f(A_i) + \sum_{z \in A^*} \Delta(z | A_i) && \text{Using submodularity} \\
&\leq f(A_i) + \sum_{z \in A^*} \Delta(v_{i+1} | A_i) && v_{i+1} = argmax_{v \in V \setminus A_i} \Delta(v|A_i) \\
&= f(A_i) + k\Delta(v_{i+1} | A_i)
\end{aligned}
$$

Rearranging the terms, we have proved that

$$
\Delta(v_{i+1} | A_i) \geq \frac{1}{k}(OPT - f(A_i))
$$

Source: https://homes.cs.washington.edu/~marcotcr/blog/greedy-submodular/

Proof (Nemhauser et al 1978)

Part I

Now we define $\delta_i = OPT - f(A_i)$. This implies
$\delta_i - \delta_{i+1} = f(A_{i+1}) - f(A_i) = \Delta(v_{i+1}|A_i)$

Plugging this into our previous equation, we have:

$$\Rightarrow \quad \delta_i - \delta_{i+1} \geq \frac{1}{k}(\delta_i)$$

$$\Rightarrow \quad \delta_{i+1} \leq (1 - \frac{1}{k})\delta_i$$

Part II

$$\Rightarrow \quad \delta_k \leq \left(1 - \frac{1}{k}\right)^k \delta_0$$

$$\Rightarrow \quad \delta_k \leq \left(1 - \frac{1}{k}\right)^k OPT \leq \frac{1}{e} OPT$$

$$\Rightarrow \quad OPT - f(A_k) \leq \frac{1}{e} OPT$$

$$\Rightarrow \quad f(A_k) \geq \left(1 - \frac{1}{e}\right) OPT$$

Source: https://homes.cs.washington.edu/~marcotcr/blog/greedy-submodular/

Monotone Submodular – Budget Constraints

$$\max \quad F(S) \ \text{ s.t. } \sum_{e \in S} c(e) \leq B$$

1. run greedy: S_{gr}
2. run a modified greedy: S_{mod}

$$e^* = \arg\max \frac{F(S_i \cup \{e\}) - F(S_i)}{c(e)}$$

3. pick better of S_{gr}, S_{mod}

➔ approximation factor:

$$\frac{1}{2}\left(1 - \frac{1}{e}\right)$$

even better but less fast: partial enumeration *(Sviridenko, 2004)* or filtering *(Badanidiyuru & Vondrák 2014)*

Sviridenko 2004:

- Run the cost-sensitive greedy algorithm starting with all possible initial sets {l,j,k}
- $O(n^3)$ initial complexity
- $(1 - 1/e)$ approximation!

Sviridenko 2004, Leskovec et al 2007

Summary: Greedy Algorithm Framework

Monotone Submodular Function

$$\max_{S \subseteq V, c(S) \leq \mathcal{B}} f(S)$$

Cost of Summary Subset S (e.g. size)

Problem Formulation

Initialization $S \leftarrow \emptyset$.
repeat
Pick an element $v^* \in \text{argmax}_{v \in V \setminus S} \frac{f(v \cup S) - f(S)}{c(v)}$
Update $S \leftarrow S \cup v^*$
until Reaching the budget, i.e., $c(S) > \mathcal{B}$

Greedy Algorithm

Non-Monotone Submodular Functions

$$\max_{S} \; F(S) \text{ s.t. } |S| \leq k$$

Start with $Y_0 = \emptyset$

for $i = 1$ *to* k **do**

Let $M_i = \text{argmax}_{X \subseteq V \setminus Y_{i-1}, |X| = k} \sum_{v \in X} f(v|Y_{i-1})$;

Choose y as a uniformly random element in M_i;

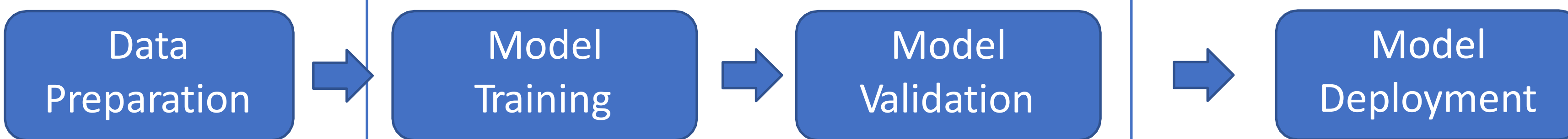
$Y_i = Y_{i-1} \cup y$;

return Y_k.

Theorem (Buchbinder et al 2014): The Randomized Greedy Algorithm achieves a 1/e approximation guarantee for Non-Monotone Submodular Maximization subject to cardinality constraints!

Data subset selection

Make ML Data Efficient and Robust



Data Preparation
1. Data Labeling
2. Feature Engineering

Model Training
1. Model Selection
2. Training Budget
3. Batch sizes, Partitions

Often repeated for hyper-parameter tuning

Model Deployment
1. Deployment Infra
2. Latency & Memory

Production Systems Constraints

1. Data Labeling => Time Consuming, Expensive, Noisy
2. Feature Selection => Latency & Memory
3. Model Training => Compute Intensive and Time Consuming
4. Hyper-Parameter Tuning/NAS => Very Time Consuming
5. Distribution Shift => Deployment vs Training

Can we train Models under these constraints without sacrificing on accuracy?

Data Subset Selection Setup

A Machine Learning model characterized by model parameters θ

Training Data: $\{(x_i, y_i), i \in \mathcal{U}\}$ Training log-likelihood function: $LL_T(\theta, \mathcal{U})$

Training a machine learning model often reduces to finding the parameters that maximizes a log-likelihood function for given training data empirically.

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, LL_T(\theta, \mathcal{U})$$

Validation Data: $\{(x_i, y_i), i \in \mathcal{V}\}$ Validation log-likelihood function: $LL_V(\theta, \mathcal{V})$

Goal: Select a subset $S \subseteq \mathcal{U}$ such that the resulting model performs the **best**!

Requirements for optimal subset selection

1. The subset selection algorithm needs to be as fast as possible.
    - Subset Selection time <<<< Full training time

Example: Subset selection algorithm with negligible time complexity

Training on 10 % Subset $\Rightarrow$ 10x Faster training

2. Theoretical guarantees of subset selection algorithm.
    - Can we show theoretical guarantees for subset selection algorithms?

Approaches for Data Subset Selection

- Several different kinds of approaches studied in literature:
  - Approach 1: Use Submodular Functions as proxy functions for data subset selection
  - **Approach 2: Choose data subset which approximates the gradient of the entire dataset**
  - Approach 3: Choose data subset which approximates the performance on full training dataset (or validation set) as a bi-level optimization!
  - Approach 4: Choose data subset which minimizes a suitable divergence (e.g. KL divergence) between the distribution induced by the subset and full data!
- Types of Data Selection
  - Supervised (Using the labels)
  - Unsupervised (No access to labels)
  - Validation based (Access to a validation set for focusing on generalization)

Idea: Gradient Matching/ CoreSets

Can we obtain a weighted gradient of a **subset** of points that approximates the full gradient?

$$\sum_{i \in X_t} w_i^t \nabla_\theta L_T^i(\theta) \approx \nabla_\theta L(\theta)$$

Sivasubramaniam & Killamsetty et. al. 2021, Mirzasoleiman et al 2020

Gradient Matching/ CoreSets Convergence

Denote L_V as the validation loss, L_T as the full training loss, and L_T^i as the training loss on the i^{th} training example. Furthermore, assume that both losses have gradients bounded by σ_T and σ_V respectively, and that the parameters satisfy $||\theta^*||^2 \leq R^2$ (θ^* is the optimal parameter). Then letting L denote either the training or validation loss (with gradient bounded by σ), any data selection algorithm, defined via weights $\mathbf{w}^t$ and subsets X_t for $t = 1, \cdots, T$, and run with a learning rate $\alpha = \frac{R}{\sigma_T\sqrt{T}}$ satisfies:

$$\min_{t=1:T} L(\theta_t) - L(\theta^*) \leq \frac{R\sigma_T}{\sqrt{T}} + \frac{R}{T}\sum_{t=1}^{T-1} \mathsf{Err}(\mathbf{w}^t, X_t, L, L_T, \theta_t)$$

where:

$$\mathsf{Err}(\mathbf{w}^t, X_t, L, L_T, \theta_t) = \left\| \sum_{i \in X_t} w_i^t \nabla_\theta L_T^i(\theta_t) - \nabla_\theta L(\theta_t) \right\|$$

Sivasubramaniam & Killamsetty et. al. 2021

Gradient Matching: Main Idea

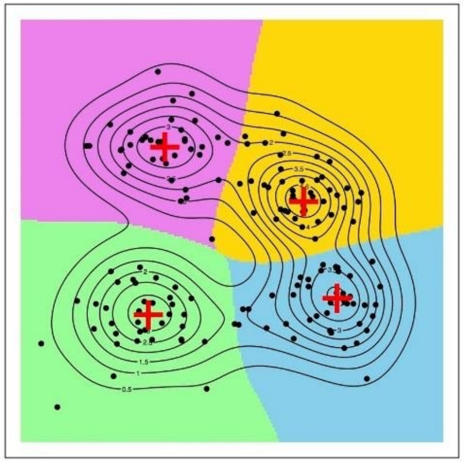
The theorem indicates that an effective data selection algorithm should try to have a low error $\text{Err}(\mathbf{w}^t, X_t, L, L_T, \theta_t)$ for $t = 1, \cdots, T$. Thus, we can pose the problem as,

$$\begin{aligned}\mathbf{w}^t, X_t &= \min_{\mathbf{w}, X : |X| \leq k} \text{Err}(\mathbf{w}, X, L, L_T, \theta_t) \\ &= \min_{\mathbf{w}, X : |X| \leq k} \| \sum_{i \in X_t} w_i^t \nabla_\theta L_T^i(\theta_t) - \nabla_\theta L(\theta_t) \|\end{aligned}$$

Sivasubramaniam & Killamsetty et. al. 2021

CRAIG as an upper bound

Facility location can be thought as an upper bound for this. Suppose we define $\pi_t^i \in \arg\min_{j \in X} \|\nabla_\theta L^i(\theta) - \nabla_\theta L_T^j(\theta)\|$, if $L = L_T$, then $W = U$ otherwise if $L = L_V$ then $W = V$ and $w_j = \sum_{i \in W} [\pi_t^i = j]$ then, for any θ_t we can write



$$\text{Err}(\mathbf{w}, X, L, L_T, \theta_t) = \left\|\nabla L(\theta_t) - \sum_{i \in X} w_i \nabla L_T^i(\theta_t)\right\|$$

$$= \left\| \sum_{i \in W} \left(\nabla_\theta L_T^i(\theta_t) - \nabla_\theta L^{\pi_t^i}(\theta_t)\right) \right\|$$

$$\leq \sum_{i \in W} \|\nabla_\theta L_T^i(\theta_t) - \nabla_\theta L^{\pi_t^i}(\theta_t)\|,$$

Sivasubramaniam & Killamsetty et. al. 2021, Mirzasoleiman et al 2020

Directly Optimizing Gradient Error: GradMatch

Define the regularized version of our objective:

$$E_\lambda(X) = \min_{\mathbf{w}} \underbrace{\Big\| \sum_{i \in X_t} w_t^i \nabla_\theta L_T^i(\theta_t) - \nabla_\theta L(\theta_t) \Big\|^2 + \lambda ||\mathbf{w}^t||^2}_{E_\lambda(X_t, \mathbf{w}^t)}$$

This problem can be solved efficiently using Orthogonal Matching Pursuit (OMP) described as,

1. Find projection of $r = \nabla_\theta L_T^i(\theta_t)$ for each $i \in W$ along $\nabla_\theta L(\theta_t)$ and chose the i with whom projection is maximum and add it X
2. Solve linear regression problem to find w_t^i for $i \in X$s.
3. Set $r = \nabla_\theta L(\theta_t) - \sum_{i \in X_t} w_t^i \nabla_\theta L_T^i(\theta_t)$
4. Repeat the steps with new r until the $|r| < \epsilon$ or $|X| < k$(budget)
5. Return X, w_t

Sivasubramaniam & Killamsetty et. al. 2021

Orthogonal Matching Pursuit

The OMP algorithm

Algorithm 1: OMP($\mathbf{A}, \mathbf{b}$)

Input: $\mathbf{A}, \mathbf{b}$

Result: $\mathbf{x}_k$

```
Initialization r_0 = b, Λ_0 = ∅;
Normalize all columns of A to unit L_2 norm;
Remove duplicated columns in A ;
for k = 1, 2, ... do
    Step-1. λ_k = argmax_{j∉Λ_{k-1}} |⟨a_j, r_{k-1}⟩|;
    Step-2. Λ_k = Λ_{k-1} ∪ {λ_k};
    Step-3. x_k(i ∈ Λ_k) = argmin_x ||A_{Λ_k} x − b||_2,   x_k(i ∉ Λ_k) = 0;
    Step-4. b̂_k = A x_k;
    Step-5. r_k ← b − b̂_k;
end
```

Initialization $\mathbf{r}_0 = \mathbf{b}$, $\Lambda_0 = \varnothing$;

Normalize all columns of $\mathbf{A}$ to unit L_2 norm;

Remove duplicated columns in $\mathbf{A}$;

for $k = 1, 2, \ldots$ **do**

- Step-1. $\lambda_k = \underset{j \notin \Lambda_{k-1}}{\operatorname{argmax}} \left|\langle \mathbf{a}_j, \mathbf{r}_{k-1} \rangle\right|$;
- Step-2. $\Lambda_k = \Lambda_{k-1} \cup \{\lambda_k\}$;
- Step-3. $\mathbf{x}_k(i \in \Lambda_k) = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{A}_{\Lambda_k}\mathbf{x} - \mathbf{b}\|_2, \quad \mathbf{x}_k(i \notin \Lambda_k) = 0$;
- Step-4. $\hat{\mathbf{b}}_k = \mathbf{A}\mathbf{x}_k$;
- Step-5. $\mathbf{r}_k \leftarrow \mathbf{b} - \hat{\mathbf{b}}_k$;

end

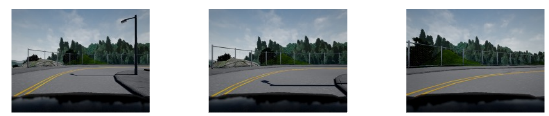
Convex DSS

Aim

- We study the problem of data efficient training of autonomous driving systems.

- Training using many frames on straight road sections may not be necessary. Frames at the turns turn out to be useful.

REDUNDANT

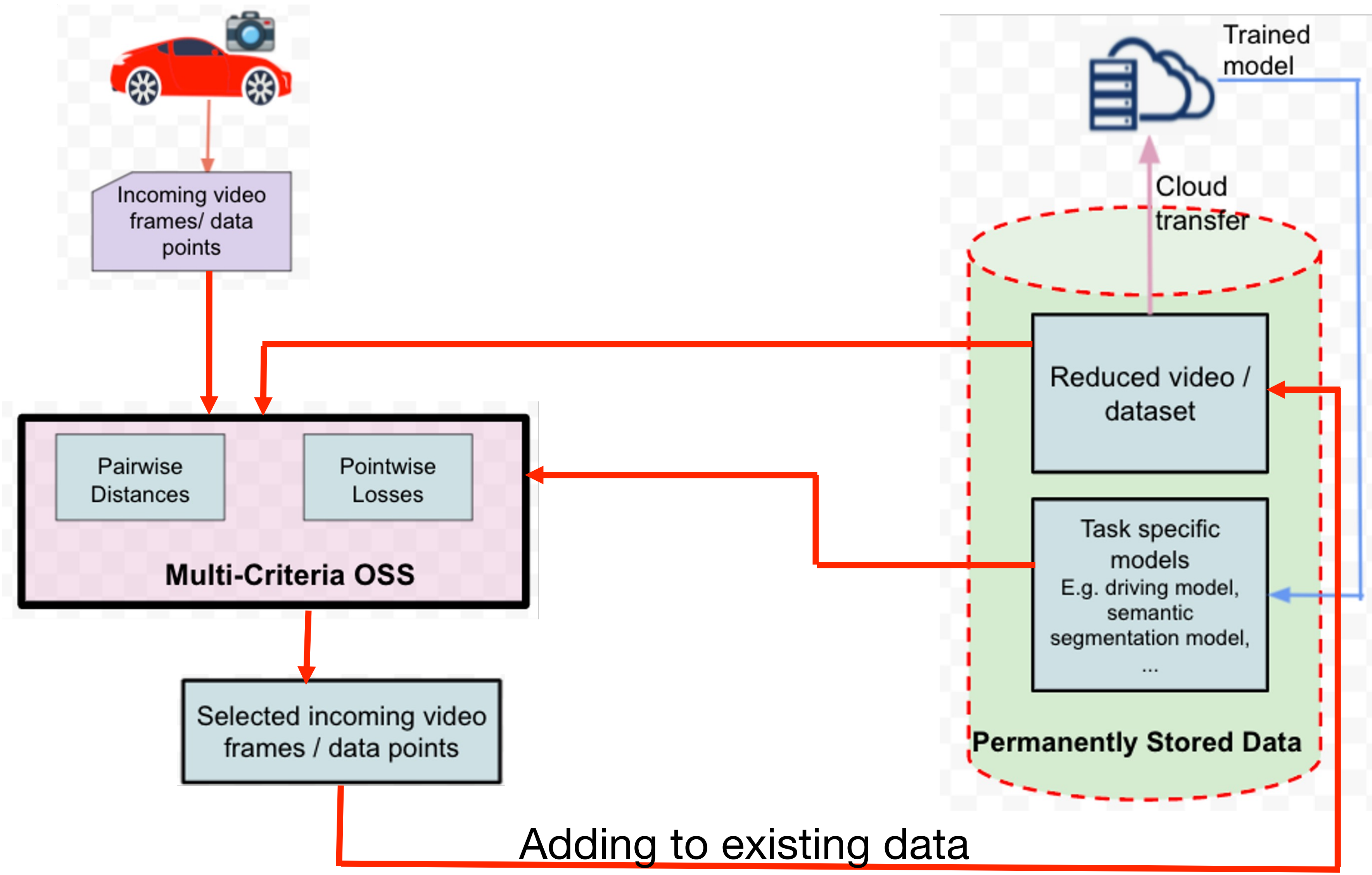


INFORMATIVE

Method	Train One-Turn	Test One-Turn
Uniform Skip	3/10	5/10

In the context of edge device deployment, multi-criteria online subset selection (OSS) framework can be useful in selecting informative frames, essential for an end-task.

Subset selection on Edge devices



High Level Idea

- Given a compression ratio, find out representatives which have the least dissimilarity with the left-out elements besides having the highest task-specific loss.

Problem Setup

- X_t: the set of incoming datapoints at time t (Size m)
- D: set of all data points (Size N)
- R_t: Reduced set of data at time t
- d_{ij}: Distance between data points I and j.
- z_{ij}: Indicator variable indicating that datapoint i is a representative for datapoint j.

Convex Subset Selection

- Original formulation in set notation:

$$\min_{\mathcal{S}\subseteq\mathcal{D}} \lambda|\mathcal{S}| + \sum_{j\in\mathcal{D}} \min_{i\in\mathcal{S}} d_{ij},$$

- Formulation using indicator random variables z_{ij}:

Size regularizer

$$\min_{\{z_{ij}\}} \lambda \sum_{i\in\mathcal{D}} \mathrm{I}\left(\left\|\begin{bmatrix} z_{i1} & z_{i2} & \cdots \end{bmatrix}\right\|_p\right) + \sum_{j\in\mathcal{D}}\sum_{i\in\mathcal{D}} d_{ij} z_i$$

$$\text{s. t.} \quad z_{ij} \in \{0,1\}, \ \sum_{i=1}^{N} z_{ij} = 1, \ \forall\, i,j \in \mathcal{D}.$$

- Convex relaxation:

$$0 \leq z_{ij} \leq 1$$

Online Subset Selection

- At time t:
  R_{t-1}: old set (denoted by superscript o)
  X_t: in the new set (denoted by superscript n)
  R_t: the new reduced set that we are trying to compute using z_{ij}
  $$R_t = R_{t-1} \cup \{i \in X_t | Z_{ij} = 1\}$$

- Revised formulation:

$$J'_{\text{enc}} \triangleq \sum_{i \in \mathcal{E}_o} \sum_{j \in \mathcal{D}_n} d_{ij}^{o,n} z_{ij}^{o,n} + \sum_{i \in \mathcal{D}_n} \sum_{j \in \mathcal{D}_n} d_{ij}^{n,n} z_{ij}^{n,n},$$

$$\begin{aligned} \min_{\mathcal{Z}'} \ & J'_{\text{enc}} + \lambda \sum_{i \in \mathcal{D}_n} \mathrm{I}(\left\| \begin{bmatrix} z_{i1}^{n,n} & z_{i2}^{n,n} & \cdots \end{bmatrix} \right\|_p) \\ \text{s.t.} \ & z_{ij}^{o,n},\ z_{ij}^{n,n} \in \{0,1\},\ \forall i,\ j, \\ & \sum_{i \in \mathcal{E}_o} z_{ij}^{o,n} + \sum_{i \in \mathcal{D}_n} z_{ij}^{n,n} = 1,\ \forall\, j \in \mathcal{D}_n, \end{aligned}$$

$$\boxed{\begin{gathered} \epsilon_o = R_{t-1} \\ D_n = X_t \end{gathered}}$$

High Level Idea

- Given a compression ratio, find out representatives which have the least dissimilarity with the left-out elements besides having the highest task-specific loss.
- Highest task-specific loss ensures having situational tasks needed to be learnt more by the model.

TMCOSS

Adopts a facility location objective involving multiple criteria

$$\min_{z^o_{ij}, z^n_{ij}} \mathscr{G}(z^o_{ij}, z^n_{ij})\, s.t. \sum_{j=1}^{|R_t|} z^o_{i,j} + \sum_{j=1}^{m} z^n_{i,j} = 1;\; z^n_{i,j}, z^o_{i,j} \in [0,1];\; \sum_{j=1}^{m} \|[z^n_{1,j} \ldots z^n_{m,j}]\|_p \leq frac * m$$

Objective function

Constraint 1

Constraint 2

Compression Ratio

$z^o_{ij} = 1$ Denotes j from existing set o is a representative of element i from incoming set n

$z^n_{ij} = 1$ Denotes j from incoming set n is a representative of element i from incoming set n

$$\mathscr{G}(z^o_{ij}, z^n_{ij}) = \rho\underbrace{\Big(\sum_{i=1}^{m}\sum_{j=1}^{|R_t|} z^o_{ij} d^o_{ij}(t) + \sum_{i,j=1}^{m} z^n_{ij} d^n_{ij}(t)\Big)}_{\text{Dissimilarity}} - (1-\rho)\underbrace{\Big(\sum_{j=1}^{|R_t|} S^o_j * L^o_j + \sum_{j=1}^{m} S^n_j * L^n_j\Big)}_{\text{Task specific Loss}}\ where,\ \underbrace{S^o_j = \frac{1}{\epsilon}\min(\epsilon, \sum_{i=1}^{m} z^o_{ij}),\ S^n_j = \frac{1}{\epsilon}\min(\epsilon, \sum_{i=1}^{m} z^n_{ij})}_{\text{Representative power of element j thresholded by } \epsilon}$$

Justification for thresholding

Theorem 1 *Let z^o_{ij} and z^n_{ij} be the optimal solution for formulation 1. A new frame $j \in X_{t+1}$ is selected as a representative frame for at least one incoming frame $i \in X_{t+1}$, i.e. $z^n_{ij} = 1$, only if BOTH these conditions hold:*

- *For some incoming frame $i \in X_{t+1}$, $Q^n_{ij} < Q^n_{ij'}$, for all $j' \in X_{t+1}$ and $j' \neq j$*
- *For some incoming frame $i \in X_{t+1}$, $Q^n_{ij} < \frac{\sum_{i'=1}^{m} z^o_{i',k} Q^o_{i'k} + \lambda \|[z^n_{1,j} \dots z^n_{m,j}]\|_p}{\|\mathbf{z}^n_j\|_1}$*

where $k = argmin_j \sum_{i=1}^{m} z^o_{i,j} Q^o_{i,j}$, and $\|\mathbf{z}^n_j\|_1 = \sum_{i'=1}^{m} z^n_{i'j}$

$\rho = 0$

Corollary 1.1 *Let z^o_{ij} and z^n_{ij} be the optimal solution for formulation 1. A new frame $j \in X_{t+1}$ is selected as a representative frame for at least one incoming frame $i \in X_{t+1}$, i.e. $z^n_{ij} = 1$, only if BOTH these conditions hold:*

- *$L^n_j > L^n_{j'}$ for all $j' \in X_{t+1}$ and $j' \neq j$*
- *$L^n_j > \frac{\sum_{i=1}^{m} z^o_{i,k} L^o_k - \lambda \|[z^n_{1,j} \dots z^n_{m,j}]\|_p}{\|\mathbf{z}^n_j\|_1}$*

where $k = argmin_j \sum_{i=1}^{m} z^o_{i,j} Q^o_{i,j}$, and $\|\mathbf{z}^n_j\|_1 = \sum_{i'=1}^{m} z^n_{i'j}$

Multi-criteria OSS (MCOSS)[1]

$$Q^n_{ij} = \rho d^n_{ij} - (1-\rho) L^n_j ; Q^o_{ij} = \rho d^o_{ij} - (1-\rho) L^o_j$$

$$\min_{z^o_{ij}, z^n_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{|R_t|} z^o_{ij} Q^o_{ij} + \sum_{i,j=1}^{m} z^n_{ij} Q^n_{ij} + \lambda \sum_{j=1}^{m} \|[z^n_{1,j} \dots z^n_{m,j}]\|_p$$

$$s.t. \sum_{j=1}^{|R_t|} z^o_{i,j} + \sum_{j=1}^{m} z^n_{i,j} = 1, \forall i \in X_{t+1} z^n_{i,j}, z^o_{i,j} \in [0,1], \forall i,j$$

1. Soumi Das, Sayan Mondal, Ashwin Bhoyar, Madhumita Bharde, Niloy Ganguly, Suparna Bhattacharya, Sourangshu Bhattacharya, "Multi-criteria onlineframe-subset selection for autonomous vehicle videos." *Pattern Recognition Letters* 133 (2020): 349-355.

References

Submodular Optimization

1. Andreas Krause and Daniel Golovin. Submodular Function Maximization. 2012.
2. IJCAI 2021 tutorial by Rishabh Iyer and Ganesh Ramakrishnan
3. Buchbinder, Niv, et al. Submodular maximization with cardinality constraints. SODA 2014.
4. Mirzasoleiman, Baharan, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. ICML 2020.

Orthogonal Matching Pursuit

1. Killamsetty, Krishnateja, S. Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. ICML 2021.
2. Tropp, Joel A., and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on information theory, (2007).
3. Cai, T. Tony, and Lie Wang. "Orthogonal matching pursuit for sparse signal recovery with noise." IEEE Transactions on Information theory, (2011).

Convex Optimization

1. Elhamifar, Ehsan, and M. Clara De Paolis Kaluza. Online summarization via submodular and convex optimization. CVPR 2017.
2. Das, Soumi, Harikrishna Patibandla, Suparna Bhattacharya, Kshounis Bera, Niloy Ganguly, and Sourangshu Bhattacharya. "TMCOSS: Thresholded Multi-Criteria Online Subset Selection for Data-Efficient Autonomous Driving." CVPR 2021.