Submodularity in Speech/NLP

Jeffrey A. Bilmes

Professor
Departments of Electrical Engineering
& Computer Science and Engineering
University of Washington, Seattle
http://melodi.ee.washington.edu/~bilmes

Wednesday, June 11th, 2014

Outline

1. Submodularity: Generalized Independence or Complexity
2. Document Summarization
3. Data Summarization
   - Speech Summarization
   - Selection in Statistical Machine Translation
   - Image Summarization
4. End

Acknowledgments

Joint work with: Hui Lin, Kai Wei, Yuzong Liu, Katrin Kirchhoff, Sebastian Tschiatschek, and Rishabh Iyer

Outline

1. Submodularity: Generalized Independence or Complexity
2. Document Summarization
3. Data Summarization
   - Speech Summarization
   - Selection in Statistical Machine Translation
   - Image Summarization
4. End

Two Equivalent Submodular Definitions

Definition (submodular)

A function $f : 2^V \rightarrow \mathbb{R}$ is submodular if for any $A, B \subseteq V$, we have that:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{1}$$

Definition (submodular (diminishing returns))

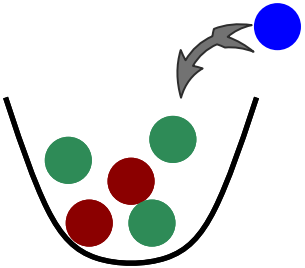
A function $f : 2^V \rightarrow \mathbb{R}$ is submodular if for any $A \subseteq B \subset V$, and $v \in V \setminus B$, we have that:

$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B) \tag{2}$$

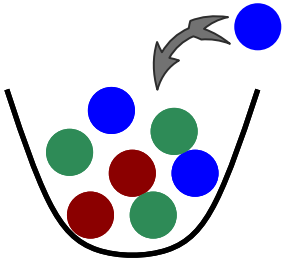
This means that the incremental "value", "gain", or "cost" of v decreases (diminishes) as the context in which v is considered grows from A to B.

Example Submodular: Number of Colors of Balls in Urns

- Consider an urn containing colored balls. Given a set S of balls, $f(S)$ counts the number of distinct colors.



Initial value: 2 (colors in urn).
New value with added blue ball: 3



Initial value: 3 (colors in urn).
New value with added blue ball: 3

- Submodularity: Incremental Value of Object Diminishes in a Larger Context (diminishing returns). Thus, f is submodular.

Discrete Optimization

- We are given a finite set of objects V of size $n = |V|$.

Discrete Optimization

- We are given a finite set of objects V of size $n = |V|$.
- There are 2^n such subsets (denoted 2^V) of the form $A \subseteq V$.

Discrete Optimization

- We are given a finite set of objects V of size $n = |V|$.
- There are 2^n such subsets (denoted 2^V) of the form $A \subseteq V$.
- We have a function $f : 2^V \to \mathbb{R}$ that judges the quality (or value, or cost, or etc.) of each subset. $f(A) =$ some real number.

Discrete Optimization

- We are given a finite set of objects V of size $n = |V|$.
- There are 2^n such subsets (denoted 2^V) of the form $A \subseteq V$.
- We have a function $f : 2^V \rightarrow \mathbb{R}$ that judges the quality (or value, or cost, or etc.) of each subset. $f(A) =$ some real number.
- Unconstrained minimization & maximization:

$$\min_{X \subseteq V} f(X) \quad (3) \qquad \max_{X \subseteq V} f(X) \quad (4)$$

Discrete Optimization

- We are given a finite set of objects V of size $n = |V|$.
- There are 2^n such subsets (denoted 2^V) of the form $A \subseteq V$.
- We have a function $f : 2^V \rightarrow \mathbb{R}$ that judges the quality (or value, or cost, or etc.) of each subset. $f(A) =$ some real number.
- Unconstrained minimization & maximization:

$$\min_{X \subseteq V} f(X) \quad (3) \qquad \max_{X \subseteq V} f(X) \quad (4)$$

- Without knowing anything about f, it takes 2^n queries to be able to offer any quality assurance on a candidate solution. Otherwise, solution can be unboundedly poor.

Discrete Optimization

- We are given a finite set of objects V of size $n = |V|$.
- There are 2^n such subsets (denoted 2^V) of the form $A \subseteq V$.
- We have a function $f : 2^V \rightarrow \mathbb{R}$ that judges the quality (or value, or cost, or etc.) of each subset. $f(A) =$ some real number.
- Unconstrained minimization & maximization:

$$\min_{X \subseteq V} f(X) \quad (3) \qquad \max_{X \subseteq V} f(X) \quad (4)$$

- Without knowing anything about f, it takes 2^n queries to be able to offer any quality assurance on a candidate solution. Otherwise, solution can be unboundedly poor.
- When f is submodular, Eq. (3) is polytime, and Eq. (4) is constant-factor approximable.

Constrained Discrete Optimization

- Often, we are interested only in a subset of the set of possible subsets, namely $\mathcal{S} \subseteq 2^V$.

Constrained Discrete Optimization

- Often, we are interested only in a subset of the set of possible subsets, namely $\mathcal{S} \subseteq 2^V$.
- Example: only sets having bounded size $\mathcal{S} = \{S \subseteq V : |S| \leq k\}$ or within a budget $\left\{S \subseteq V : \sum_{s \in S} w(s) \leq b\right\}$ where w is a cost vector.

Constrained Discrete Optimization

- Often, we are interested only in a subset of the set of possible subsets, namely $\mathcal{S} \subseteq 2^V$.
- Example: only sets having bounded size $\mathcal{S} = \{S \subseteq V : |S| \leq k\}$ or within a budget $\left\{S \subseteq V : \sum_{s \in S} w(s) \leq b\right\}$ where w is a cost vector.
- Example: sets could correspond to combinatorial object (i.e., feasible $\mathcal{S}$ might be trees, matchings, paths, vertex covers, or cuts).

Constrained Discrete Optimization

- Often, we are interested only in a subset of the set of possible subsets, namely $\mathcal{S} \subseteq 2^V$.
- Example: only sets having bounded size $\mathcal{S} = \{S \subseteq V : |S| \leq k\}$ or within a budget $\left\{S \subseteq V : \sum_{s \in S} w(s) \leq b\right\}$ where w is a cost vector.
- Example: sets could correspond to combinatorial object (i.e., feasible $\mathcal{S}$ might be trees, matchings, paths, vertex covers, or cuts).
- Ex: $\mathcal{S}$ might be a function of some g (e.g., sub-level sets of g, $\mathcal{S} = \{S \subseteq V : g(S) \leq \alpha\}$, sup-level sets $\mathcal{S} = \{S \subseteq V : g(S) \geq \alpha\}$).

Constrained Discrete Optimization

- Often, we are interested only in a subset of the set of possible subsets, namely $\mathcal{S} \subseteq 2^V$.
- Example: only sets having bounded size $\mathcal{S} = \{S \subseteq V : |S| \leq k\}$ or within a budget $\left\{S \subseteq V : \sum_{s \in S} w(s) \leq b\right\}$ where w is a cost vector.
- Example: sets could correspond to combinatorial object (i.e., feasible $\mathcal{S}$ might be trees, matchings, paths, vertex covers, or cuts).
- Ex: $\mathcal{S}$ might be a function of some g (e.g., sub-level sets of g, $\mathcal{S} = \{S \subseteq V : g(S) \leq \alpha\}$, sup-level sets $\mathcal{S} = \{S \subseteq V : g(S) \geq \alpha\}$).
- Constrained discrete optimization problems:

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{S} \end{array} \quad (5) \qquad \begin{array}{ll} \text{minimize} & f(S) \\ \text{subject to} & S \in \mathcal{S} \end{array} \quad (6)$$

Constrained Discrete Optimization

- Often, we are interested only in a subset of the set of possible subsets, namely $\mathcal{S} \subseteq 2^V$.
- Example: only sets having bounded size $\mathcal{S} = \{S \subseteq V : |S| \leq k\}$ or within a budget $\left\{S \subseteq V : \sum_{s \in S} w(s) \leq b\right\}$ where w is a cost vector.
- Example: sets could correspond to combinatorial object (i.e., feasible $\mathcal{S}$ might be trees, matchings, paths, vertex covers, or cuts).
- Ex: $\mathcal{S}$ might be a function of some g (e.g., sub-level sets of g, $\mathcal{S} = \{S \subseteq V : g(S) \leq \alpha\}$, sup-level sets $\mathcal{S} = \{S \subseteq V : g(S) \geq \alpha\}$).
- Constrained discrete optimization problems:

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{S} \end{array} \quad (5) \qquad \begin{array}{ll} \text{minimize} & f(S) \\ \text{subject to} & S \in \mathcal{S} \end{array} \quad (6)$$

- Fortunately, when f (and g) are submodular, solving these problems can often be done with guarantees (and often efficiently)!

Where is submodularity useful as a model?

- Useful as a model of a physical process. Meaning of the value depends on if we either wish to maximize or minimize it.

Where is submodularity useful as a model?

- Useful as a model of a physical process. Meaning of the value depends on if we either wish to maximize or minimize it.
- For maximization: diversity, coverage, span, and information.

Where is submodularity useful as a model?

- Useful as a model of a physical process. Meaning of the value depends on if we either wish to maximize or minimize it.
- For maximization: diversity, coverage, span, and information.
- For minimization: cooperative costs, complexity, roughness, and irregularity.

Where is submodularity useful as a model?

- Useful as a model of a physical process. Meaning of the value depends on if we either wish to maximize or minimize it.
- For maximization: diversity, coverage, span, and information.
- For minimization: cooperative costs, complexity, roughness, and irregularity.
- In speech/text/NLP, there are many instances of problems that are inherently discrete.

Generalized information/complexity functions

- Entropy, given a joint distribution $p(x_V)$ over $|V|$ random variables:

$$f(A) = H(X_A) = -\sum_{x_A} p(x_A) \log p(x_A) \tag{7}$$

with $p(x_V)$ joint probability distribution over X_V.

Generalized information/complexity functions

- Entropy, given a joint distribution $p(x_V)$ over $|V|$ random variables:

$$f(A) = H(X_A) = -\sum_{x_A} p(x_A) \log p(x_A) \tag{7}$$

  with $p(x_V)$ joint probability distribution over X_V.

- Many other functions are submodular. E.g., set cover, graph cut, bipartite neighborhoods, sums of weighted concave composed with additive functions, matroid rank, etc.

Generalized information/complexity functions

- Entropy, given a joint distribution $p(x_V)$ over $|V|$ random variables:

$$f(A) = H(X_A) = -\sum_{x_A} p(x_A) \log p(x_A) \tag{7}$$

  with $p(x_V)$ joint probability distribution over X_V.
- Many other functions are submodular. E.g., set cover, graph cut, bipartite neighborhoods, sums of weighted concave composed with additive functions, matroid rank, etc.
- All submodular functions express a form of "abstract independence" or "generalized complexity"

Generalized Information

- Given submodular f, there a notion of "independence" , i.e., $A \perp\!\!\!\perp B$:

$$f(A \cup B) = f(A) + f(B), \tag{8}$$

- and a notion of "conditional independence" , i.e., $A \perp\!\!\!\perp B | C$:

$$f(A \cup B \cup C) + f(C) = f(A \cup C) + f(B \cup C) \tag{9}$$

- and a notion of "dependence" (conditioning reduces valuation):

$$f(A|B) \triangleq f(A \cup B) - f(B) < f(A), \tag{10}$$

- and a notion of "conditional mutual information"

$$I_f(A; B|C) \triangleq f(A \cup C) + f(B \cup C) - f(A \cup B \cup C) - f(C) \geq 0$$

- and notions of "information amongst a collection of sets", e.g.:

$$I_f(S_1; S_2; \ldots; S_k) = \sum_{i=1}^{k} f(S_k) - f(S_1 \cup S_2 \cup \cdots \cup S_k) \tag{11}$$

Submodular Approaches to Big Data Summarization

- We are given a set indexed by V
- Approach: 1) find a good function $f : 2^V \rightarrow \mathbb{R}_+$ that represents information in V. 2) Then optimize f to obtain a subset.

1) Heuristic: design f by hand, hoping that f is a good proxy for the information within V. Acknowledge that f is a surrogate objective, guarantees are only in terms of f.
2) More promising approach: attempt to learn f, or some aspects of a good f, in some fashion based on training data.

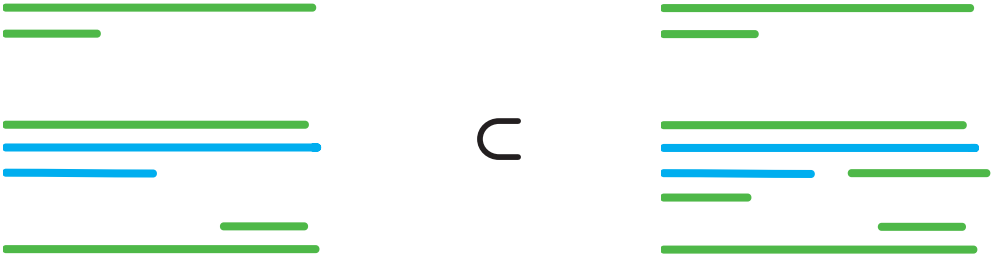
- We report on both kinds of results for document summarization, speech training data subset selection, subset selection in statistical machine translation problems, and image summarization.

Outline

1. Submodularity: Generalized Independence or Complexity
2. Document Summarization
3. Data Summarization
   - Speech Summarization
   - Selection in Statistical Machine Translation
   - Image Summarization
4. End

Extractive Document Summarization

- We extract sentences (green) as a summary of the full document



- The summary on the left is a subset of the summary on the right.
- Consider adding a new (blue) sentence to each of the two summaries.
- Marginal benefit of adding the new (blue) sentence to the smaller (left) summary is no less than the marginal benefit of adding blue sentence to the larger (right) summary.

Submodularity for document summarization?

As further evidence for submodularity's appropriateness, $\exists$ many instances of submodularity's use in the NLP community, originally unbeknownst to the authors.

- E.g., maximum marginal relevance (MMR) (Carbonell & Goldstein, 1998) has a diminishing returns property.
- Modified MMR - (McDonald, 2007)
- Concept-based approaches (Filatova & Hatzivassiloglou 2004; Takamura & Okumura, 2009; Riedhammer et al., 2010; Qazvinian et al., 2010).
- Two standard methods for automatic evaluation of candidate summarizes are submodular, including ROUGE-N (Lin 2004, described on next slide) and Pyramid (Nenkova & Passonneau, 2004).
- Both ROUGE-N and Pyramid are parameterized by good quality summarizes produced by humans, used only for evaluation.

NIST's ROUGE-N (Lin-04) evaluation function

`http://www.nist.gov/tac/2011/Summarization`: NIST's ROUGE-N recall score, a widely used standard, turns out to be submodular:

$$f_{\text{ROUGE-N}}(S) \triangleq \frac{\sum_{i=1}^{K} \sum_{e \in R_i} \min(c_e(S), r_{e,i})}{\sum_{i=1}^{K} \sum_{e \in R_i} r_{e,i}},$$

- S is the candidate summary (set of sentences extracted from the ground set V)
- $c_e : 2^V \to \mathbb{Z}_+$ is the number of times an n-gram e occurs in summary S, clearly a modular function for each e.
- R_i is the set of n-grams contained in the reference summary i (given K reference summaries).
- and $r_{e,i}$ is the number of times n-gram e occurs in reference summary i.
- ROUGE is based on a collection of human generated summaries, so the measure can be only used to evaluate a summary.

Coverage function

Coverage Function

$$\mathcal{L}(S) = \sum_{i \in V} \min \{\mathcal{C}_i(S), \alpha\, \mathcal{C}_i(V)\}$$

- $\mathcal{C}_i$ measures how well i is covered by S.
- One simple possible $\mathcal{C}_i$ (that we use) is:

$$\mathcal{C}_i(S) = \sum_{j \in S} w_{i,j},$$

  where $w_{i,j} \geq 0$ measures the similarity between i and j.
- With this C_i, $\mathcal{L}(S)$ is monotone submodular, as required.

Diversity reward function

Diversity Reward Function

$$\mathcal{R}(S) = \sum_{i=1}^{K} \sqrt{\sum_{j \in P_i \cap S} r_j}.$$

- $P_i, i = 1, \cdots K$ is a partition of the ground set V
- $r_j \geq 0$: **singleton reward** of j, which represents the importance of j to the summary.
- square root over the sum of rewards of sentences belong to the same partition (diminishing returns).
- $\mathcal{R}(S)$ is monotone submodular as well.

Diversity Reward Function Mixtures

Alternatively, we can utilize multiple partitions/clusterings, produce a diversity reward function for each one, and mix them together.

Multi-resolution Diversity Reward

$$\mathcal{R}(S) = \lambda_1 \sum_{i=1}^{K_1} \sqrt{\sum_{j \in P_i^{(1)} \cap S} r_j} + \lambda_2 \sum_{i=1}^{K_2} \sqrt{\sum_{j \in P_i^{(2)} \cap S} r_j} + \cdots$$

Structured Prediction: Approach with inference

- Constraints specified in inference form:

$$\underset{\mathbf{w}\geq 0,\xi_t}{\text{minimize}} \quad \frac{1}{T}\sum_t \xi_t + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{12}$$

$$\text{subject to} \quad \mathbf{w}^\top \mathbf{f}_t(\mathbf{y}^{(t)}) \geq \max_{\mathbf{y}\in\mathcal{Y}_t} \left(\mathbf{w}^\top \mathbf{f}_t(\mathbf{y}) + \ell_t(\mathbf{y})\right) - \xi_t, \forall t \tag{13}$$

$$\xi_t \geq 0, \forall t. \tag{14}$$

- Exponential set of constraints reduced to an embedded optimization problem, "inference."

Learning Submodular Mixtures: Unconstrained Form

- Unconstrained form with hinge-loss

$$\min_{\mathbf{w}\geq 0} \frac{1}{T} \sum_t \left[\max_{\mathbf{y}\in\mathcal{Y}_t} \left(\mathbf{w}^\top \mathbf{f}_t(\mathbf{y}) + \ell_t(\mathbf{y}) \right) - \mathbf{w}^\top \mathbf{f}_t(\mathbf{y}^{(t)}) \right] + \frac{\lambda}{2} \|\mathbf{w}\|^2 \tag{15}$$

- Subgradient approach: To compute a subgradient, we must solve the following embedded optimization problem

$$\max_{\mathbf{y}\in\mathcal{Y}_t} \left(\mathbf{w}^\top \mathbf{f}_t(\mathbf{y}) + \ell_t(\mathbf{y}) \right) \tag{16}$$

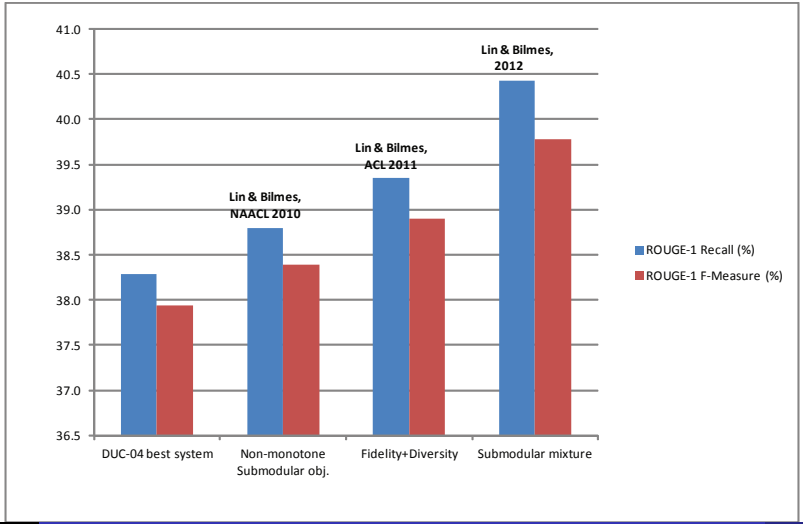
- Convex in $\mathbf{w}$, and $\mathbf{w}^\top \mathbf{f}_t(\mathbf{y})$ presumably submodular, but what about $\ell_t(\mathbf{y})$?
- Often one uses Hamming loss (in general structured prediction problems), but here we ask that that $\ell_t(\mathbf{y})$ is also submodular.

DUC Evaluations

- DUC (Document Understanding Conference) data `http://duc.nist.gov/`
- Standard Evaluation of extractive document summarization managed by NIST in the years 2004-2007.
- Tasks are both query independent (DUC '04) and query dependent summarization (DUC '05-'07), which is more like web search.
- Standard measure of evaluation performance is the ROUGE measure.

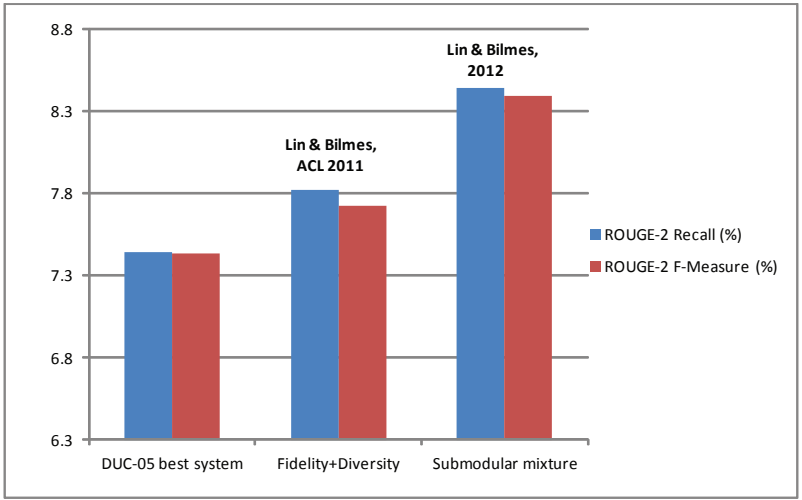
DUC-04 Results

Rouge-1: higher is better



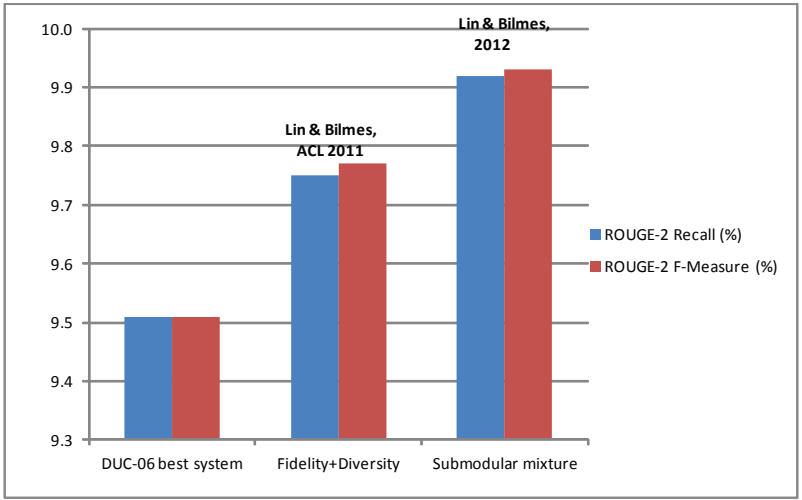
DUC-05 Results

Rouge-2: higher is better



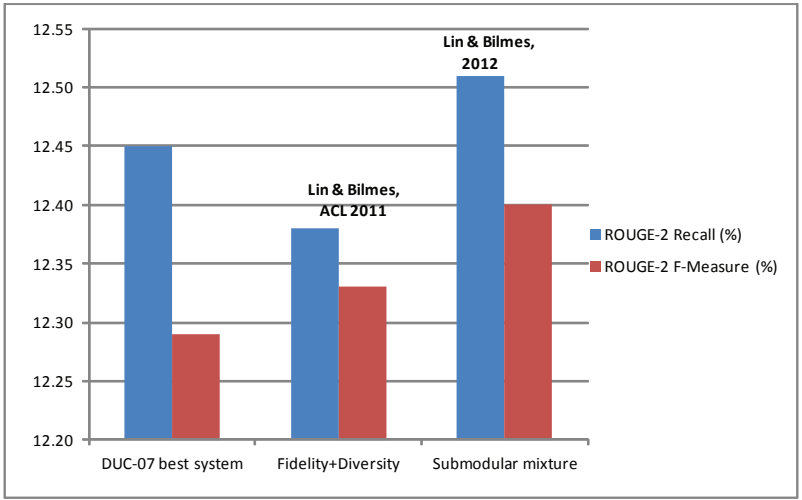
DUC-06 Results

Rouge-2: higher is better



DUC-07 Results

Rouge-2: higher is better



Outline

1. Submodularity: Generalized Independence or Complexity
2. Document Summarization
3. Data Summarization
   - Speech Summarization
   - Selection in Statistical Machine Translation
   - Image Summarization
4. End

As the data set size grow . . .

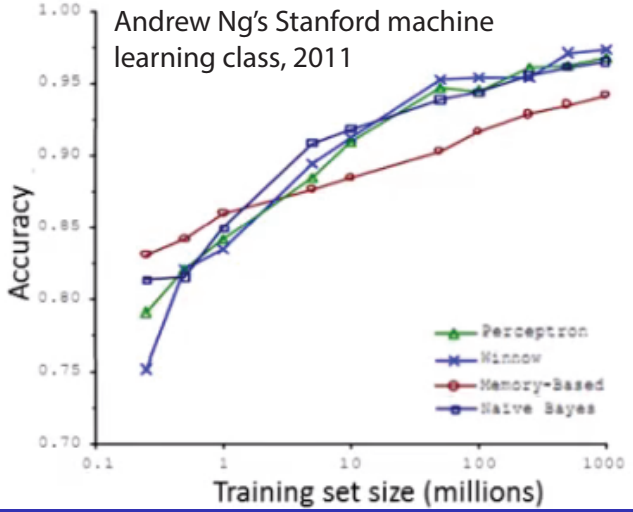
There is no data like more data

As the data set size grow . . .

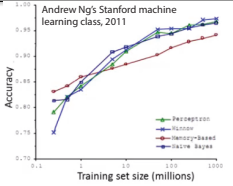
There is no data like more data $\Rightarrow$ rather, more data is like no data.

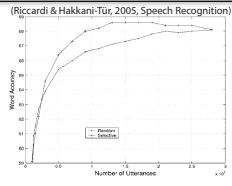
As the data set size grow . . .

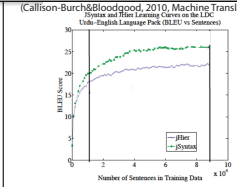
There is no data like more data $\Rightarrow$ rather, more data is like no data.

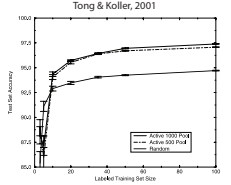


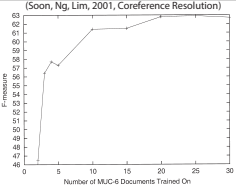
As the data set sizes grow . . .

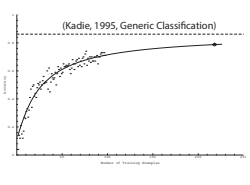


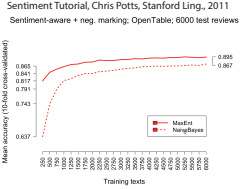


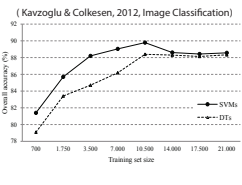


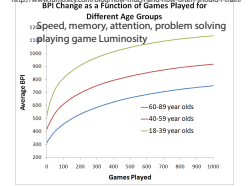












Submodularity and Learning Curves

Diminishing Returns: the more you have, the less valuable is anything you don't have. Bad for complex machine learning systems (e.g., deep models, SVMs).

Submodularity and Learning Curves

Diminishing Returns: the more you have, the less valuable is anything you don't have. Bad for complex machine learning systems (e.g., deep models, SVMs).

Proposition

Let $V = \{1, 2, \ldots, n\}$ be a finite ground set, and let $f : 2^V \to \mathbb{R}$ be a set function. If for all permutations σ of V, we have that for all $i \leq j$:

$$f(\sigma_j | S_{i-1}) \geq f(\sigma_j | S_{j-1}) \tag{17}$$

with $S_i = \{\sigma_1, \sigma_2, \ldots, \sigma_i\}$, then f is submodular.

- Learning curves might not be exactly submodular, but submodularity seems a reasonable model.

Practical Goals: Submodular Proxies

- Key question: Can statistical predictions be cost effective using small data?

Outline

1 Submodularity: Generalized Independence or Complexity

2 Document Summarization

3 Data Summarization
- Speech Summarization
- Selection in Statistical Machine Translation
- Image Summarization

4 End

Speech Subset Selection: Two Forms

- Corpus Summarization: Given a large set of speech utterances (training corpus) $V = \{v_1, v_2, \ldots, v_n\}$, choose a small subset $A \subseteq V$ that is representative of V.
- Goal: training on summary should yield highest accuracy possible.
- In this work, concentrate on drastic reductions in training set (one to two orders magnitude) to reduce model design iteration cycle time.

Corpus Summarization: motivation

- Large vocabulary speech recognition training is both resource (disk, memory) and time consuming.
- Particularly acute with recent models (e.g., Deep Neural Networks) that can take appreciable time to train.
- Training data can be redundant: Why waste time/resources training on information you already know?
- Switchboard, Switchboard Cellular, & Fisher: very large vocabulary, fluent spontaneous speech, much more difficult, state of the art speech recognition system, supervised labels.
- Probably more true as the data set sizes grow.

Submodular Switchboard selection: GMM and DNN

	1%	5%	10%	20%	all
Rand	52.1± 1.5	38.2±0.2	35.1±0.3	34.4±0.2	31.0
HE (words)	49.6	36.5	34.8	N/A	
HE (3-phones)	47.5	37.6	34.2	N/A	
SM (3-phones)	**47.5**	**35.7**	**33.3**	**32.6**	

Table : Word error rates, random (Rand), histogram-entropy (HE), the submodular (SM) system. Histogram-entropy results saturate after 10%.

	1%	5%	10%	20%	all
Rand	43.7±0.5	34.3±0.9	31.5±0.5	29.8±0.2	26.0
HE (3-phones)	42.8	33.9	31.3	N/A	
SM (3-phones)	**41.1**	**31.8**	**29.3**	**28.2**	

Table : Word error rates for DNN system.

Outline

1 Submodularity: Generalized Independence or Complexity

2 Document Summarization

3 Data Summarization
- Speech Summarization
- Selection in Statistical Machine Translation
- Image Summarization

4 End

Data subset selection for machine translation

- Statistical Machine Translation (SMT): automatically translate from one human language to another.
- Common problems in SMT: 1) test data is from a target domain while training data is mixed-domain; 2) phrase translation table, when based on all training data, can be massive.
- Solution: choose and then train using only a (domain-specific) subset of training data.
- Many previous approaches (e.g., *n*-gram overlap (Ittycheriah & Roukos, 2007), coverage of unseen *n*-grams (Eck et al. 2005), feature decay approach (Biçici & Yuret, 2011-2013)) are inadvertently submodular.
- Some (e.g., Moore & Lewis, 2010) are only modular.
- We approach directly using submodular functions.

Feature based submodular functions

- V is ground set of data items (sentences) and U is a set of features, n-grams in our work (but could be parse-based or deep features as well).

Feature based submodular functions

- V is ground set of data items (sentences) and U is a set of features, n-grams in our work (but could be parse-based or deep features as well).
- Feature-based submodular functions:

$$f(X) = \sum_{u \in U} w_u \phi_u(m_u(X)) \tag{18}$$

where $w_u > 0$ is a feature weight, $m_u(X) = \sum_{x \in X} m_u(x)$ is a non-negative modular function specific to feature u, $m_u(x)$ is a relevance score, a non-negative scalar score indicating the relevance of feature u in object x, and ϕ_u is a u-specific non-negative non-decreasing concave function.

Results: NIST 2009 Arabic → English

Method	Data Subset Sizes			
	10%	20%	30%	40%
Rand	0.3991 (± 0.004)	0.4142 (± 0.003)	0.4205 (± 0.002)	0.4220 (± 0.002)
Xent	0.4235 (± 0.004)	0.4292 (± 0.002)	0.4290 (± 0.003)	0.4292 (± 0.001)
SM-1	**0.4313**	0.4345	0.4333	0.4351
SM-2	**0.4335**	**0.4380**	0.4328	**0.4365**
SM-3	0.4306	0.4319	**0.4374**	0.4319
SM-4	**0.4313**	0.4345	0.4333	0.4351
SM-5	0.4286	**0.4364**	0.4327	0.4328
SM-6	**0.4356***	**0.4359**	**0.4384***	**0.4366**
100%	0.4257			

BLEU on NIST 2009 test set for random (Rand), cross-entropy (Xent), and various submodular (SM) data selection methods. Bold = significant over best Xent system.

Outline

1. Submodularity: Generalized Independence or Complexity
2. Document Summarization
3. Data Summarization
   - Speech Summarization
   - Selection in Statistical Machine Translation
   - Image Summarization
4. End

Image collections

Many images, also that have a higher level gestalt than just a few.

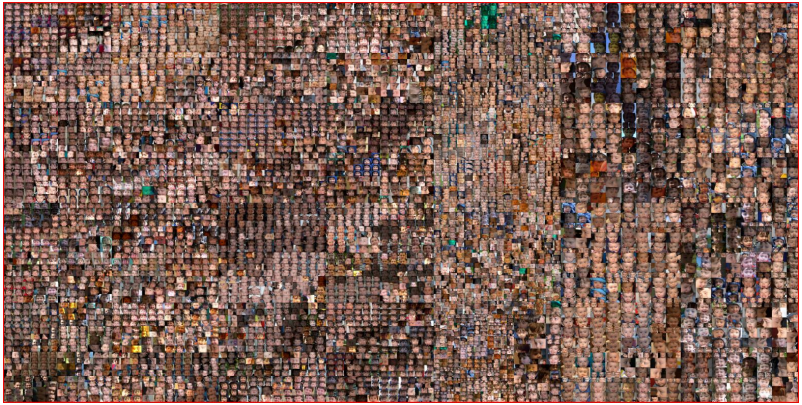


Image Summarization

Task: Summarize collection of images by representative subset of the images

Applications:

- Summarizing your holiday pictures.
- Summarizing image search results
- Efficient browsing of image collections
- Video frame summarization

Image Summarization - Data Collection

Data Statistics

- 14 image collections with 100 pictures each
- $\sim$ 400 human summaries for every image collection, via Amazon Turk, about 5500 summaries total!

Example collections:

Image Summarization

Super-Pixel Based V-Rouge

Whole collection:

3 best summaries:

3 medium summaries:

3 worst summaries:

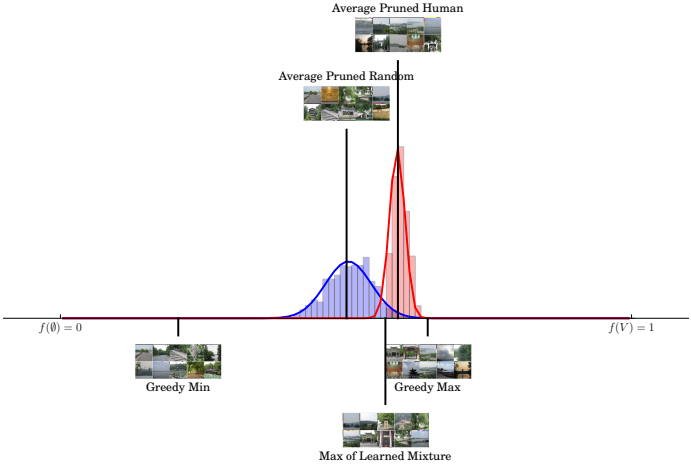
Image Summarization

Learning to Summarize

- Define submodular functions for measuring
  - *coverage* (how well a subset of images covers all images),
  - *diversity* (how different is a given subset of images),
  - *feature functions* (how present are certain visual words),
- Learn large-margin mixture of these functions on 13 out of 14 image collections and test on held out image collection

Image Summarization

Early Results - Learnt mixture using Max-Margin



Other Applications of Submodularity in NLP

- Alignment between two bi-text strings e and f via submodular maximization subject to matroid constraints (generalizing bipartite matching). Lin & Bilmes, NAACL/HLT 2011.

Other Applications of Submodularity in NLP

- Alignment between two bi-text strings e and f via submodular maximization subject to matroid constraints (generalizing bipartite matching). Lin & Bilmes, NAACL/HLT 2011.
- Balanced clustering via symmetric submodular functions. Application to word clustering with bipartite neighborhood functions (a words neighbors are its features, which is the contexts in which the word occurs). Used for constructed class-based or factored language models. Narasimhan & Bilmes, IJCAI 2007.

Recent Tutorials on Submodularity

Deep Mathematical Properties of Submodularity
with Applications to Machine Learning
a tutorial at NIPS 2013
http://nips.cc/Conferences/2013/Program/event.php?ID=3688

MLSS Machine Learning Summer School

Mathematical Properties of Submodularity
with Applications to Machine Learning
a tutorial at MLSS 2014
http://mlss2014.hiit.fi/