Submodularity in Data Subset Selection and Active Learning

Kai Wei  Rishabh Iyer  Jeff Bilmes

MELODI Lab, University of Washington, Seattle

Presenter: Kai Wei
Outline

1. Background

2. Submodular Data Subset Selection for ML Classifiers

3. Extension to Active Learning

4. Experiments
What is data subset selection?

**Problem Scenario:**

Large Data Set → Subset Selection → Evaluation

- \( V \): large set of data items;
- Select a small sized subset \( S \subseteq V \);
- Goal: choose subset \( S \) to attain high utility for an underlying task, relative to other sets of similar size.

**In this work:**

- Study data subset selection for reduced-cost training of ML classifiers by choosing a good subset of a training set.
Motivations

- Training data can be fairly redundant;

  1. all_right how are_you doing
  2. how are_you with yours
  3. hi nadine my name is lorraine how are_you
  4. good how are_you
  5. hello hi how are_you
  6. good thanks how are_you
  7. uh how are_you
  8. i'm good how are_you
  9. fine how are_you

- Reduce training and, hence, experimental turnaround time;
  - Ex: state-of-the-art deep models require enormous computation to train — wasteful when there is redundancy in training data.

- Reduce labeling cost.
  - Active learning.
Problem Formulation: Discrete Optimization

\[
\max_{S \subseteq C} f(S)
\]

- \(f : 2^V \rightarrow \mathbb{R}\) represents the utility of each subset \(S \subseteq V\);
- \(C\): constraint on the selected subset.
  - e.g. size constraint, knapsack constraint, matroid constraint, etc.
- Optimization is provably exponential cost in general, even to approximate.

Fortunately

- Becomes easy to approximate when \(f\) is submodular.
Submodular Set Functions

- A class of set function $f : 2^V \rightarrow \mathbb{R}$ that satisfies diminishing returns:

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

- Example: $f$ gives number of colors of the balls in an urn.

- Modular function $f(X) = \sum_{i \in X} f(i)$ analogous to linear functions.
Data Subset Selection for Naïve Bayes Classifiers

- **Goal:** Select a small sized data set to train a good NB classifier.

- **Utility function:** Log-likelihood function $\ell^\text{NB}(S)$.

  $$\ell^\text{NB}(S) = \sum_{i \in V} \log p(x^i, y^i; \theta(S)).$$

- $\theta(S)$: ML estimate of parameters given training data subset $S$.

- **Intuition about $\ell^\text{NB}$:**

  Data set $S$ $\rightarrow$ parameters $\theta(S)$ $\rightarrow$ Log-likelihood on $V$
Connect $\ell_{NB}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in X^d, y^i \in Y)\}_{i=1}^n$. 
Connect $\ell^{NB}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}^d, y^i \in \mathcal{Y})\}_{i=1}^n$.

- $p(y; \theta(S)) = \frac{m_y(S)}{|S|}$ with $m_y(S) = \sum_{i \in S} 1\{y^i = y\}$.
  - $m_y(S)$: # of samples in $S$ with label $y \in \mathcal{Y}$.
Connect $\ell_{NB}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}^d, y^i \in \mathcal{Y})\}_{i=1}^n$.

- $p(y; \theta(S)) = \frac{m_y(S)}{|S|}$ with $m_y(S) = \sum_{i \in S} 1\{y^i = y\}$.
  - $m_y(S)$: # of samples in $S$ with label $y \in \mathcal{Y}$.

- $p(x_j|y; \theta(S)) = \frac{m_{x_j,y}(S)}{m_y(S)}$ with $m_{x_j,y}(S) = \sum_{i \in S} 1\{x_j^i = x_j \land y^i = y\}$.
  - $m_{x_j,y}(S)$: # of samples in $S$ whose label is $y$ and $j$th feature is $x_j$.  

Connect $\ell_{\text{NB}}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}^d, y^i \in \mathcal{Y})\}_{i=1}^n$.

- $p(y; \theta(S)) = \frac{m_y(S)}{|S|}$ with $m_y(S) = \sum_{i \in S} 1\{y^i = y\}$.
  - $m_y(S)$: number of samples in $S$ with label $y \in \mathcal{Y}$.

- $p(x_j|y; \theta(S)) = \frac{m_{x_j,y}(S)}{m_y(S)}$ with $m_{x_j,y}(S) = \sum_{i \in S} 1\{x^i_j = x_j \land y^i = y\}$.
  - $m_{x_j,y}(S)$: number of samples in $S$ whose label is $y$ and $j$th feature is $x_j$.

- Simplify $\ell_{\text{NB}}(S)$ yields:

$$\ell_{\text{NB}}(S) = \sum_{i \in V} \sum_{j=1}^d \log p(x_j^i|y^i; \theta(S)) + \log p(y^i; \theta(S))$$

$$= \sum_{j=1}^d \sum_{x_j \in \mathcal{X}} \sum_{y \in \mathcal{Y}} m_{x_j,y}(V) \log(m_{x_j,y}(S)) + C$$

- Constant

monotone submodular function: $f_{\text{NB}}(S)$
Data Selection as Constrained Submodular Maximization

\[ \ell_{NB}^{NB}(S) = \sum_{j=1}^{d} \sum_{x_j \in X} \sum_{y \in Y} m_{x_j,y}(V) \log(m_{x_j,y}(S)) + C \]

monotone submodular function: \( f_{NB}(S) \)

- Equivalence:

\[ \max_{S \in C} \ell_{NB}^{NB}(S) \Leftrightarrow \max_{S \in C} f_{NB}(S) \]

- \( C \): constraint on the selected subset.

- Greedy **efficiently** solves it with **optimality** guarantees.
Goal: Select a small sized data set to train a good NN classifier.

Utility function: Log-likelihood function $\ell_{\text{NN}}(S)$.

$$\ell_{\text{NN}}(S) = \sum_{i \in V} \log p(x^i, y^i; \theta(S)).$$

$\theta(S)$: ML estimate of parameters given the subset $S.$

Intuition:

Data set $S$ $\rightarrow$ parameters $\theta(S)$ $\rightarrow$ Log-likelihood on $V$
Connect $\ell^{\text{NN}}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}, y^i \in \mathcal{Y})\}_{i=1}^n$.
- $V^y = \{i \in V : y^i = y\}$: samples in $V$ with label $y$. 

$\ell^{\text{NN}}(S) = \sum_{i \in V} \log p(x^i | y^i; \theta(S)) + \log p(y^i; \theta(S)) = \sum_{y \in \mathcal{Y}} \sum_{i \in V^y} \max_{j \in S \cap V^y} w(i, j)$

A monotone submodular function:

$f^{\text{NN}}(S) + C$
Connect $\ell^{\text{NN}}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}, y^i \in \mathcal{Y})\}_{i=1}^n$.
  - $V^y = \{i \in V : y^i = y\}$: samples in $V$ with label $y$.
- Pairwise similarity measure $w(i, j) = \text{const} - \|x^i - x^j\|^2$. 
Connect $\ell^{NN}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}, y^i \in \mathcal{Y})\}_{i=1}^n$.
  - $V^y = \{i \in V : y^i = y\}$: samples in $V$ with label $y$.
- Pairwise similarity measure $w(i, j) = \text{const} - \|x^i - x^j\|^2$.
- $p(y^i; \theta(S)) = \frac{m_y(S)}{|S|}$ with $m_y(S) = \sum_{j \in S} 1\{y^j = y^i\}$.
Connect $\ell^\text{NN}$ to Submodularity

- Ground set of data items: $V = \{(x^i \in \mathcal{X}, y^i \in \mathcal{Y})\}_{i=1}^n$. 
  - $V^y = \{i \in V : y^i = y\}$: samples in $V$ with label $y$.

- Pairwise similarity measure $w(i, j) = \text{const} - \|x^i - x^j\|^2$.

- $p(y^i; \theta(S)) = \frac{m_{y^i}(S)}{|S|}$ with $m_{y^i}(S) = \sum_{j \in S} 1 \{y^j = y^i\}$.

- $p(x^i | y^i; \theta(S)) \propto \exp(\max_{j \in S \cap V^y} w(i, j))$.
  - Gaussian kernel assumption.
Connect $\ell_{\text{NN}}$ to Submodularity

Ground set of data items: $V = \{(x^i \in \mathcal{X}, y^i \in \mathcal{Y})\}_{i=1}^n$.
- $V^y = \{i \in V : y^i = y\}$: samples in $V$ with label $y$.

Pairwise similarity measure $w(i, j) = \text{const} - \|x^i - x^j\|^2$.

$p(y^i; \theta(S)) = \frac{m_{y^i}(S)}{|S|}$ with $m_{y^i}(S) = \sum_{j \in S} 1\{y^j = y^i\}$.

$p(x^i|y^i; \theta(S)) \propto \exp(\max_{j \in S \cap V^y} w(i, j))$.
- Gaussian kernel assumption.

Simplify $\ell_{\text{NN}}(S)$ yields:

$$\ell_{\text{NN}}(S) = \sum_{i \in V} \log p(x^i|y^i; \theta(S)) + \log p(y^i; \theta(S))$$

$$= \sum_{y \in \mathcal{Y}} \sum_{i \in V^y} \max_{j \in S \cap V^y} w(i, j) + C$$

monotone submodular function: $f_{\text{NN}}(S)$

[Wei et al, 2015] Submodularity in Data Subset Selection and Active Learning
About \( f_{\text{NN}} = \sum_{y \in Y} \sum_{i \in V^y} \max_{j \in S \cap V^y} w(i, j) \)

- Generalizes the facility location function: \( f_{\text{fac}} = \sum_{i \in V} \max_{j \in S} w(i, j) \).
- Connect \( f_{\text{fac}} \) to utility of training NN classifiers.
- \( f_{\text{NN}} \) performs just as well on a sparse similarity graph.
About \( f_{\text{NN}} = \sum_{y \in Y} \sum_{i \in V^y} \max_{j \in S \cap V^y} w(i, j) \)

- Generalizes the *facility location function*: \( f_{\text{fac}} = \sum_{i \in V} \max_{j \in S} w(i, j) \).
- Connect \( f_{\text{fac}} \) to utility of training NN classifiers.
- \( f_{\text{NN}} \) performs just as well on a *sparse* similarity graph.

Both \( f_{\text{NN}} \) and \( f_{\text{NB}} \) only work for *supervised* setting.

- Supervised setting: requires *data label* to define function.
Active Learning

- Active learning: decide what data to label.
  - Uncertainty sampling: choose the most **uncertain** data to label.

**Figure**: Source: Settle, 2010
Filtered Active Submodular Selection (FASS)

Intuition about FASS:
- label data items that are both uncertain and diverse;

Sketch of FASS:

(a) Uncertainty filtering:
- Remove data that system is already certain about.

(b) Subset selection:
- Formulate as $\max_{|S| \leq k} f(S)$;
- $f$ is instantiated by hypothesized labels.
Experimental set-up

Evaluation Tasks:

- Text classification (20 Newsgroup).
- Handwritten digit recognition (MNIST data).

Subset Selection Baselines:

1. Random Sampling;
2. Uncertainty Sampling;

Proposed Subset Selection Methods:

1. Supervised selection with $f_{\text{NN}}$ or $f_{\text{NB}}$;
2. FASS with $f_{\text{NN}}$, $f_{\text{fac}}$, or $f_{\text{NB}}$. 
Text Classification with Naïve Bayes Classifier

![Graph showing error rate vs. number of data points for Naive Bayes Classifier (20 Newsgroup)]

- **Baselines**: **Uncertainty Sampling** (US); **Random Sampling** (RS), **Filtered Active Submodular Selection** (FASS+\{f_{fac}, f_{NN}, f_{NB}\}); **Supervised Selection** (SS+f_{NB}).

Wei et al, 2015
Text Classification with Nearest Neighbor Classifier

**Nearest Neighbor Classifier (20 Newsgroup)**

- **Baselines:** Uncertainty Sampling (US); Random Sampling (RS).
- Filtered Active Submodular Selection (FASS+$f_{NB}$);
- Supervised Selection (SS+$f_{NB}$).

![Graph showing error rate vs. number of data points for different sampling methods.](image-url)
MNIST Digit Recognition with Deep Neural Network

![Deep Neural Network classifier (MNIST)](image)

Baselines: Uncertainty Sampling (US); Random Sampling (RS), Filtered Active Submodular Selection (FASS+$\{f_{fac}, f_{NN}\}$); Supervised Selection (SS+$f_{NN}$).

Wei et al, 2015
Conclusions

- We connect submodularity to machine learning training data subset selection.

- We extend this to an active learning setting.

- We offer empirical validation, including improved deep model training.
Thank You!
Questions please.