Active Learning on Weighted Graphs Using Adaptive and Non-adaptive Approaches

Eyal En Gad, Akshay Gadde, Salman Avestimehr and Antonio Ortega

University of Southern California

ICASSP 2016

Motivation

Challenge in machine learning applications

- Unlabeled data abundant
- Labels are expensive and scarce

Motivation

Challenge in machine learning applications

- Unlabeled data abundant
- Labels are expensive and scarce

Solution: Active semi-supervised learning

- Allow the learner to select the data points to be labeled
- Predict using the labels and inherent clustering in unlabeled data

Motivation

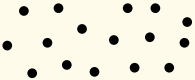
Challenge in machine learning applications

- Unlabeled data abundant
- Labels are expensive and scarce

Solution: Active semi-supervised learning

- Allow the learner to select the data points to be labeled
- Predict using the labels and inherent clustering in unlabeled data

Graph based formulation of active SSL



Unlabeled data

$\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$

Motivation

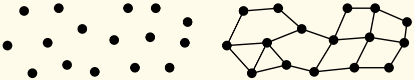
Challenge in machine learning applications

- Unlabeled data abundant
- Labels are expensive and scarce

Solution: Active semi-supervised learning

- Allow the learner to select the data points to be labeled
- Predict using the labels and inherent clustering in unlabeled data

Graph based formulation of active SSL



Unlabeled data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$

Similarity graph (Weighted k-nn)

Motivation

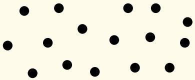
Challenge in machine learning applications

- Unlabeled data abundant
- Labels are expensive and scarce

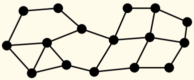
Solution: Active semi-supervised learning

- Allow the learner to select the data points to be labeled
- Predict using the labels and inherent clustering in unlabeled data

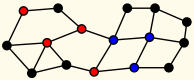
Graph based formulation of active SSL



Unlabeled data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$



Similarity graph (Weighted k-nn)



Select the nodes to be labeled

Motivation

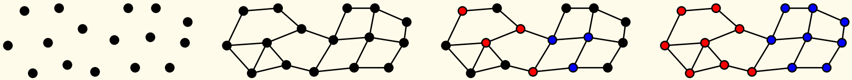
Challenge in machine learning applications

- Unlabeled data abundant
- Labels are expensive and scarce

Solution: Active semi-supervised learning

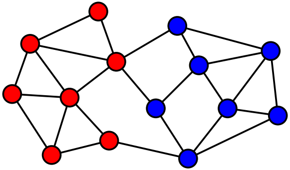
- Allow the learner to select the data points to be labeled
- Predict using the labels and inherent clustering in unlabeled data

Graph based formulation of active SSL



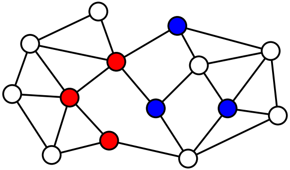
Unlabeled data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ — Similarity graph (Weighted k-nn) — Select the nodes to be labeled — Predict the rest of the labels

Problem Definition



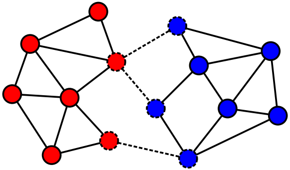
- Graph $G = (V, E)$
- Signal $f\colon V \to \{+1, -1\}$

Problem Definition



- Graph $G = (V, E)$
- Signal $f \colon V \to \{+1, -1\}$
- Observe f on $U \subset V$
- Predict f on U^c
- How to find the smallest U?

Problem Definition



- Graph $G = (V, E)$
- Signal $f\colon V \to \{+1, -1\}$
- Observe f on $U \subset V$
- Predict f on U^c
- How to find the smallest U?

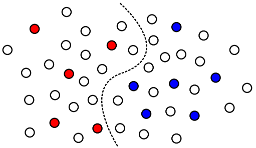
When can we expect $|U|$ to be less than $|V|$?

- Smoothness: strongly connected nodes will have similar signal
- Small cut size: very few edges with oppositely labeled endpoints compared to the total number of edges

Related Work

Global smoothness based sampling

Sample most informative nodes for good signal estimation

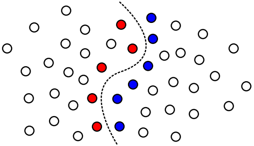


- [Guillory and Bilmes ’11]
- [Ji and Han ’12]
- [Anis, G., Ortega ’14]

non-adaptive: sample all at once

Boundary refinement sampling

Sample in order to recover the boundary nodes



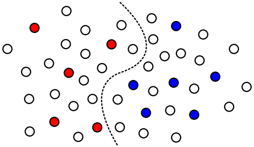
- [Zhu, Lafferty, Ghahramani ’03]
- [Osugi, Kim, Scott ’05]
- [Dasarathy, Nowak, Zhu ’15]

adaptive: sample one by one

Related Work

Global smoothness based sampling

Sample most informative nodes for good signal estimation

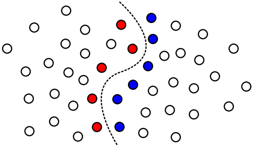


- [Guillory and Bilmes '11]
- [Ji and Han '12]
- [Anis, G., Ortega '14]

non-adaptive: sample all at once

Boundary refinement sampling

Sample in order to recover the boundary nodes



- [Zhu, Lafferty, Ghahramani '03]
- [Osugi, Kim, Scott '05]
- [Dasarathy, Nowak, Zhu '15]

adaptive: sample one by one

Which approach is better: depends on error tolerance/sampling budget

Contributions

A new sampling algorithm: Weighted S^2

- a boundary sampling approach

Contributions

A new sampling algorithm: Weighted S^2

- a boundary sampling approach
- generalization of S^2 algorithm [Dasarathy, Nowak, Zhu '15]
  - S^2 algorithm assumes an unweighted graph
  - weights capture additional info. about node similarities
  - weighted S^2 exploits the information given by the weights

Contributions

A new sampling algorithm: Weighted S^2

- a boundary sampling approach
- generalization of S^2 algorithm [Dasarathy, Nowak, Zhu '15]
  - S^2 algorithm assumes an unweighted graph
  - weights capture additional info. about node similarities
  - weighted S^2 exploits the information given by the weights
- sample complexity of weighted S^2

Contributions

A new sampling algorithm: Weighted S^2

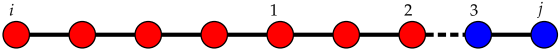
- a boundary sampling approach
- generalization of S^2 algorithm [Dasarathy, Nowak, Zhu '15]
  - S^2 algorithm assumes an unweighted graph
  - weights capture additional info. about node similarities
  - weighted S^2 exploits the information given by the weights
- sample complexity of weighted S^2

Hybrid approach: begin with global approach then switch to boundary refinement

- idea is to accelerate the convergence of label prediction using boundary refinement approach
- cutoff maximization [Anis, G., Ortega '14] $\rightarrow$ weighted S^2

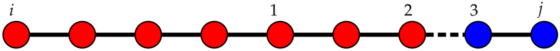
Motivation for Weighted S^2

- Weighted S^2 is a generalization of S^2 algorithm [Dasarathy, Nowak, Zhu '15]
- S^2 algorithm works on unweighted graphs
- Finds cut edges by bisecting paths connecting two oppositely labeled nodes

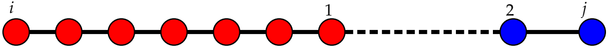


Motivation for Weighted S^2

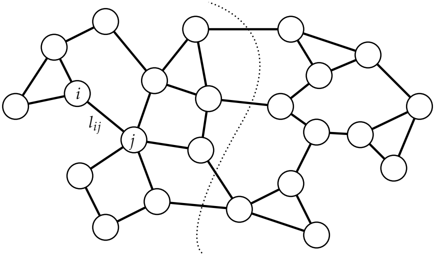
- Weighted S^2 is a generalization of S^2 algorithm [Dasarathy, Nowak, Zhu '15]
- S^2 algorithm works on unweighted graphs
- Finds cut edges by bisecting paths connecting two oppositely labeled nodes



- In ML, node $i \Leftrightarrow \mathbf{x}_i \in \mathbb{R}^d$
- Weighted S^2 takes into account $l_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ for edge (i, j)
- $d(\mathbf{x}_i, \mathbf{x}_j)$ expected to be larger for cut edges than within class edges
- Bisection based on l_{ij} can find cut edges faster

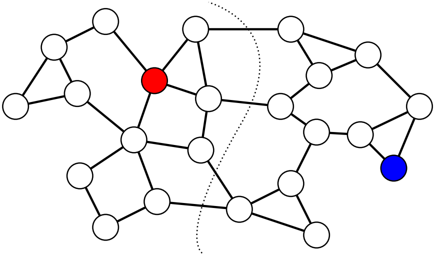


Weighted S^2 Algorithm



Given $G = (V, E)$, Lengths $l \colon E \rightarrow \mathbb{R}^+$

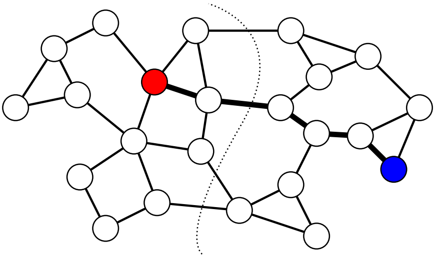
Weighted S^2 Algorithm



Given $G = (V, E)$, Lengths $l\colon E \to \mathbb{R}^+$

1. Random sample until two opposite labeled, connected nodes u, v are found

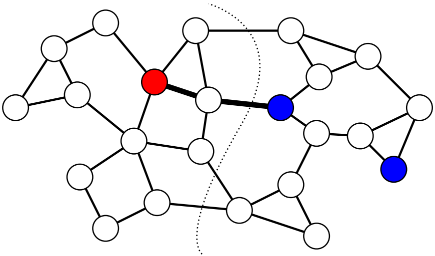
Weighted S^2 Algorithm



Given $G = (V, E)$, Lengths $l\colon E \to \mathbb{R}^+$

1. Random sample until two opposite labeled, connected nodes u, v are found
2. Find the shortest path between u and v

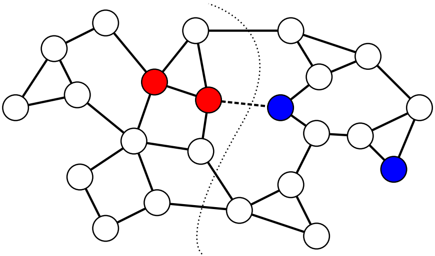
Weighted S^2 Algorithm



Given $G = (V, E)$, Lengths $l: E \to \mathbb{R}^+$

1. Random sample until two opposite labeled, connected nodes u, v are found
2. Find the shortest path between u and v
3. Bisection search: Find the cut-edge by successively sampling the nodes closest to the midpoint of the path

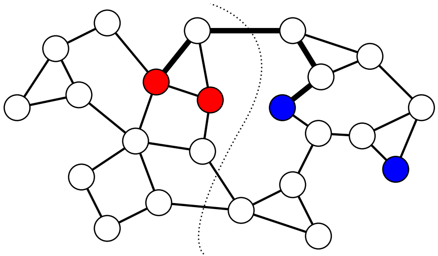
Weighted S^2 Algorithm



Given $G = (V, E)$, Lengths $l : E \to \mathbb{R}^+$

1. Random sample until two opposite labeled, connected nodes u, v are found
2. Find the shortest path between u and v
3. Bisection search: Find the cut-edge by successively sampling the nodes closest to the midpoint of the path

Weighted S^2 Algorithm

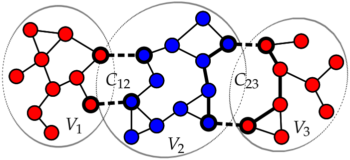


Given $G = (V, E)$, Lengths $l: E \to \mathbb{R}^+$

1. Random sample until two opposite labeled, connected nodes u, v are found
2. Find the shortest path between u and v
3. Bisection search: Find the cut-edge by successively sampling the nodes closest to the midpoint of the path
4. Remove the cut-edge and repeat until all the cut-edges are found

Sample Complexity of Weighted S^2

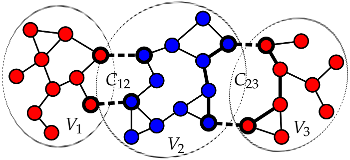
Quantities to parametrize the complexity of cut induced by f



- f partitions G into conn. comp.'s $\{V_i\}$
- Cut C into corresponding cut comp.'s $\{C_{ij}\}$
- $\beta \approx$ balanced-ness of $|V_i|$'s
- $m =$ number of cut components
- $l_n =$ max. shortest path length
- $l_{\text{cut}} =$ min cut edge length
- $l_\kappa \approx$ max dist betn two cut edges in C_{ij}

Sample Complexity of Weighted S^2

Quantities to parametrize the complexity of cut induced by f



- f partitions G into conn. comp.'s $\{V_i\}$
- Cut C into corresponding cut comp.'s $\{C_{ij}\}$
- $\beta \approx$ balanced-ness of $|V_i|$'s
- $m =$ number of cut components
- $l_n =$ max. shortest path length
- $l_{\text{cut}} =$ min cut edge length
- $l_\kappa \approx$ max dist betn two cut edges in C_{ij}

Theorem (Sample Complexity)

Weighted S^2 recovers f with prob. $> (1-\epsilon)$ if the sampling budget is at least

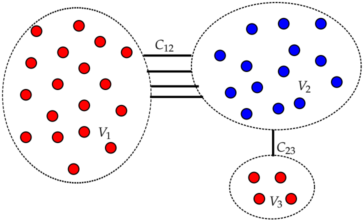
$$\underbrace{\frac{\log(1/(\beta\epsilon))}{\log(1/(1-\beta))}}_{A:\ \text{random sampling phase}} + \underbrace{m\left\lceil 2\log_2\left(\frac{l_n}{l_{\text{cut}}}\right)\right\rceil + (|\partial C| - m)\left\lceil 2\log_2\left(\frac{l_\kappa}{l_{\text{cut}}}\right)\right\rceil}_{B:\ \text{bisection search phase}}$$

Sample Complexity of Random Sampling Phase

$$A = \frac{\log(1/(\beta\epsilon))}{\log(1/(1-\beta))}$$

f partitions G into similarly labeled connected components $\{V_1, \ldots, V_p\}$

- First sample in each V_i is obtained by random sampling
- $A =$ # samples needed to sample at least one node from each V_i[1]

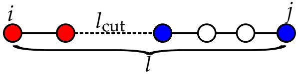


- $\beta := \min_{1 \le i \le p} |V_i|/|V|$
- measures how balanced V_i's are
- small $\beta \Rightarrow$ more samples
- less likely to sample from small component

[1] [Dasarathy, Nowak, Zhu '15]

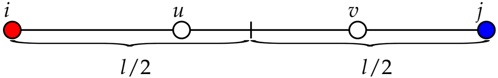
Sample Complexity of Bisection Search

Consider a sub-problem:



Lemma (Bisection search on a path)

Bisection search on path of length l discovers a cut edge of length l_{cut} in no more than $\left\lceil 2\log_2\left(\frac{l}{l_{\text{cut}}}\right)\right\rceil$ steps.



- length of the path of interest is at least halved after two queries
- bisect until discovery of cut edge $\sim$ path of interest has length l_{cut}
- number of samples $=$ number of bisections $\left\lceil 2\log_2\left(\frac{l}{l_{\text{cut}}}\right)\right\rceil$
- more samples if l is large (longer path) and l_{cut} is small (short cut edge)

Sample Complexity of Bisection Search (contd.)

$$B = \underbrace{m \left\lceil 2\log_2\left(\frac{l_n}{l_{\text{cut}}}\right)\right\rceil}_{B_1} + \underbrace{(|\partial C| - m)\left\lceil 2\log_2\left(\frac{l_\kappa}{l_{\text{cut}}}\right)\right\rceil}_{B_2}$$

Question: How many bisection searches and on what path lengths?

B_1: To discover the first cut edge (with length $\geq l_{\text{cut}}$) in each cut component bisect paths of length $\leq l_n$

B_2: To discover the remaining cut edges (with length $\geq l_{\text{cut}}$) in each cut component bisect paths of length $\leq l_\kappa$

Sample Complexity of Bisection Search (contd.)

$$B = \underbrace{m \left\lceil 2 \log_2 \left(\frac{l_n}{l_{\text{cut}}} \right) \right\rceil}_{B_1} + \underbrace{(|\partial C| - m) \left\lceil 2 \log_2 \left(\frac{l_\kappa}{l_{\text{cut}}} \right) \right\rceil}_{B_2}$$

Question: How many bisection searches and on what path lengths?

B_1 : To discover the first cut edge (with length $\geq l_{\text{cut}}$) in each cut component bisect paths of length $\leq l_n$

B_2 : To discover the remaining cut edges (with length $\geq l_{\text{cut}}$) in each cut component bisect paths of length $\leq l_\kappa$

Number of samples needed to recover f increases with

- number of boundary nodes $|\partial C|$ and number of cut components m
- graph diameter l_n and distance between cut edges l_κ
- shorter cut edges (i.e., small l_{cut})

Experiment Setup

Graph construction: data $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$ and distances $d(\mathbf{x}_i, \mathbf{x}_j)$

- G: unweighted, symmetric k-nn graph (with $k = 4$)
- G_d: same topology as G but edge-weights $w_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$
- G_s: same topology as G with $w_{ij} = \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)$ … ($\uparrow d \Leftrightarrow \mathrm{sim} \downarrow$)

Experiment Setup

Graph construction: data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ and distances $d(\mathbf{x}_i, \mathbf{x}_j)$

- G: unweighted, symmetric k-nn graph (with $k = 4$)
- G_d: same topology as G but edge-weights $w_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$
- G_s: same topology as G with $w_{ij} = \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) \ldots$ ($\uparrow d \Leftrightarrow \mathrm{sim} \downarrow$)

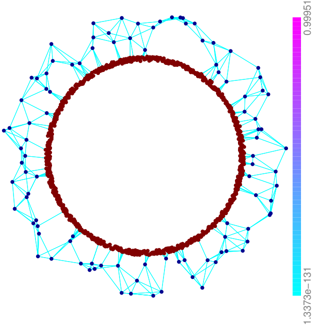
Sampling algorithms

- Weighted S^2 on G_d
- S^2 on G [Dasarathy, Nowak, Zhu '15]
- Cutoff maximization on G_s [Anis, G., Ortega '15]

Label prediction from observed samples

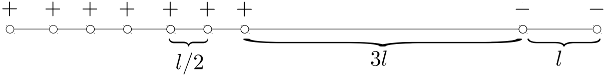
- soft labels $\hat{\mathbf{f}}$ using bandlimited interpolation [Narang et al. '13]
- threshold $\hat{\mathbf{f}}$ to get the final predictions

Synthetic Data: Advantage of Weighted S^2 over S^2



- 900 points (red) with $f = +1$ on inner circle of mean radius 1 and var 0.05
- 100 points (blue) with $f = -1$ on outer circle of mean radius 1.1 and var 0.45
- 4-nn graph using Euclidean distance in $\mathbb{R}^2$

n	$\lvert C\rvert$	$\lvert\partial C\rvert$	$\frac{\text{mean}(l_{\text{cut}})}{\text{mean}(l_{\text{non-cut}})}$	Unweighted S^2	Weighted S^2	Cutoff
1000	129	160	4.0533	237	179.2	999



An illustration of advantage of weighted S^2 (2 samples) over unweighted S^2 (3 samples)

Real World Data: Samples for Exact Recovery

USPS: handwritten digits

- $\mathbf{x}_i \in \mathbb{R}^{256}$ 16×16 image
- $\mathrm{sim}(i,j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$
- $d(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|$

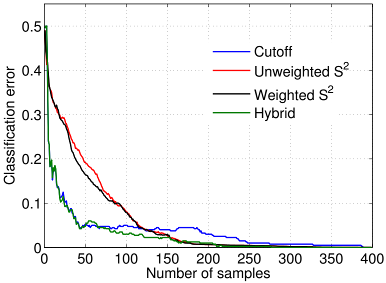
Newsgroups: documents

- $\mathbf{x}_i \in \mathbb{R}^{3000}$ tf-idf of words
- $\mathrm{sim}(i,j) = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$
- $d(i,j) = \sqrt{1 - \mathrm{sim}^2(i,j)}$

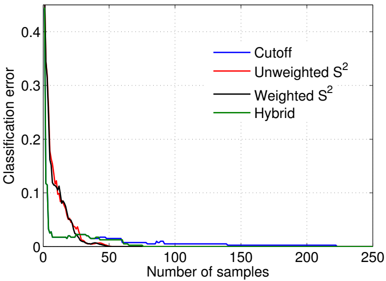
Data	n	$\lvert C\rvert$	$\lvert\partial C\rvert$	$\frac{\text{mean}(l_{\text{cut}})}{\text{mean}(l_{\text{non-cut}})}$	UW. S^2	W. S^2	Cutoff	Hybrid	n_{switch}
7 v 9	400	154	180	1.1074	312.37	312.07	399	277	47
2 v 4	400	29	39	1.1183	49.13	48.37	394	76	38
B v H	400	255	235	1.0691	368.07	368.17	399	384	42

- Weights don't help much (since $l_{\text{cut}} \approx l_{\text{non-cut}}$)
- Global approach (max cutoff) not good at recovering exact boundary
- But good at signal approximation with fewer samples

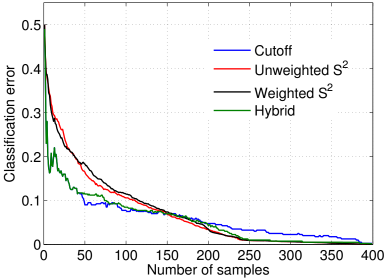
Real World Data: Error vs. Number of Samples



7 v. 9



2 v. 4



Baseball v. Hockey

- Fewer samples $\Rightarrow$ cutoff max.
- More samples $\Rightarrow$ weighted S^2
- Hybrid: start with cutoff max. then switch to weighted S^2
- Switch at sample i

$$1 - \frac{\langle \hat{\mathbf{f}}_i, \hat{\mathbf{f}}_{i-1} \rangle}{\|\hat{\mathbf{f}}_i\| \|\hat{\mathbf{f}}_{i-1}\|} < \delta \quad (\Rightarrow \hat{\mathbf{f}}_i = \hat{\mathbf{f}}_{i-1})$$

Conclusion

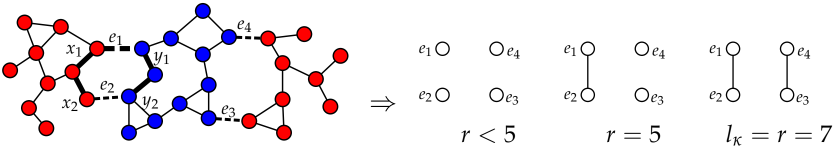
- Weighted S^2 algorithm: generalization of S^2 to weighted graphs
- Analysis of sample complexity
- Demonstration of advantage of weighted S^2 over unweighted S^2
- Active learning approach given sampling budget / error tolerance:
  - small budget / more error tolerance $\Rightarrow$ global smoothness approach (e.g., cutoff maximization)
  - large budget / less error tolerance $\Rightarrow$ boundary refinement approach (e.g., weighted S^2)
- Hybrid approach: best of both methods

References

- G. Dasarathy, R. Nowak, and X. Zhu, "S^2: An efficient graph based active learning algorithm with application to nonparametric classification," *JMLR*, 2015.
- A. Guillory and J. Bilmes, "Active semi-supervised learning using submodular functions," in *UAI*, 2011.
- B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2010.
- X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.
- S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *GlobalSIP*, 2013.
- A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *ICASSP*, 2014.
- A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *KDD*, 2014.
- M. Ji and J. Han, "A variance minimization criterion to active learning on graphs," in *AISTATS*, 2012.
- T. Osugi, D. Kim, and S. Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," in *ICDM*, 2005.

Appendix: Clustered-ness of the Cut

- $e_1, e_2 \in C$: $\delta(e_1, e_2) = d^{G-C}(x_1, x_2) + d^{G-C}(y_1, y_2) + \max\{l_{e_1}, l_{e_2}\}$



$m = 2$, Let $l_e = 1\ \forall e$

- $H_r(C, \mathcal{E})$: graph with nodes $\leftrightarrow$ cut edges in G

  for $e_1, e_2 \in C$: $\{e_1, e_2\} \in \mathcal{E}$ if and only if $\delta(e_1, e_2) \le r$

- As r increases, number of connected components in H_r decreases

  $l_\kappa =$ the smallest r for which H_r has m connected components

Larger $l_\kappa \Rightarrow$ need to bisect a longer path to get the next cut edge