Overview

1. Motivation
2. Background
3. Our Model
4. Problem Formulation
5. Results
6. Proof
7. Conclusions and Future Work
Motivation

- There is a huge number of applications for clustering
- Tons of algorithmic choices
  - Clustering algorithm/objective
  - Feature extraction
  - Preprocessing techniques
- Many conflicting outcomes
- How should we select among them?

Domain Knowledge
- How can such knowledge be incorporated into the clustering?
  - Trial and error?
  - Intuitions?
  - A more principled way?
Motivation

- There is a huge number of applications for clustering
- Tons of algorithmic choices
  - Clustering algorithm/objective
  - Feature extraction
  - Preprocessing techniques
- Many conflicting outcomes
- How should we select among them?
  - Domain Knowledge
- How can such knowledge be incorporated into the clustering?
  - Trial and error?
  - Intuitions?
  - A more principled way?
Motivation: Main Challenges

- Is there an automatic/more-principled approach for incorporating task-specific knowledge into clustering?
Motivation: Main Challenges

- Is there an automatic/more-principled approach for incorporating task-specific knowledge into clustering?
- What should be the communication protocol?
  - What should be the communication protocol?
Motivation: Main Challenges

- Is there an automatic/more-principled approach for incorporating task-specific knowledge into clustering?
  - What should be the communication protocol?
  - What should be the learning model?
Motivation: Main Challenges

- Is there an automatic/more-principled approach for incorporating task-specific knowledge into clustering?
  - What should be the communication protocol?
  - What should be the learning model?
  - What kind of algorithm should we use for learning?
Motivation: Main Challenges

- Is there an automatic/more-principled approach for incorporating task-specific knowledge into clustering?
  - What should be the communication protocol?
  - What should be the learning model?
  - What kind of algorithm should we use for learning?
  - What type of guarantees should we expect?
Overview

1 Motivation

2 Background

3 Our Model

4 Problem Formulation

5 Results

6 Proof

7 Conclusions and Future Work
Previous Work: Semi-Supervised Clustering

- Must/cannot-link constraints (Wagstaff et al. (2001))
- Constrained Clustering (E.g., Demiriz et al. (1999))
- Metric Learning (E.g., Xing et al. (2002), Alipanahi, Biggs and Ghodsi (2008))

Issues:
- Heuristic objective functions
- How many constraints do we need?

Property-based Clustering (Ackerman, Ben-David and Loker, 2010)
- Appropriate for selecting the algorithm
- Properties are not yet user-level
Overview

1. Motivation
2. Background
3. Our Model
4. Problem Formulation
5. Results
6. Proof
7. Conclusions and Future Work
Communication Protocol

CLustering with ADvice (CLAD)

1. Take a small random subset of the data
2. Have a domain expert cluster the subset
3. "Learn" a model consistent with that clustering
4. Cluster the rest of data based on the model
Communication Protocol

CLustering with ADvice (CLAD)

1. Take a small random subset of the data
2. Have a domain expert cluster the subset
3. ”Learn” a model consistent with that clustering
4. Cluster the rest of data based on the model

- A natural choice for clustering
- How can we model the task-specific knowledge?
How can we enumerate clustering algorithms?

- Fix a clustering algorithm and search for a suitable notion of representation.
- Find a mapping from the domain to a new space, similar to the metric learning approach, but with a more direct objective: the result of clustering should be consistent with what experts have in mind.
- Is this flexible enough?
- The clustering algorithm should have the richness property: for any partition, there should exist a corresponding mapping.
- E.g., $k$-means clustering enjoys $k$-richness.
- How can we avoid overfitting?
- Select the mapping from a specific class of candidate mappings.
How can we *enumerate* clustering algorithms?

Fix a clustering algorithm and search for a suitable notion of representation.
How can we enumerate clustering algorithms?

Fix a clustering algorithm and search for a suitable notion of representation.

- Find a mapping from the domain to a new space
- Similar to the metric learning approach
- With a more direct objective: the result of clustering should be consistent with what expert has in mind

Is this flexible enough?

The clustering algorithm should have the richness property: for any partition, there should exist a corresponding mapping.

E.g., $k$-means clustering enjoys $k$-richness.

How can we avoid overfitting?

Select the mapping from a specific class of candidate mappings.
How can we enumerate clustering algorithms?

Fix a clustering algorithm and search for a suitable notion of representation.

- Find a mapping from the domain to a new space
- Similar to the metric learning approach
- With a more direct objective: the result of clustering should be consistent with what expert has in mind

Is this flexible enough?

- The clustering algorithm should have the richness property: for any partition, there should exist a corresponding mapping.
- E.g., $k$-means clustering enjoys $k$-richness
How can we enumerate clustering algorithms?

Fix a clustering algorithm and search for a suitable notion of representation.

- Find a mapping from the domain to a new space
- Similar to the metric learning approach
- With a more direct objective: the result of clustering should be consistent with what expert has in mind

Is this flexible enough?

- The clustering algorithm should have the richness property: for any partition, there should exist a corresponding mapping.
  - E.g., \( k \)-means clustering enjoys \( k \)-richness

How can we avoid overfitting?

- Select the mapping from a specific class of candidate mappings
Representation Learning for CLustering with ADvice (ReCLAD)

1. Fix a core clustering method (e.g., $k$-means)
2. Take a small random subset of the data
3. Have a domain expert cluster the subset
4. Let the algorithm select a mapping (from a class of mappings)
5. Perform clustering in the mapped space

- What kind of guarantee can we expect?
  - We will establish **PAC-type** guarantees.
Overview

1. Motivation
2. Background
3. Our Model
4. Problem Formulation
5. Results
6. Proof
7. Conclusions and Future Work
Definitions

- \( f : X \mapsto \mathbb{R}^d \)
- \( C^f_X \): clustering of \( X \) induced by \( f \)
- \( C^* \): Optimal (unknown) clustering of \( X \)
- Learning algorithm \( A(S, C^*_S) \mapsto \mathcal{F} \)
- The error is the \( \Delta_X(C^*, C^f_X) \) (the difference between \( C^* \) and the clustering induced by \( f_A \))
Definitions II

- $f_A$ is $\epsilon$-optimal when $\Delta_X(C^*, C_{X}^{f_A}) \leq \epsilon$
- Agnostic $\epsilon$-optimality:

$$\Delta_X(C^*, C_{X}^{f_A}) \leq \inf_{f \in \mathcal{F}} \Delta_X(C^*, C_{X}^{f}) + \epsilon$$
Definitions II

- $f_A$ is $\epsilon$-optimal when $\Delta_X(C^*, C_X^{f_A}) \leq \epsilon$
- Agnostic $\epsilon$-optimality:

$$\Delta_X(C^*, C_X^{f_A}) \leq \inf_{f \in \mathcal{F}} \Delta_X(C^*, C_X^f) + \epsilon$$

- Distance between two $k$-clusterings:

$$\Delta_X(C^1, C^2) = \min_{\sigma \in \pi^k} \frac{1}{|X|} \sum_{i=1}^k |C^1_i \Delta C^2_{\sigma(i)}|$$
A is a PAC-ReCLAD learner for $\mathcal{F}$ with $m_\mathcal{F}$ samples if

For every $X$ and $C^*$, if $S$ is a randomly (uniformly) selected subset of $X$ of size at least $m_\mathcal{F}(\epsilon, \delta)$, then with probability at least $1 - \delta$

$$\Delta_X(C^*, C_{X}^{f_A}) \leq \inf_{f \in \mathcal{F}} \Delta_X(C^*, C_{X}^{f}) + \epsilon$$

Intuitively, for richer $\mathcal{F}$ we would need more samples. Particularly, can we bound $m_\mathcal{F}(\epsilon, \delta)$ when the core clustering method is $k$-means?
Problem Formulation - PAC-ReCLAD

PAC Representation Learning for CLustering with ADvice

A is a PAC-ReCLAD learner for $\mathcal{F}$ with $m_\mathcal{F}$ samples if

For every $X$ and $C^*$, if $S$ is a randomly (uniformly) selected subset of $X$ of size at least $m_\mathcal{F}(\epsilon, \delta)$, then with probability at least $1 - \delta$

$$\Delta_X(C^*, C^A_X) \leq \inf_{f \in \mathcal{F}} \Delta_X(C^*, C^f_X) + \epsilon$$

- Intuitively, for richer $\mathcal{F}$ we would need more samples.
- Particularly, can we bound $m_\mathcal{F}(\epsilon, \delta)$ when the core clustering method is $k$-means?
Overview

1. Motivation
2. Background
3. Our Model
4. Problem Formulation
5. Results
6. Proof
7. Conclusions and Future Work
We want to bound the sample complexity based on a notion of capacity of $\mathcal{F}$.

- VC-dimension: the size of the largest shattered set (binary functions).
- Pseudo-dimension: the size of the largest pseudo-shattered set (real-valued functions).
- We have defined a vector-valued version of it.
Uniqueness of Solution Assumption

- $k$-means’ solution may not be unique for some mappings
- Such mappings should not be selected!

($\eta, \epsilon$)-Uniqueness: Every $\eta$-optimal solution to $k$-means’ cost is $\epsilon$-close to the optimal solution

For simplifying the presentation of the results, we assume that the class $\mathcal{F}$ includes only the mappings under which the solution is unique.
Theorem

The sample complexity of representation learning for $k$-means clustering (PAC-ReKAD) with respect to $\mathcal{F}$ is upper bounded by

$$m_\mathcal{F}(\epsilon, \delta) \leq \mathcal{O}\left(\frac{k + \text{Pdim}(\mathcal{F}) + \log\left(\frac{1}{\delta}\right)}{\epsilon^2}\right)$$

where $\mathcal{O}$ hides logarithmic factors.
**Theorem**

The sample complexity of representation learning for $k$-means clustering (PAC-ReKAD) with respect to $\mathcal{F}$ is upper bounded by

$$m_{\mathcal{F}}(\epsilon, \delta) \leq O\left(\frac{k + Pdim(\mathcal{F}) + \log\left(\frac{1}{\delta}\right)}{\epsilon^2}\right)$$

where $O$ hides logarithmic factors.

**Corollary**

Let $\mathcal{F}$ be a set of linear mappings from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$. Then

$$m_{\mathcal{F}}(\epsilon, \delta) \leq O\left(\frac{k + d_1 d_2 + \log\left(\frac{1}{\delta}\right)}{\epsilon^2}\right)$$
• ReKAD: REPresentation learning for K-means clustering with ADvice
• What kind of algorithm can be a PAC-ReKAD learner?

Transductive Empirical Risk Minimization (TERM)

A TERM learner for \( \mathcal{F} \) takes as input a sample \( S \subset X \) and its clustering \( Y \) and outputs:

\[
A^{TERM}(S, Y) = \arg \min_{f \in \mathcal{F}} \Delta_S(C^f_{X|S}, Y)
\]

• It finds the mapping based on which if you cluster \( X \), the empirical error will be minimized.
Overview

1. Motivation
2. Background
3. Our Model
4. Problem Formulation
5. Results
6. Proof
7. Conclusions and Future Work
Sufficiency of Uniform Convergence

**ε-Representative Sample**

Sample $S$ is $\varepsilon$-representative with respect to $\mathcal{F}$, $X$ and the clustering $C^*$, if for every $f \in \mathcal{F}$ the following holds

$$|\Delta_X(C^*, C_f^X) - \Delta_S(C^*, C_f^X)| \leq \varepsilon$$
**Sufficiency of Uniform Convergence**

**$\epsilon$-Representative Sample**

Sample $S$ is $\epsilon$-representative with respect to $\mathcal{F}$, $X$ and the clustering $C^*$, if for every $f \in \mathcal{F}$ the following holds

\[ |\Delta_X(C^*, C_X^f) - \Delta_S(C^*, C_X^f)| \leq \epsilon \]

**Theorem (Sufficiency of Uniform Convergence)**

If $S$ is an $\frac{\epsilon}{2}$-representative sample with respect to $X$, $\mathcal{F}$ and $C^*$ then

\[ \Delta_X(C^*, C_X^{\hat{f}}) \leq \Delta_X(C^*, C_X^{f^*}) + \epsilon \]

where $f^* = \arg \min_{f \in \mathcal{F}} \Delta_X(C^*, C_X^f)$ and $\hat{f} = A^{TERM}(S, C^*_S)$. 

---

How large should be the sample so that with high probability it is $\epsilon_2$-representative (Sample complexity of uniform convergence)?

Hassan Ashtiani & Shai Ben-David

Clustering with Advice

December 2015
Sufficiency of Uniform Convergence

**ε-Representative Sample**

Sample $S$ is $\varepsilon$-representative with respect to $\mathcal{F}$, $X$ and the clustering $C^*$, if for every $f \in \mathcal{F}$ the following holds

$$|\Delta_X(C^*, C_X^f) - \Delta_S(C^*, C_X^f)| \leq \varepsilon$$

**Theorem (Sufficiency of Uniform Convergence)**

If $S$ is an $\varepsilon/2$-representative sample with respect to $X$, $\mathcal{F}$ and $C^*$ then

$$\Delta_X(C^*, C_{\hat{f}}^X) \leq \Delta_X(C^*, C_{\hat{f}^*}^X) + \varepsilon$$

where $f^* = \arg\min_{f \in \mathcal{F}} \Delta_X(C^*, C_X^f)$ and $\hat{f} = A^{TERM}(S, C^*|_S)$.

How large should be the sample so that with high probability it is $\varepsilon/2$-representative (Sample complexity of uniform convergence)?

Hassan Ashtiani & Shai Ben-David ()

Clustering with Advice

December 2015
Proof of Uniform Convergence: Covering Numbers

- If $|\mathcal{F}| = 1$ then Hoeffding’s inequality says we have uniform convergence.
- For finite classes: we can use union bound.
- For infinite classes?
Proof of Uniform Convergence: Covering Numbers

- If $|\mathcal{F}| = 1$ then Hoeffding’s inequality says we have uniform convergence.
- For finite classes: we can use union bound.
- For infinite classes?
- $\mathcal{N}(\mathcal{F}, d, \epsilon)$ or covering number: Roughly, the number of $\epsilon$-different members of $\mathcal{F}$ with respect to $d(., .)$.
- $\Delta$-distance between two mappings:
  \[
  \Delta_X(f_1, f_2) = \Delta_X(C_{X}^{f_1}, C_{X}^{f_2})
  \]
- $L_1$ distance between two mappings:
  \[
  d_{L_1}^X(f_1, f_2) = \frac{1}{|X|} \sum_{x \in X} \| f_1(x) - f_2(x) \|_2
  \]
Plan of Attack

1. Bound $m^F(\epsilon, \delta)$ based on $m^F_{UC}(\epsilon, \delta)$
Plan of Attack

1. Bound $m^F(\epsilon, \delta)$ based on $m^F_{UC}(\epsilon, \delta)$
2. Bound the $m^F_{UC}(\epsilon, \delta)$ based on $N(F, \Delta_X, \epsilon)$ and $\delta$
Plan of Attack

1. Bound $m^\mathcal{F}(\epsilon, \delta)$ based on $m^\mathcal{F}_{UC}(\epsilon, \delta)$
2. Bound the $m^\mathcal{F}_{UC}(\epsilon, \delta)$ based on $\mathcal{N}(\mathcal{F}, \Delta_X, \epsilon)$ and $\delta$
3. Bound $\mathcal{N}(\mathcal{F}, \Delta_X, \epsilon)$ based on $\mathcal{N}(\mathcal{F}, d^X_{L_1}, \epsilon)$
Plan of Attack

1. Bound $m^\mathcal{F} (\epsilon, \delta)$ based on $m^\mathcal{F}_{UC} (\epsilon, \delta)$
2. Bound the $m^\mathcal{F}_{UC} (\epsilon, \delta)$ based on $\mathcal{N} (\mathcal{F}, \Delta_X, \epsilon)$ and $\delta$
3. Bound $\mathcal{N} (\mathcal{F}, \Delta_X, \epsilon)$ based on $\mathcal{N} (\mathcal{F}, d^X_{L_1}, \epsilon)$
4. Bound $\mathcal{N} (\mathcal{F}, d^X_{L_1}, \epsilon)$ based on $Pdim (\mathcal{F})$ and $\epsilon$
Plan of Attack

1. Bound $m^\mathcal{F}(\epsilon, \delta)$ based on $m^\mathcal{F}_{UC}(\epsilon, \delta)$
2. Bound the $m^\mathcal{F}_{UC}(\epsilon, \delta)$ based on $\mathcal{N}(\mathcal{F}, \Delta_X, \epsilon)$ and $\delta$
3. Bound $\mathcal{N}(\mathcal{F}, \Delta_X, \epsilon)$ based on $\mathcal{N}(\mathcal{F}, d^X_{L_1}, \epsilon)$
4. Bound $\mathcal{N}(\mathcal{F}, d^X_{L_1}, \epsilon)$ based on $Pdim(\mathcal{F})$ and $\epsilon$
5. Bound $Pdim(\mathcal{F})$
Overview

1 Motivation

2 Background

3 Our Model

4 Problem Formulation

5 Results

6 Proof

7 Conclusions and Future Work
Conclusions

- We proposed a formal framework for exploiting domain knowledge in clustering.
  - Supervision Protocol: CLAD
  - Model: ReCLAD
  - Formal specification: PAC-ReCLAD
- For ReKAD, the sample complexity was bounded based on the pseudo-dimension of the class of mappings.
Conclusions

- We proposed a formal framework for exploiting domain knowledge in clustering.
  - Supervision Protocol: CLAD
  - Model: ReCLAD
  - Formal specification: PAC-ReCLAD
- For ReKAD, the sample complexity was bounded based on the pseudo-dimension of the class of mappings.
- Future Work
  - Provide computationally efficient algorithms
  - Fixed number of clusters
Thank You!
Proof of Uniform Convergence

Lemma

Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then for $f_1, f_2 \in \mathcal{F}$ if

$$d_{L_1}^X(f_1, f_2) < \frac{\eta}{12}$$

then

$$\Delta_X(f_1, f_2) < 2\epsilon$$
Proof of Uniform Convergence

**Theorem**

Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$m_{\mathcal{F}}^{UC}(\epsilon, \delta) \leq O\left(\frac{\log k! + \log \mathcal{N}(\mathcal{F}, d_{L_1}^X, \frac{\eta}{\alpha})}{\epsilon^2} + \log\left(\frac{1}{\delta}\right)\right)$$
Proof of Uniform Convergence

**Theorem**

Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$m_{\mathcal{F}}^{UC}(\epsilon, \delta) \leq O\left(\frac{\log k! + \log \mathcal{N}(\mathcal{F}, d_{L_1}^X, \frac{\eta}{\alpha}) + \log(\frac{1}{\delta})}{\epsilon^2}\right)$$

**Theorem**

Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$m_{\mathcal{F}}^{UC}(\epsilon, \delta) \leq O\left(\frac{k + \text{Pdim}(\mathcal{F}) + \log(\frac{1}{\delta})}{\epsilon^2}\right)$$

where $O()$ hides logarithmic factors of $k$ and $\frac{1}{\eta}$. 
$k$-means’ solution may not be unique for some mappings
 Such mappings should not be selected!
Uniqueness of Solution Assumption

- $k$-means’ solution may not be unique for some mappings.
- Such mappings should not be selected!
- We should compare the output of the algorithm only to those mappings in $\mathcal{F}$ that have unique solutions.
Uniqueness of Solution Assumption

- $k$-means’ solution may not be unique for some mappings
- Such mappings should not be selected!
- We should compare the output of the algorithm only to those mappings in $\mathcal{F}$ that have unique solutions
- $(\eta, \epsilon)$-Uniqueness: Every $\eta$-optimal solution to $k$-means’ cost is $\epsilon$-close to the optimal solution
Uniqueness of Solution Assumption

- *k*-means’ solution may not be unique for some mappings
- Such mappings should not be selected!
- We should compare the output of the algorithm only to those mappings in $\mathcal{F}$ that have unique solutions
- $(\eta, \epsilon)$-Uniqueness: Every $\eta$-optimal solution to *k*-means’ cost is $\epsilon$-close to the optimal solution
- For simplifying the presentation of the results, we assume that the class $\mathcal{F}$ includes only the mappings under which the solution is unique.
ε-Representative Sample

Sample $S$ is $\epsilon$-representative with respect to $\mathcal{F}$, $X$ and the clustering $C^*$, if for every $f \in \mathcal{F}$ the following holds

$$|\Delta_X(C^*, C^*_f) - \Delta_S(C^*, C^*_f)| \leq \epsilon$$
\(\epsilon\)-Representative Sample

Sample \(S\) is \(\epsilon\)-representative with respect to \(\mathcal{F}\), \(X\) and the clustering \(C^*\), if for every \(f \in \mathcal{F}\) the following holds

\[|\Delta_X(C^*, C^f_X) - \Delta_S(C^*, C^f_X)| \leq \epsilon\]

Theorem (Sufficiency of Uniform Convergence)

If \(S\) is an \(\frac{\epsilon}{2}\)-representative sample with respect to \(X\), \(\mathcal{F}\) and \(C^*\) then

\[\Delta_X(C^*, C^\hat{f}_X) \leq \Delta_X(C^*, C^{f^*}_X) + \epsilon\]

where \(f^* = \arg\min_{f \in \mathcal{F}} \Delta_X(C^*, C^f_X)\) and \(\hat{f} = A^{TERM}(S, C^*|_S)\).
ε-Representative Sample

Sample \( S \) is \( \epsilon \)-representative with respect to \( \mathcal{F}, X \) and the clustering \( C^* \), if for every \( f \in \mathcal{F} \) the following holds

\[
|\Delta_X(C^*, C_X^f) - \Delta_S(C^*, C_X^f)| \leq \epsilon
\]

Theorem (Sufficiency of Uniform Convergence)

If \( S \) is an \( \frac{\epsilon}{2} \)-representative sample with respect to \( X, \mathcal{F} \) and \( C^* \) then

\[
\Delta_X(C^*, C_X^{\hat{f}}) \leq \Delta_X(C^*, C_X^{f^*}) + \epsilon
\]

where \( f^* = \arg\min_{f \in \mathcal{F}} \Delta_X(C^*, C_X^f) \) and \( \hat{f} = A^{TERM}(S, C^* \mid S) \).

How large should be the sample so that with high probability it is \( \frac{\epsilon}{2} \)-representative (Sample complexity of uniform convergence)?
\[ \Delta_X(C^*, C^f_X) \leq \Delta_S(C^*, C^f_X) + \frac{\epsilon}{2} \]

\[ \leq \Delta_S(C^*, C^{f*}_X) + \frac{\epsilon}{2} \]

\[ \leq \Delta_X(C^*, C^{f*}_X) + \frac{\epsilon}{2} + \frac{\epsilon}{2} \]

\[ \leq \Delta_X(C^*, C^{f*}_X) + \epsilon \]
Reduction to Binary Classes

\[ h_{f_1, f_2}^{f_1, f_2}(x) = \begin{cases} 
1 & x \in \bigcup_{i=1}^k (C_{i}^{f_1} \Delta C_{\sigma(i)}^{f_2}) \\
0 & \text{otherwise} 
\end{cases} \]
Reduction to Binary Classes

\[ h_{\sigma}^{f_1, f_2}(x) = \begin{cases} 
1 & x \in \bigcup_{i=1}^{k} (C_{f_1}^i \Delta C_{\sigma(i)}^{f_2}) \\
0 & \text{otherwise} 
\end{cases} \]

\[ H^F = \{ h_{\sigma}^{f_1, f_2}(.) : f_1, f_2 \in F, \sigma \in \pi \} \]
Reduction to Binary Classes

\[
h_{\sigma}^{f_1, f_2}(x) = \begin{cases} 
1 & x \in \bigcup_{i=1}^{k} (C_{\sigma(i)}^{f_1} \Delta C_{\sigma(i)}^{f_2}) \\
0 & \text{otherwise}
\end{cases}
\]

\[H^F = \{ h_{\sigma}^{f_1, f_2}(.) : f_1, f_2 \in \mathcal{F}, \sigma \in \pi \}\]

**Theorem (Reduction to Binary Classification)**

If \( S \subset X \) is such that for all \( h \in H^F \)

\[|h(S) - h(X)| \leq \epsilon\]

then \( S \) will be \( \epsilon \)-representative, i.e., for all \( f_1, f_2 \in \mathcal{F} \) we have

\[|\Delta_X(C_X^{f_1}, C_X^{f_2}) - \Delta_S(C_X^{f_1}, C_X^{f_2})| \leq \epsilon\]
Reduction to Binary Classes: Proof

$$|\Delta_S(C_X^{f_1}, C_X^{f_2}) - \Delta_X(C_X^{f_1}, C_X^{f_2})|$$
Reduction to Binary Classes: Proof

$$|\Delta_S(C_X^{f_1}, C_X^{f_2}) - \Delta_X(C_X^{f_1}, C_X^{f_2})|$$

$$= \left| \left( \min_{\sigma} \frac{1}{|S|} \sum_{x \in S} h_{\sigma}^{f_1,f_2} \right) - \left( \min_{\sigma} \frac{1}{|X|} \sum_{x \in X} h_{\sigma}^{f_1,f_2} \right) \right|$$
Reduction to Binary Classes: Proof

\[ |\Delta_s(C^f_1, C^f_2) - \Delta_X(C^f_1, C^f_2)| \]

\[ = \left| \left( \min_\sigma \frac{1}{|S|} \sum_{x \in S} h^{f_1, f_2}_\sigma \right) - \left( \min_\sigma \frac{1}{|X|} \sum_{x \in X} h^{f_1, f_2}_\sigma \right) \right| \]

\[ \leq \left| \max_\sigma \left( \frac{1}{|S|} \sum_{x \in S} h^{f_1, f_2}_\sigma - \frac{1}{|X|} \sum_{x \in X} h^{f_1, f_2}_\sigma \right) \right| \]
Reduction to Binary Classes: Proof

\[ |\Delta_S(C_{X}^{f_1}, C_{X}^{f_2}) - \Delta_X(C_{X}^{f_1}, C_{X}^{f_2})| = \left| \left( \min_{\sigma} \frac{1}{|S|} \sum_{x \in S} h_{\sigma}^{f_1,f_2} \right) - \left( \min_{\sigma} \frac{1}{|X|} \sum_{x \in X} h_{\sigma}^{f_1,f_2} \right) \right| \leq \left| \max_{\sigma} \left( \frac{1}{|S|} \sum_{x \in S} h_{\sigma}^{f_1,f_2} - \frac{1}{|X|} \sum_{x \in X} h_{\sigma}^{f_1,f_2} \right) \right| \leq \max_{\sigma} \left( h_{\sigma}^{f_1,f_2}(S) - h_{\sigma}^{f_1,f_2}(X) \right) \leq \epsilon \]
Reduction to Binary Classes

- Enough to have an $\epsilon$-representative sample w.r.t. $H^F$
Reduction to Binary Classes

- Enough to have an $\epsilon$-representative sample w.r.t. $H^F$
- How large should be the training sample so that with high probability it is $\epsilon$-representative w.r.t. $H^F$?
Reduction to Binary Classes

- Enough to have an $\epsilon$-representative sample w.r.t. $H^F$
- How large should be the training sample so that with high probability it is $\epsilon$-representative w.r.t. $H^F$?
- Or, sample complexity of uniform convergence w.r.t. $H^F$?
Reduction to Binary Classes

- Enough to have an $\epsilon$-representative sample w.r.t. $H^F$.
- How large should be the training sample so that with high probability it is $\epsilon$-representative w.r.t. $H^F$?
- Or, sample complexity of uniform convergence w.r.t. $H^F$?
- Can we bound the sample complexity based on VC-DIM($H^F$)?
Reduction to Binary Classes

- Enough to have an \( \epsilon \)-representative sample w.r.t. \( H^F \)
- How large should be the training sample so that with high probability it is \( \epsilon \)-representative w.r.t. \( H^F \)?
- Or, sample complexity of uniform convergence w.r.t. \( H^F \)?
- Can we bound the sample complexity based on VC-DIM(\( H^F \))?  
- Yes, but \( H^F \) depends on the distribution (i.e., on \( X \))

\[
h^f_\sigma(x) = \begin{cases} 
1 & x \in \bigcup_{i=1}^k (C_{i}^{f_1} \Delta C_{\sigma(i)}^{f_2}) \\
0 & \text{otherwise}
\end{cases}
\]
Reduction to Binary Classes

- Enough to have an $\epsilon$-representative sample w.r.t. $H^F$
- How large should be the training sample so that with high probability it is $\epsilon$-representative w.r.t. $H^F$?
- Or, sample complexity of uniform convergence w.r.t. $H^F$?
- Can we bound the sample complexity based on VC-DIM($H^F$)?
- Yes, but $H^F$ depends on the distribution (i.e., on $X$)

$$h_{\sigma,f_1,f_2}^F(x) = \begin{cases} 
1 & x \in \bigcup_{i=1}^k (C_{\sigma(i)}^{f_1} \Delta C_{\sigma(i)}^{f_2}) \\
0 & \text{otherwise}
\end{cases}$$

- Can we give a (distribution-free) bound based on the capacity of $F$?
Proof of Uniform Convergence

**Lemma**

Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then for $f_1, f_2 \in \mathcal{F}$ if

$$d_{L_1}^X(f_1, f_2) < \frac{\eta}{12}$$

then

$$\Delta_X(f_1, f_2) < 2\epsilon$$
Proof of Uniform Convergence

Lemma
Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then for $f_1, f_2 \in \mathcal{F}$ if

$$d_{L_1}(f_1, f_2) < \frac{\eta}{12}$$

then

$$\Delta_X(f_1, f_2) < 2\epsilon$$

Lemma
Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$\mathcal{N}(H^\mathcal{F}, d_{L_1}^X, 2\epsilon) \leq k! \mathcal{N}(\mathcal{F}, d_{L_1}^X, \frac{\eta}{12})$$

Hassan Ashtiani & Shai Ben-David ( )
Clustering with Advice
December 2015
Proof of Uniform Convergence

**Theorem**

Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$m_{\mathcal{F}}^{UC} (\epsilon, \delta) \leq O\left( \log k! + \log \mathcal{N}(\mathcal{F}, d^X_{L_1}, \frac{\eta}{\alpha}) + \log \left( \frac{1}{\delta} \right) \right)$$
Proof of Uniform Convergence

Theorem
Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$m_{\mathcal{F}}^{UC}(\epsilon, \delta) \leq O\left(\frac{\log k! + \log \mathcal{N}(\mathcal{F}, d_{L_1}^X, \frac{\eta}{\alpha}) + \log(\frac{1}{\delta})}{\epsilon^2}\right)$$

Theorem
Let $\mathcal{F}$ be a class of mappings with $(\eta, \epsilon)$-uniqueness property. Then

$$m_{\mathcal{F}}^{UC}(\epsilon, \delta) \leq O\left(\frac{k + Pdim(\mathcal{F}) + \log(\frac{1}{\delta})}{\epsilon^2}\right)$$

where $O()$ hides logarithmic factors of $k$ and $\frac{1}{\eta}$. 
Covering Number

- \( d(.,.) \): a metric over \( \mathcal{F} \)
Covering Number

- $d(., .)$: a metric over $\mathcal{F}$
- $\Delta$-distance between two mappings:

$$\Delta_X(f_1, f_2) = \Delta_X(C_{f_1}^X, C_{f_2}^X)$$
Covering Number

- $d(., .)$: a metric over $\mathcal{F}$
- $\Delta$-distance between two mappings:
  $$\Delta_X(f_1, f_2) = \Delta_X(C_{f_1}^X, C_{f_2}^X)$$
- $L_1$ distance between two mappings:
  $$d_{L_1}^X(f_1, f_2) = \frac{1}{|X|} \sum_{x \in X} \|f_1(x) - f_2(x)\|_2$$
Covering Number

- \(d(.,.)\): a metric over \(\mathcal{F}\)
- \(\Delta\)-distance between two mappings:
  \[
  \Delta_X(f_1, f_2) = \Delta_X(C_{f_1}^X, C_{f_2}^X)
  \]

- \(L_1\) distance between two mappings:
  \[
  d_{L_1}^X(f_1, f_2) = \frac{1}{|X|} \sum_{x \in X} \|f_1(x) - f_2(x)\|_2
  \]

- \(\mathcal{N}(\mathcal{F}, d, \epsilon)\) or covering number: Roughly, the number of \(\epsilon\)-different members of \(\mathcal{F}\) with respect to \(d(.,.)\)
Let $\mathcal{F}$ be a class of $(\eta, \epsilon)$-unique mappings. Then the sample complexity of learning representation for $k$-means clustering with respect to $\mathcal{F}$ is upper bounded by

$$m_{\mathcal{F}}(\epsilon, \delta) \leq O\left(\frac{k + Pdim(\mathcal{F}) + \log\left(\frac{1}{\delta}\right)}{\epsilon^2}\right)$$

where $O$ hides logarithmic factors of $k$ and $\frac{1}{\eta}$. 