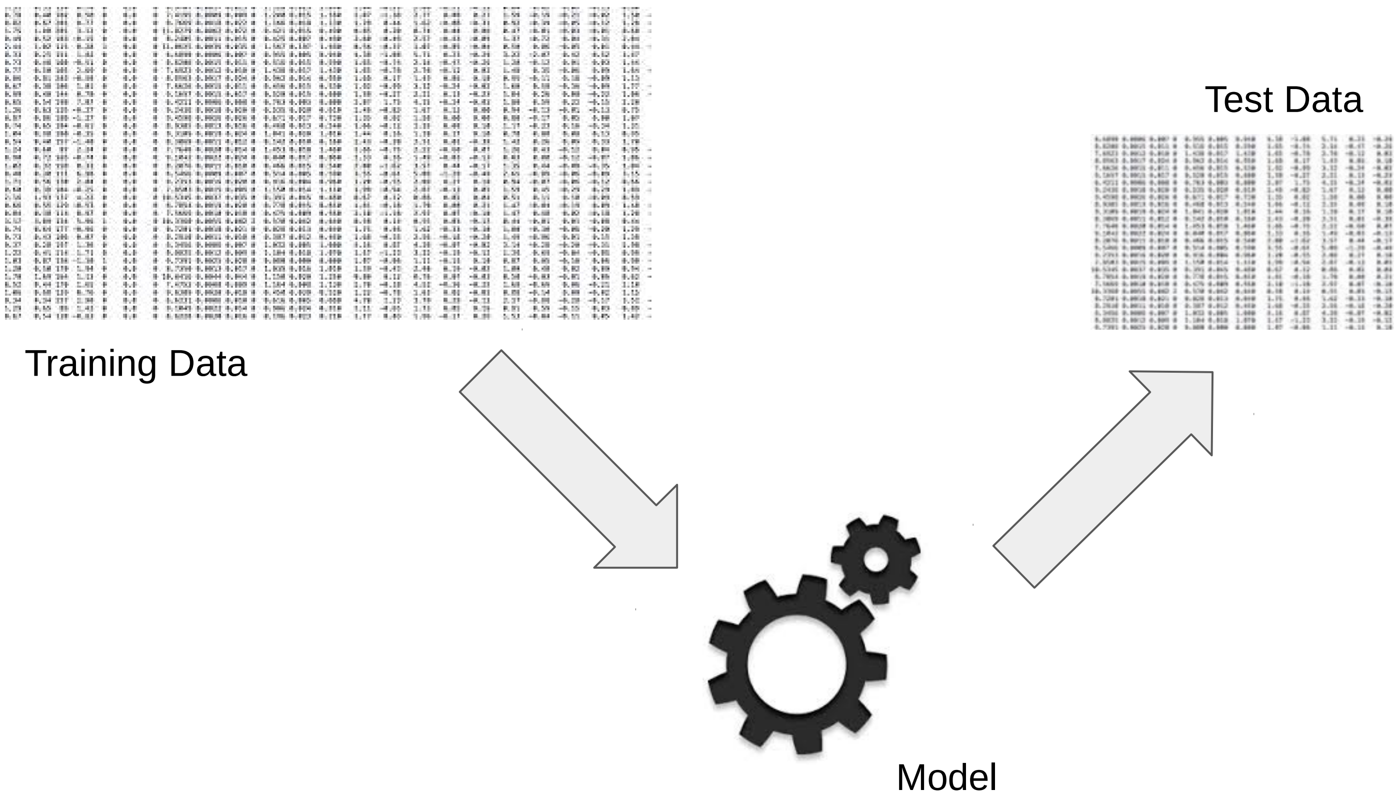
Efficient Sampling-Based Kernel Mean Matching

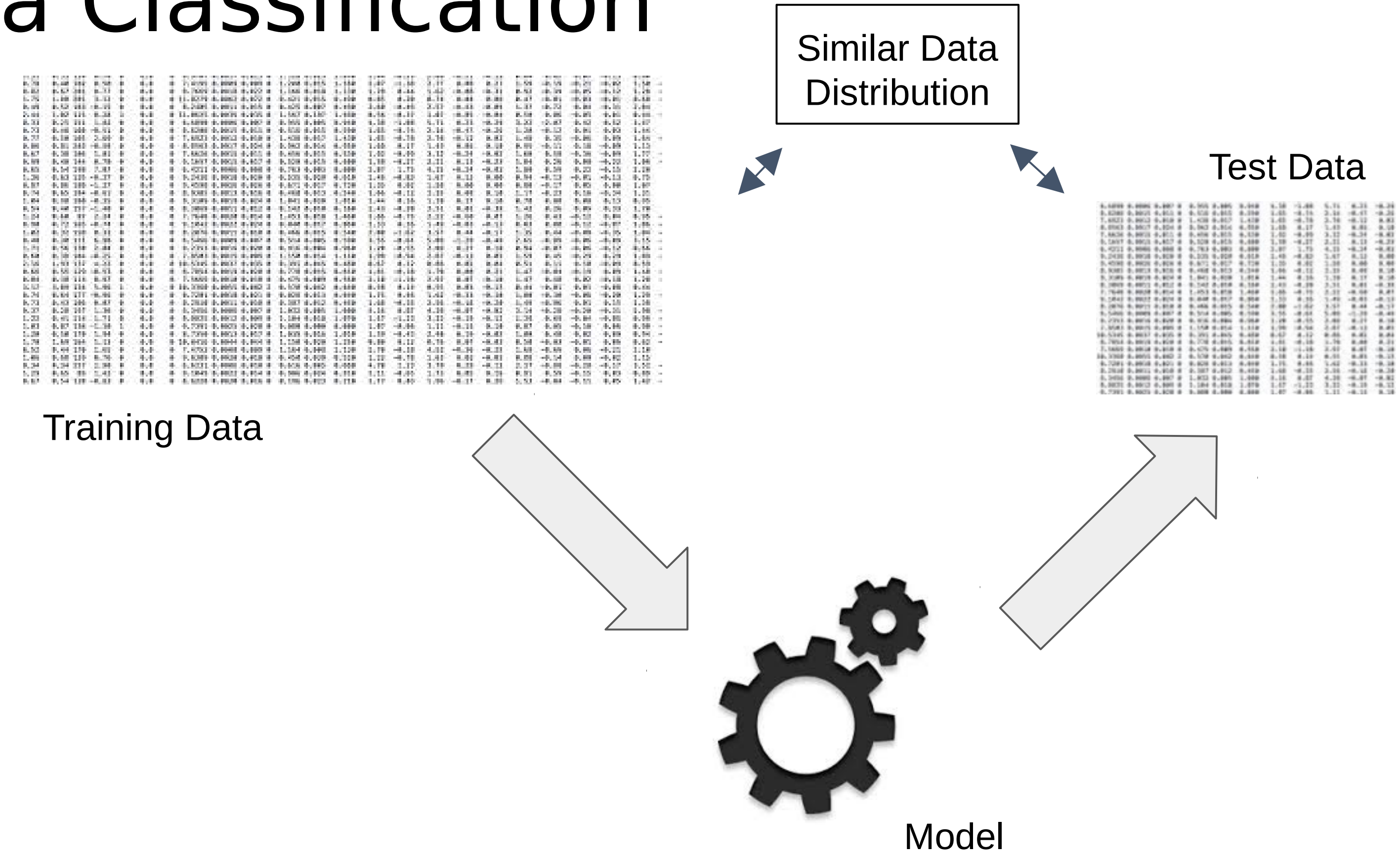
Swarup Chandra, Ahsanul Haque, Latifur Khan and Charu Aggarwal*
University of Texas at Dallas
*IBM Research

This material is based upon work supported by

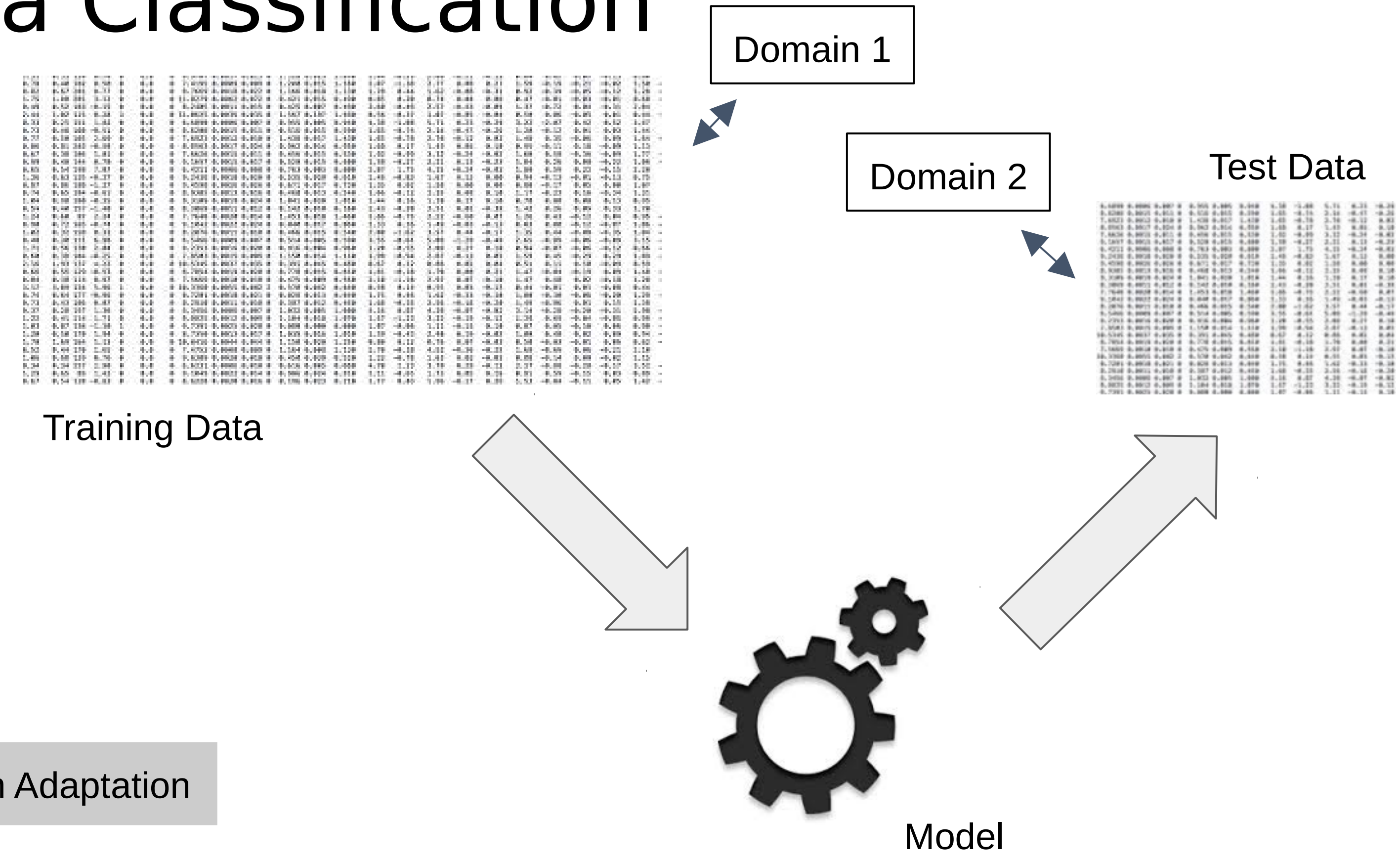
Data Classification



Data Classification

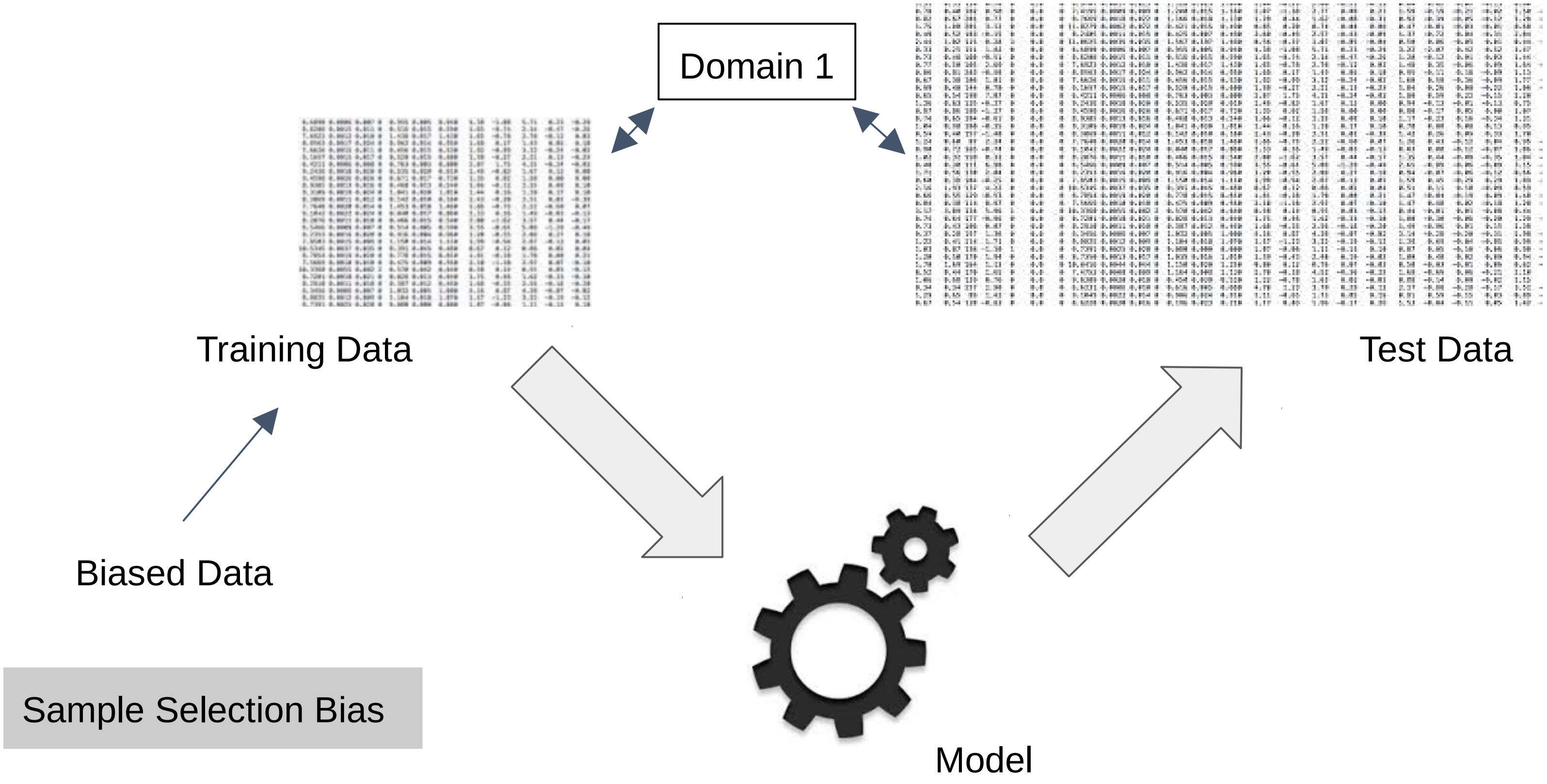


Data Classification



Domain Adaptation

Data Classification



Data Classification

- Biased Training Data
  - Limited labeled data
  - High cost
- Covariate shift assumption
  - Between training (tr) and $p_{tr}(y|\mathbf{x}) = p_{te}(y|\mathbf{x})$ ution
  - Equal conditional distribu $p_{tr}(\mathbf{x}) \neq p_{te}(\mathbf{x})$
  - Unequal marginal distribution:
- Use training c $\hat{p}_{tr} = p_{te}$ bels to classify test d
  - Solution: make
  - Comput $\beta(\mathbf{x}) = \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})}$ nstance weight

Problem:
- Unknown training data distribution
  Unknown test data distribution

Solution:
- Compute weight directly

Kernel Mean Matching

- Minimize mean distance between weighted training data distribution and test data distribution
- Maximum Mean Discrepency

$$\left\| E_{\mathbf{x} \sim p_{tr}(\mathbf{x})}[\beta(\mathbf{x})\phi(\mathbf{x})] - E_{\mathbf{x} \sim p_{te}(\mathbf{x})}[\phi(\mathbf{x})] \right\|$$

- Empirical

$$\hat{\beta} \approx \underset{\beta}{\text{minimize}} \ \frac{1}{2}\boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} - \boldsymbol{\kappa}^T \boldsymbol{\beta}$$

$$\text{subject to} \quad \beta(\mathbf{x}^{(i)}) \in [0, B], \forall i \in \{1 \ldots n_{tr}\}$$

$$\left| \sum_{i=1}^{n_{tr}} \beta(\mathbf{x}^{(i)}) - n_{tr} \right| \leq n_{tr}\epsilon$$

Requires complete training and test data to be in the memory

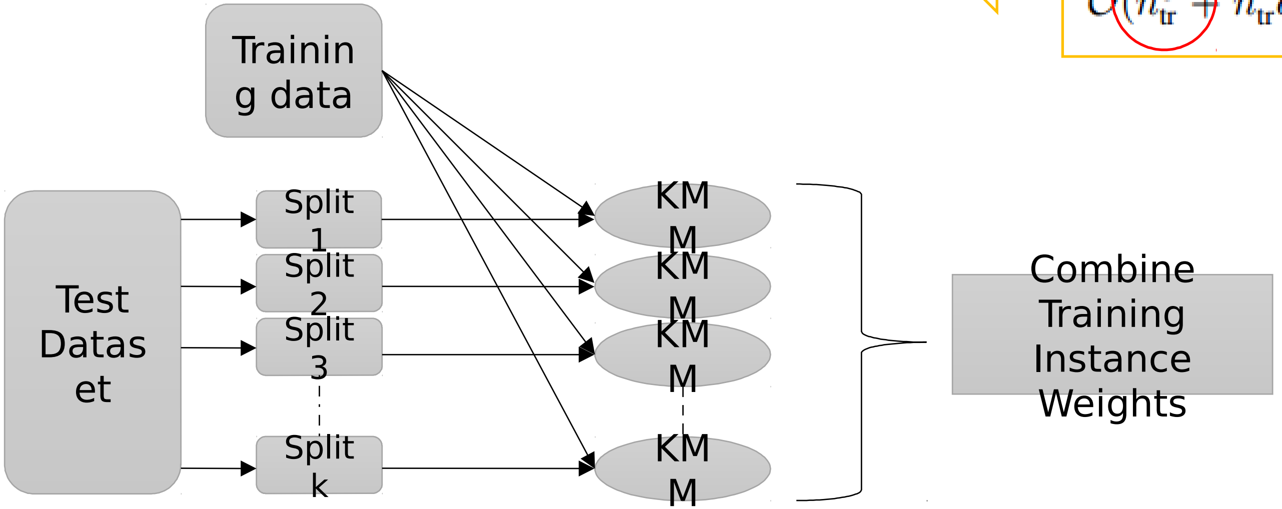
$$K^{(ij)} = h(\mathbf{x}_{tr}^{(i)}, \mathbf{x}_{tr}^{(j)}) \qquad \kappa^{(i)} = \frac{n_{tr}}{n_{te}} \sum_{j=1}^{n_{te}} h(\mathbf{x}_{tr}^{(i)}, \mathbf{x}_{te}^{(j)})$$

Kernel Mean Matching

- Time Complexity $\mathcal{O}(n_{tr}^3 + n_{tr}^2 d + n_{tr} n_{te} d)$
- Related Work: Ensemble Kernel Mean M

Time Complexity:

$$\mathcal{O}(n_{\mathrm{tr}}^3 + n_{\mathrm{tr}}^2 d + n_{\mathrm{tr}} m d + k n_{tr})$$

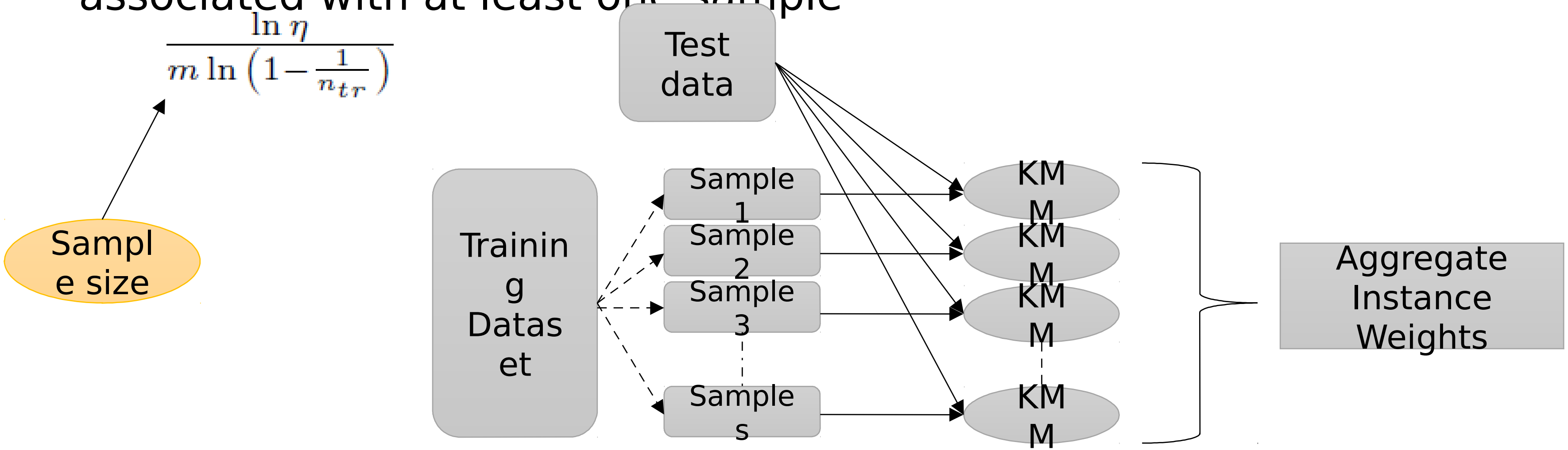


Miao, Yun-Qian, Ahmed K. Farahat, and Mohamed S. Kamel. "Ensemble Kernel Mean Matching.", *ICDM 2015*.

Sampling-Based Approach

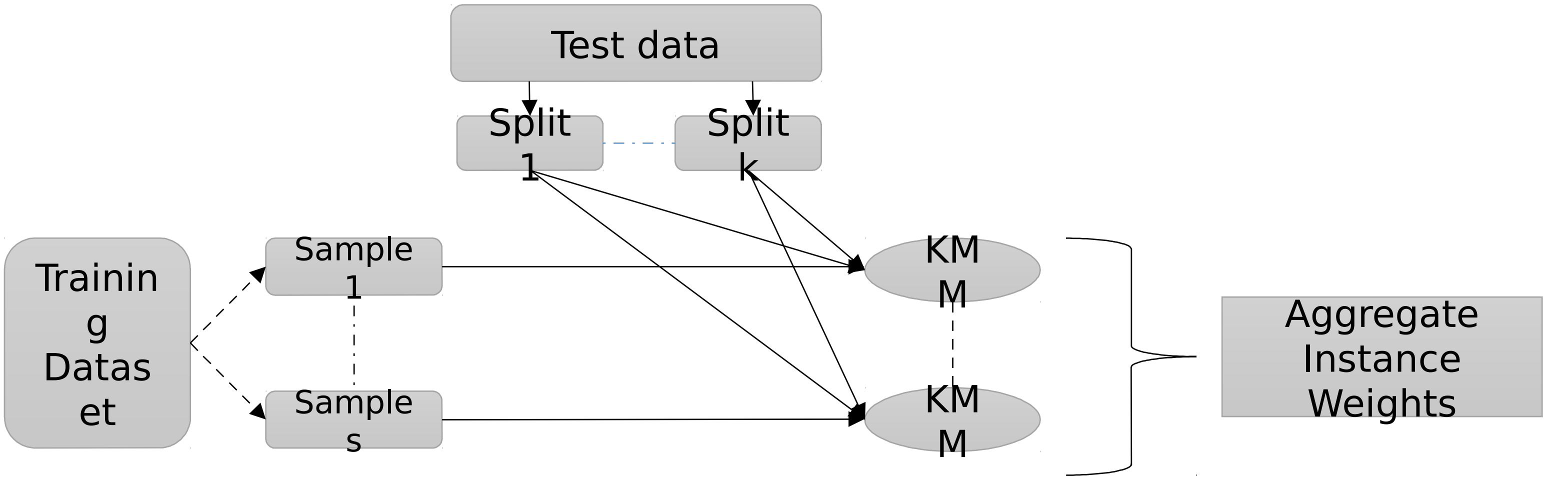
- Very Fast Kernel Mean Matching (VFKMM)
  - m/n bootstrap sampling
  - Sample training data with replacement
  - Minimum number of samples such that each training instance is associated with at least one sample

$$\frac{\ln \eta}{m \ln\left(1 - \frac{1}{n_{tr}}\right)}$$



Sampling-Based Approach

- Extended Very Fast Kernel Mean Matching (EVFKMM)
  - Sample training data with replacement
  - Split test data into k parts (sampling without replacement)



Empirical Evaluation

Dataset

Dataset	# Features	Total Size
ForestCover	54	50,000
KDD	34	50,000
Syn002	70	50,000
MNIST	780	50,000

- Available on UCI data repository
- Training data bias induction:

$$p(\xi = 1|\mathbf{x}^{(i)}) = \exp \frac{-\|\mathbf{x}^{(i)} - \bar{\mathbf{x}}\|}{\sigma}$$

Competing Methods

Methods	Description
cenKMM	Original Method
ensKMM	Related Work* (split test data)
ensTrKMM	Baseline Method (split training data)
VFKMM	Proposed Method (sample training data)
EVFKMM	Extended VFKMM (also split test data)

cenKMM				✓
ensKMM		✓		
ensTrKMM	✓			
VFKMM	✓			✓
EVFKMM				

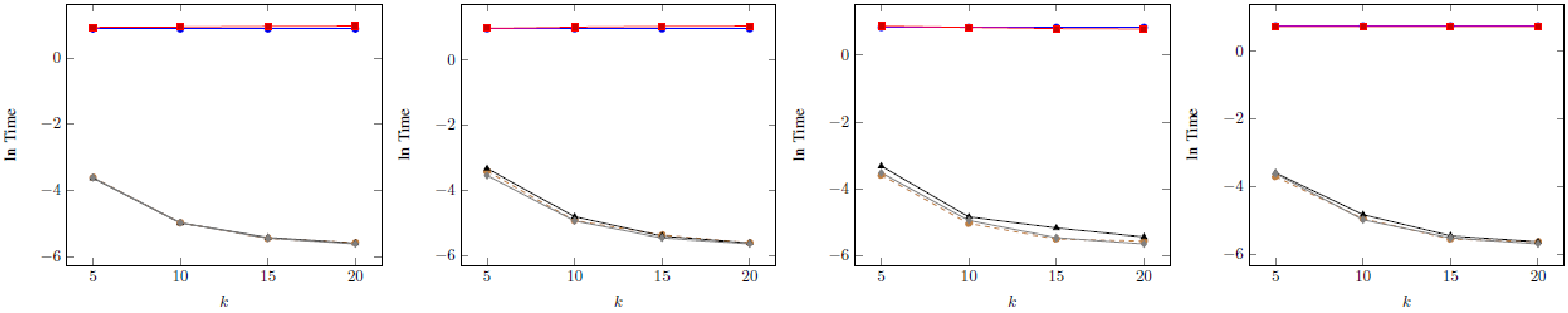
SR: Sampling with replacement
SWR: Sampling without replacement

*Miao, Yun-Qian, Ahmed K. Farahat, and Mohamed S. Kamel. "Ensemble Kernel Mean Matching." *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015.

k = number of test data split
m = training sample size
For uniformity: $k\ (\propto \frac{1}{m})$

Results

Execution time with different sample size

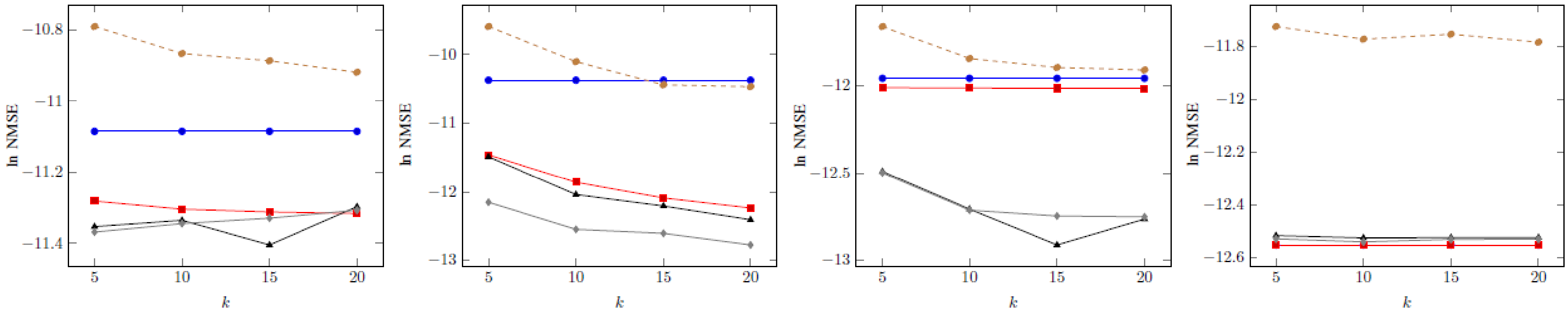


(a) ForestCover (b) KDD (c) Syn002 (d) MNIST

$$\text{NMSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{\beta}(\mathbf{x}^{(i)})}{\sum_{j=1}^{n}\hat{\beta}(\mathbf{x}^{(j)})} - \frac{\beta(\mathbf{x}^{(i)})}{\sum_{j=1}^{n}\beta(\mathbf{x}^{(j)})}\right)$$

Results

NMSE with different sample size

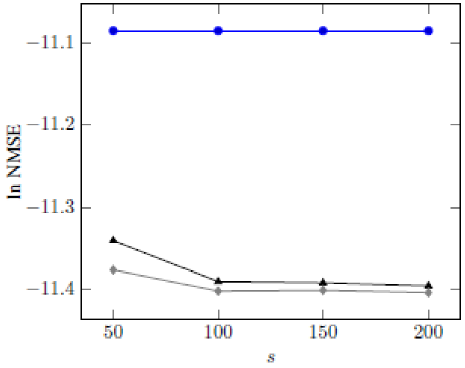


(a) ForestCover (b) KDD (c) Syn002 (d) MNIST

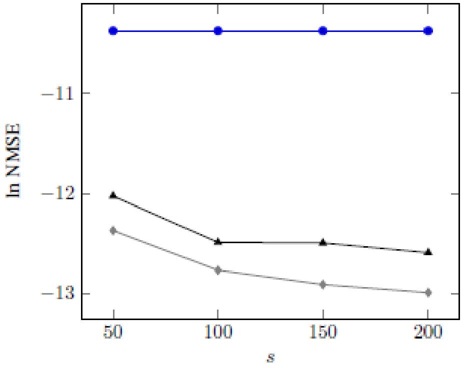
CENKMM ENSKMM ENSTRKMM EVFKMM VFKMM

Results

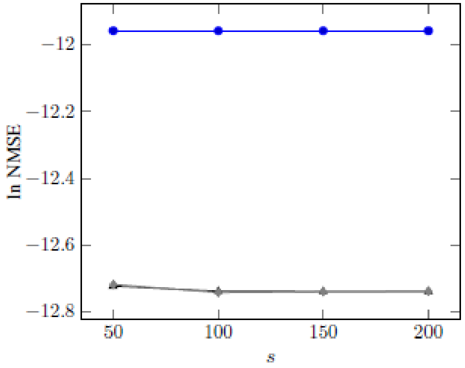
NMSE with different number of samples



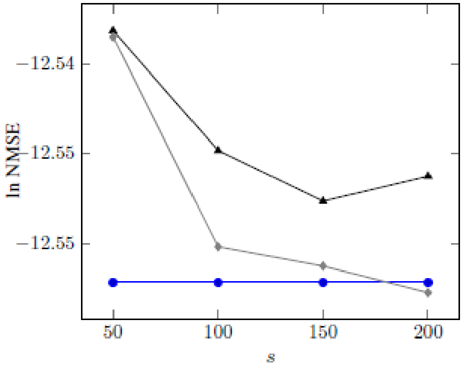
(a) ForestCover



(b) KDD



(c) Syn002

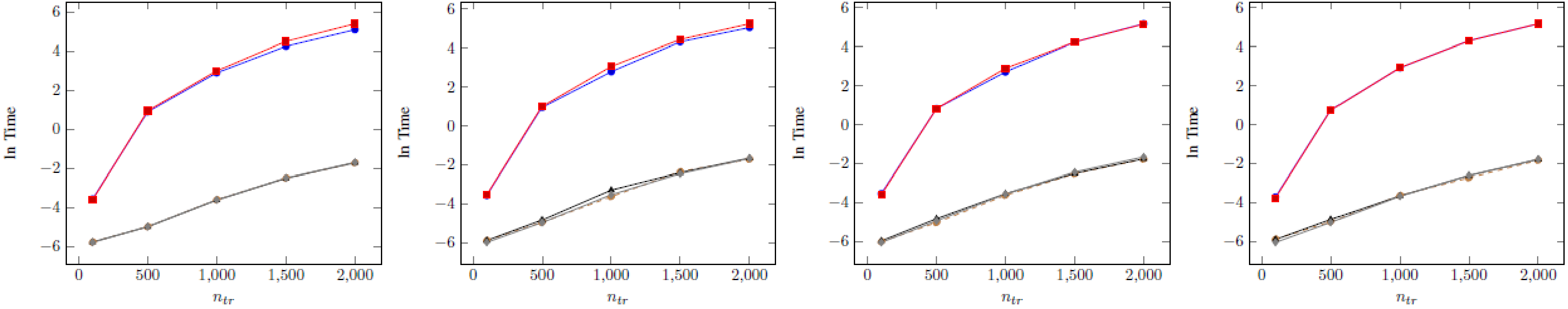


(d) MNIST

CENKMM ENSKMM ENSTRKMM EVFKMM VFKMM

Results

Execution time with different training dataset size

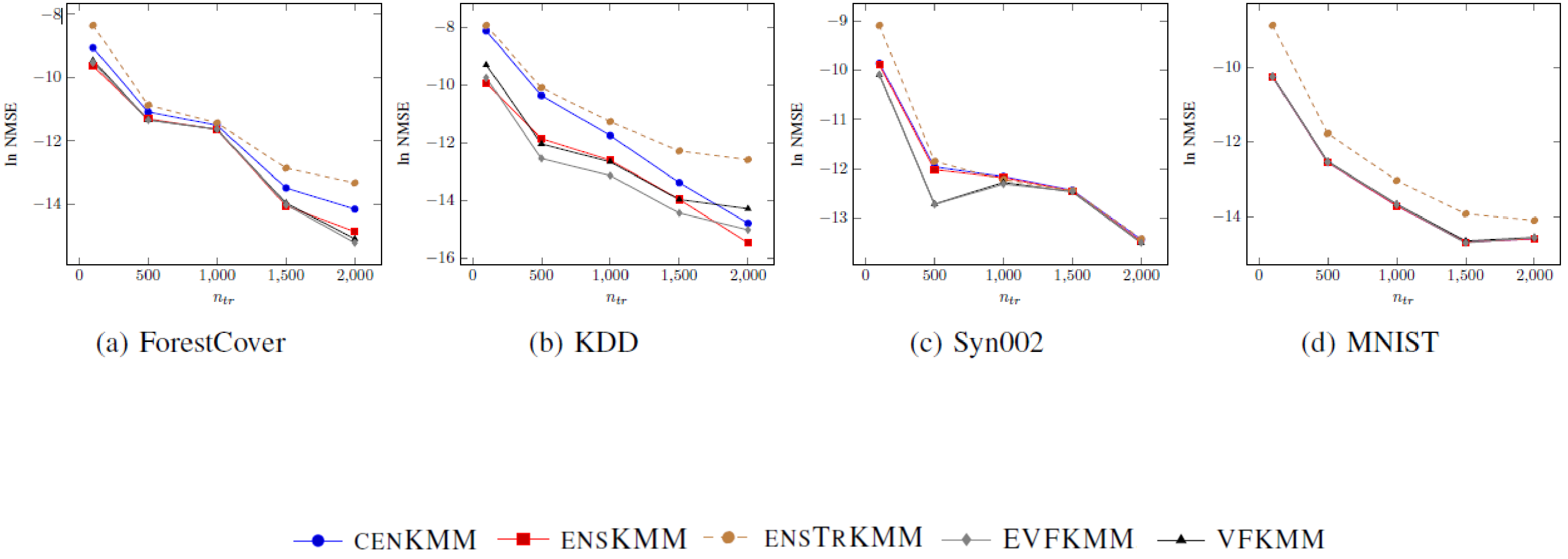


(a) ForestCover (b) KDD (c) Syn002 (d) MNIST

cenKMM ensKMM ensTrKMM EVFKMM VFKMM

Results

NMSE with different training dataset size



Conclusion

- Scalable sampling-based method for Kernel Mean Matching
  - Use M/N bootstrap sampling to generate training data
  - Combine training data instance weights
- Fully scalable KMM
  - Sampling over training dataset
  - Splitting of test dataset.
- Empirical results show large improvements in execution time with similar error.