Sample Selection Bias

Lei Tang

Feb. 20th, 2007

Classical ML vs. Reality

- Training data and Test data share the same distribution (In classical Machine Learning)
- But that's not always the case in reality.
  - Survey data
  - Species habitat modeling based on data of only one area
  - Training and test data collected by different experiments
  - Newswire articles with timestamps

Sample selection bias

- Standard setting: data (x,y) are drawn independently from a distribution D
- If the selected samples is not a random samples of D, then the samples are biased.
- Usually, training data are biased, but we want to apply the classifier to unbiased samples.

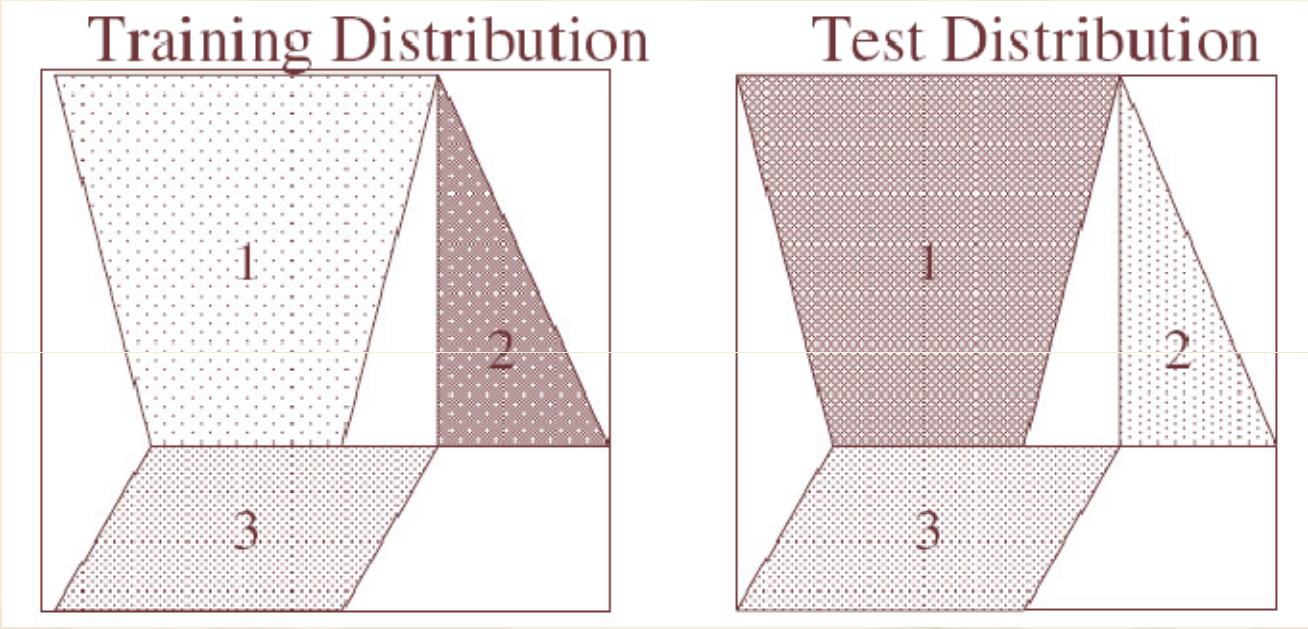
Four cases of Bias(1)

- Let *s* denote whether or not a sample is selected.
- $P(s=1|x,y) = P(s=1)$ (not biased)
- $P(s=1|x,y) = P(s=1|x)$ (depending only on the feature vector)
- $P(s=1|x,y) = P(s=1|y)$ (depending only on the class label)
- $P(s=1|x,y)$ (depending on both x and y)

Four cases of Bias(2)

- $P(s=1|x, y)= P(s=1|y)$: learning from imbalanced data. Can alleviate the bias by changing the class prior.
- $P(s=1|x,y) = P(s=1|x)$ imply $P(y|x)$ remain unchanged. This is mostly studied.
- If the bias depends on both x and y, lack information to analyze.

An intuitive Example



P(s=1|x,y) = P(s=1|x) => s and y are independent.
So P(y|x, s=1) = P(y|x).
Does it really matter as P(y|x) remain unchanged??

Bias Analysis for Classifiers(1)

- Logistic Regression

$$P(y=1|x,s=1)=\frac{1}{1+\exp(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}$$

Any classifiers directly models P(y|x) won't be affected by bias

- Bayesian Classifier

$$\frac{P(x|y,s=1)P(y|s=1)}{P(x|s=1)}=P(y|x,s=1)=P(y|x)$$

But for naïve Bayesian classifier

$$\frac{P(x_1|y,s=1)\ldots P(x_n|y,s=1)P(y|s=1)}{P(x|s=1)},$$

Bias Analysis for Classifiers(2)

- Hard margin SVM: no bias effect.

  Soft margin SVM: has bias effect as the cost of misclassification might change.
- Decision Tree usually results in a different classifier if the bias is presented
- In sum, most classifiers are still sensitive to the sample bias.
- This is in asymptotic analysis assuming the samples are “enough”

Correcting Bias

- Expected Risk:

$$R[\Pr, \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim\Pr}\left[l(x, y, \theta)\right]$$

- Suppose training set from *Pr*, test set from *Pr'*

$$R[\Pr', \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim\Pr'}\left[l(x, y, \theta)\right] = \mathbf{E}_{(x,y)\sim\Pr}\left[\underbrace{\frac{\Pr'(x,y)}{\Pr(x,y)}}_{:=\beta(x,y)} l(x, y, \theta)\right]$$

$$= R[\Pr, \theta, \beta(x, y) l(x, y, \theta)],$$

- So we minimize the empirical regularized risk:

$$R_{\mathrm{reg}}[Z, \beta, l(x, y, \theta)] := \frac{1}{m}\sum_{i=1}^{m} \beta_i l(x_i, y_i, \theta) + \lambda\Omega[\theta],$$

Estimate the weights

- The samples which are likely to appear in the test data will gain more weight.
- But how to estimate the weight of each sample?

$$\left.\begin{array}{l} Pr(x,y) = Pr(x)Pr(y|x) \\ Pr'(x,y) = Pr'(x)Pr'(y|x) \end{array}\right\} \Longrightarrow \beta(x,y) = \frac{Pr'(x)}{Pr(x)}$$

- Brute force approach:
  - Estimate the density of Pr(x) and Pr'(x), respectively,
  - Then calculate the sample weight.
- Not applicable as density estimation is more difficult than classification given limited number of samples.
- Existing works use simulation experiments in which both Pr(x) and Pr'(x) are known (NOT REALISTIC)

Distribution Matching

- The expectation in feature space:

$$\mu(\Pr) := \mathbf{E}_{x \sim \Pr(x)} [\Phi(x)]$$

- We have $Pr = Pr' \iff ||\mu(Pr) - \mu(Pr')|| = 0$
- Hence, the problem can be formulated as

$$\underset{\beta}{\text{minimize}} \left\| \mu(\Pr') - \mathbf{E}_{x \sim \Pr(x)} [\beta(x)\Phi(x)] \right\|$$

$$\text{subject to } \beta(x) \geq 0 \text{ and } \mathbf{E}_{x \sim \Pr(x)} [\beta(x)] = 1$$

- Solution is: $Pr'(x) = \beta(x) Pr(x)$

Empirical KMM optimization

$$\left\| \frac{1}{m} \sum_{i=1}^{m} \beta_i \Phi(x_i) - \frac{1}{m'} \sum_{i=1}^{m'} \Phi(x'_i) \right\|^2 = \frac{1}{m^2} \beta^\top K \beta - \frac{2}{m^2} \kappa^\top \beta + \text{const.}$$

where $K_{ij} := k(x_i, x_j)$ and $\kappa_i := \frac{m}{m'} \sum_{j=1}^{m'} k(x_i, x'_j)$

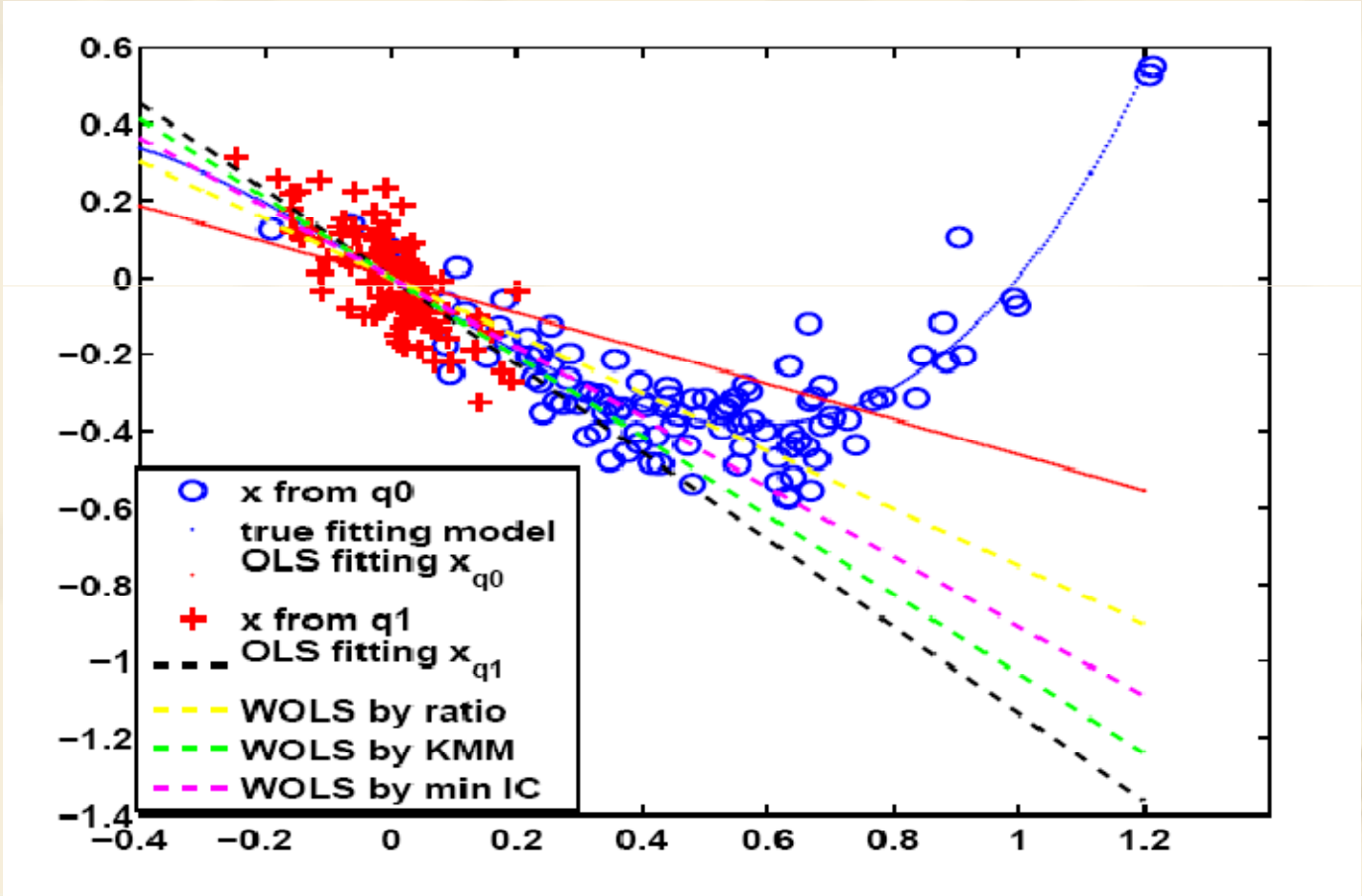
Therefore, it's equivalent to solve the QP problem:

$$\underset{\beta}{\text{minimize}} \ \frac{1}{2} \beta^\top K \beta - \kappa^\top \beta$$

$$\text{subject to } \beta_i \in [0, B] \text{ and } \left| \sum_{i=1}^{m} \beta_i - m \right| \leq m\epsilon.$$

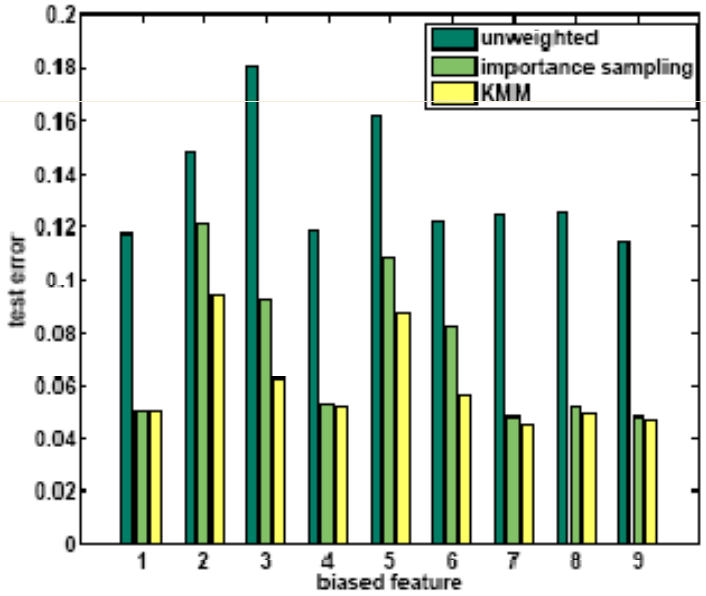
Experiments

- A Toy Regression Example

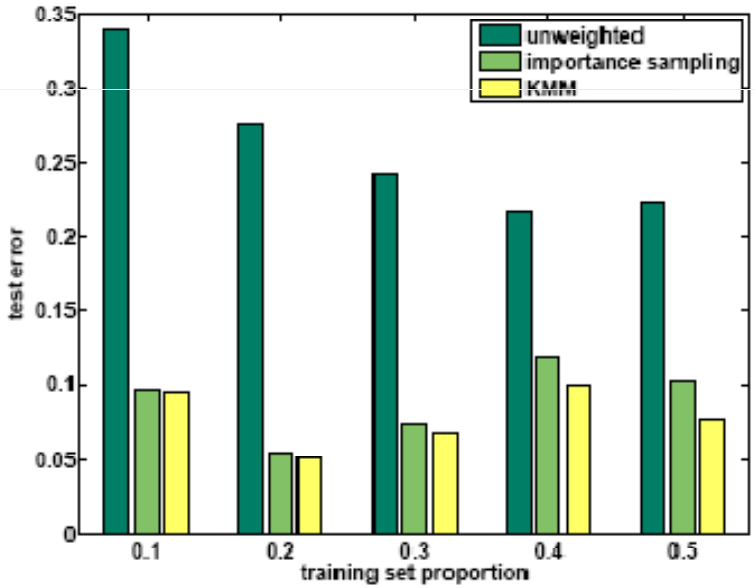


Simulation

- Select some UCI datasets to inject some sample selection bias into training, then test on unbiased samples.

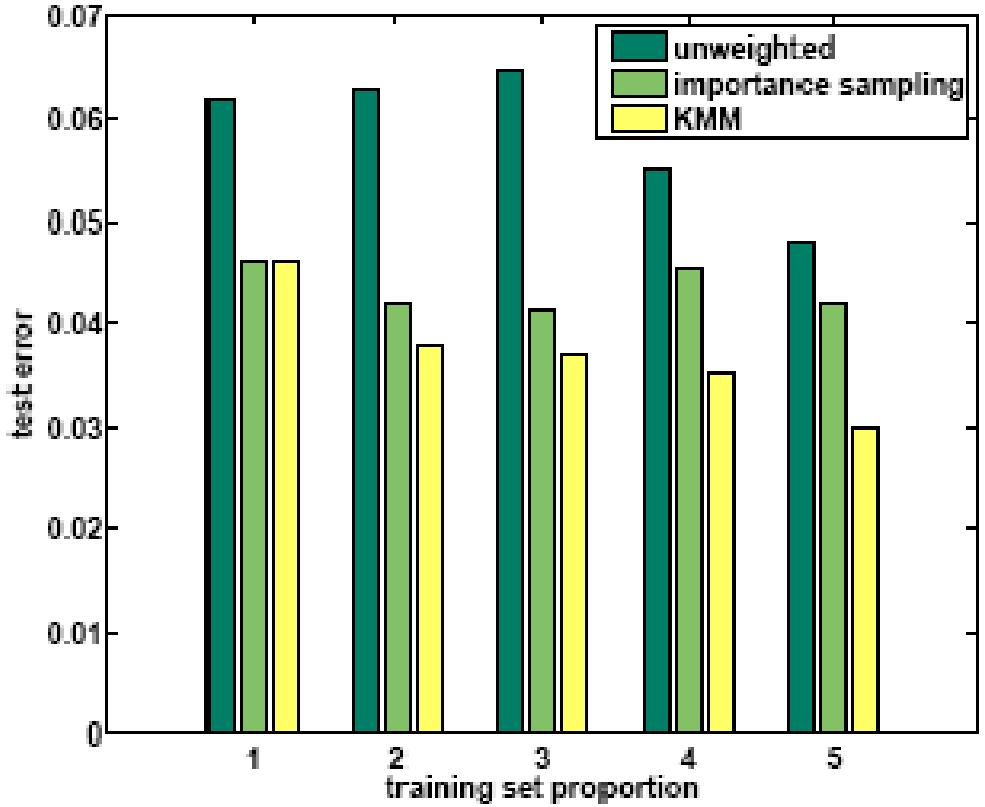


(a) Simple bias on features



(b) Joint bias on features

Bias on Labels



(c) Bias on labels

Unexplained

- From theory, the importance sampling should be the best, why KMM performs better?
- Why kernel methods? Can we just do the matching using input features?
- Can we just perform a logistic regression to estimate \beta by treating test data as positive class, and training data as negative. Then, \beta is the odds.

Some Related Problems

- Semi-supervised Learning (Is it equivalent??)
- Multi-task Learning: assume P(y|x) to be different. But sample selection bias(mostly) assume P(y|x) to be the same. MTL requires training data for each task.
- Is it possible to discriminate features which introduce the bias? Or find invariant dimensionalities?

Any Questions?

Happy Pig Year!