Sample Selection Bias Correction

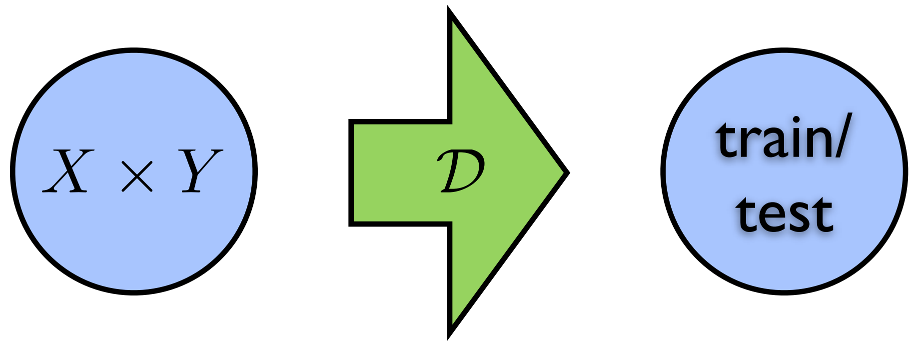
Afshin Rostamizadeh

Joint work with:
Corinna Cortes, Mehryar Mohri & Michael Riley

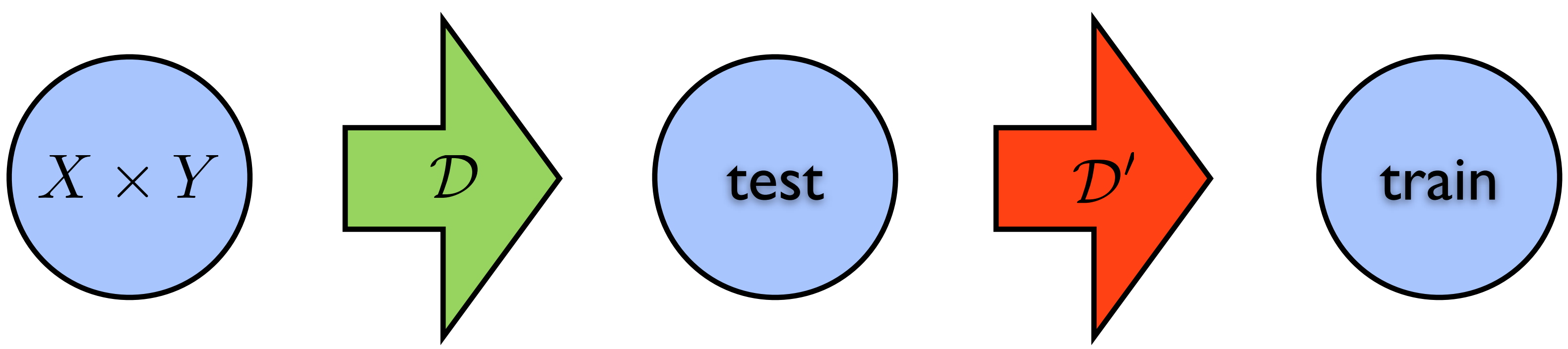
Courant Institute & Google Research

Motivation

- Critical Assumption: Samples for training are drawn according to the target distribution.



- Does not necessarily hold in practice!



Sample Selection Bias: Only a biased sub-sample is available for training. (Heckman '79)

Example

- Building a classifier to detect disease.
- Features: age, weight, height, family history, etc...
- Labels: presence/absence of disease
- Training Set: People who are voluntarily tested.
- Bias: Volunteers are probably at risk for the disease! NOT representative of general population.

Motivation

- Approach: Re-weight sample points to account for bias.
- Important question we address:
  - How do imperfections in re-weighting effect algorithm accuracy?
- Related questions:
  - How well does the re-weighting reconstitute the target distribution?
  - How does one compare different re-weighting algorithms?

Sample Bias Correction

- Model bias with additional random variable s (as in Zadrozny et al., 2003).

$$\forall z \in X \times Y \qquad \Pr_{\mathcal{D}'}[z] = \Pr_{\mathcal{D}}[z|s=1]$$

- Using Bayes' rule, we find re-weighting factors.

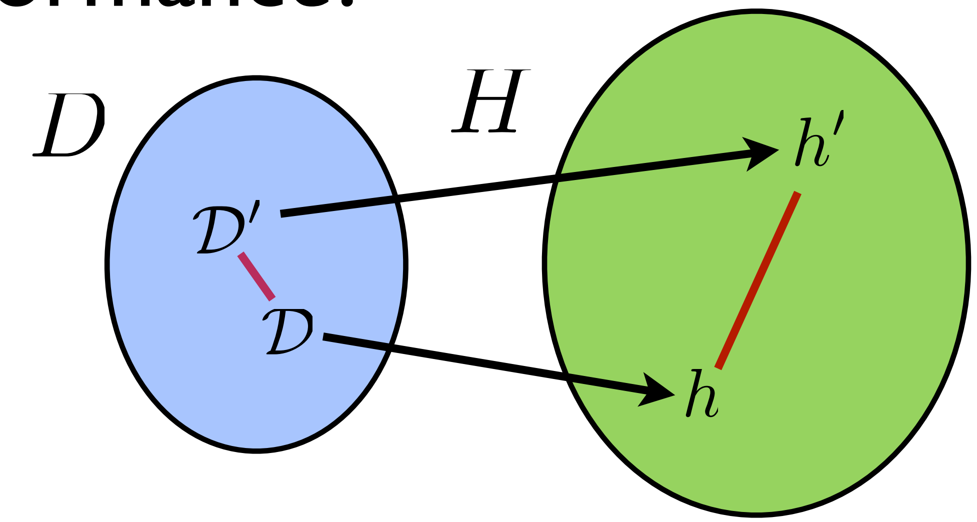
$$\Pr_{\mathcal{D}}[z] = \frac{\Pr[z|s=1]\Pr[s=1]}{\Pr[s=1|z]} = \frac{\Pr[s=1]}{\Pr[s=1|z]} \Pr_{\mathcal{D}'}[z]$$

- We will assume bias is independent of label.

$$\Pr_{\mathcal{D}}[z] = \underbrace{\frac{\Pr[s=1]}{\Pr[s=1|x]}}_{w_x} \Pr_{\mathcal{D}'}[z]$$

Estimate Performance

- How does our empirical estimate effect performance?



Small error in distribution weighting = large deviation in hypothesis selection?

- Analysis follows two main steps:
  - Introduction of distributional stability.
  - Analysis of distributional stable algorithms, under imperfect re-weighting.

Distributional Stability

- This definition is an extension of point-based stability (Bousquet & Elisseeff, 2002).
- Given divergence measure *d* and cost function *c*, an algorithm is *β-distributionally stable* if for two hypotheses $h_{\mathcal{W}}$ and $h_{\mathcal{W}'}$ produced by weighted samples $S_{\mathcal{W}}$ and $S_{\mathcal{W}'}$ the following bound holds,

$$\boxed{\forall z \in X \times Y, \quad |c(h_{\mathcal{W}}, z) - c(h_{\mathcal{W}'}, z)| \leq \beta d(\mathcal{W}, \mathcal{W}')}$$

- Implies,

$$|R(h_{\mathcal{W}}) - R(h_{\mathcal{W}'})| \leq \beta d(\mathcal{W}, \mathcal{W}')$$

Distributional Stability

- What type of algorithms are distributional stable?
- First some definitions,
    - *sigma-admissible* cost function:

$$|c(h,z) - c(h',z)| \leq \sigma |h(x) - h'(x)|$$

    - bounded kernel function:

$$\infty > \kappa \geq K(x,x), \;\; \forall x \in X$$

    - define the maximum eigenvalue of the kernel matrix as: $\lambda_{\max}(\mathbf{K})$

Distributional Stability

- We show that regularized kernel algorithms of the type:

$$\min_{h \in H} \widehat{R}_{\mathcal{W}}(h) + \lambda \|h\|_K^2, \quad \text{where } \widehat{R}_{\mathcal{W}}(h) = \frac{1}{m} \sum_{i=1}^{m} \underline{w_i} c(h, x_i)$$

Weight sensitive empirical error.

are stable, with the following coefficients:

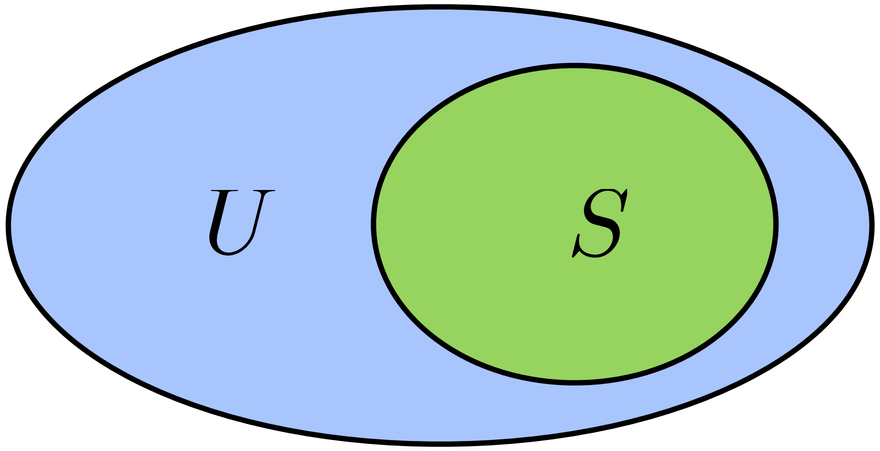
$$\boxed{\begin{array}{c} \ell_1 \text{ distance :} \\ \beta \leq \dfrac{\sigma^2 \kappa^2}{2\lambda} \end{array}} \qquad \boxed{\begin{array}{c} \ell_2 \text{ distance :} \\ \beta \leq \dfrac{\sigma^2 \kappa \lambda_{\max}^{\frac{1}{2}}(\mathbf{K})}{2\lambda} \end{array}}$$

- The ℓ_1 bound coincides with point-based stability.

Empirical Estimates

- How do we estimate re-weighting factors?
- Make use of unlabeled data.

For example, partially labeled sample:
$U = (x_1, \ldots, x_n)$



- One simple method, use counts (histogram).

$$\text{Given sample,} \quad U = (x_1, \ldots, x_n)$$

$$\Pr[s = 1 | x] \approx \frac{m_x}{n_x} = \frac{|\{x_i : x_i = x, x_i \in S\}|}{|\{x_i : x_i = x, x_i \in U\}|}$$

Empirical Estimates

- Other methods:
  - Kernel Mean Matching (Huang et al., 2006)
  - Discriminative Methods (Bickel et al., 2007)
- Density Estimation Methods
  - Kernel Density Estimation
  - Logistic Regression
- However Note: No need to generalize, need to reweight only training points.

Weight Estimation Error

- For *distinct* point x equal to *sampled* point x_i define $p(x_i) = \Pr[s = 1|x]$ and $\hat{p}(x_i) = m_x/n_x$.
- Thus, perfectly and estimated re-weightings are written (modulo constant).

$$\mathcal{W}(x_i) = \frac{1}{m}\frac{1}{p(x_i)} \qquad \widehat{\mathcal{W}}(x_i) = \frac{1}{m}\frac{1}{\hat{p}(x_i)}$$

- Define $p_0 = \min_{x \in U} \Pr[x] \neq 0$, m' as # of distinct labeled points and $B = \max_{i=1,\ldots,m} \max(1/p(x_i), 1/\hat{p}(x_i))$ then w.h.p. we show,

$$l_1 \leq B^2 \sqrt{\frac{\log 2m' + \log \frac{1}{\delta}}{p_0 n}}, \; l_2 \leq B^2 \sqrt{\frac{\log 2m' + \log \frac{1}{\delta}}{p_0 n m}}$$

Final Bound

- For any regularization algorithm based on L2 norm and count-based weights, the following bounds hold with probability (1 - δ).

$$|R(h_{\mathcal{W}}) - R(h_{\widehat{\mathcal{W}}})| \leq \frac{\sigma^2\kappa^2 B^2}{2\lambda}\sqrt{\frac{\log 2m' + \log\frac{1}{\delta}}{p_0 n}}$$

$$|R(h_{\mathcal{W}}) - R(h_{\widehat{\mathcal{W}}})| \leq \frac{\sigma^2\kappa\lambda_{\max}^{\frac{1}{2}}(\mathbf{K})B^2}{2\lambda}\sqrt{\frac{\log 2m' + \log\frac{1}{\delta}}{p_0 nm}}.$$

- For kernel matrix with bounded eigenvalue, L2 based bound converges at least as fast.

Empirical Results

- We use several public regression data sets, and artificially introduce bias into training sample.

Data set	$\lvert U \rvert$	$\lvert S \rvert$	n_{test}	Unweighted	Ideal	Clustered
abalone	2000	724	2177	$.654 \pm .019$	$.551 \pm .032$	$.623 \pm .034$
bank32nh	4500	2384	3693	$.903 \pm .022$	$.610 \pm .044$	$.635 \pm .046$
bank8FM	4499	1998	3693	$.085 \pm .003$	$.058 \pm .001$	$.068 \pm .002$
cal-housing	16512	9511	4128	$.395 \pm .010$	$.360 \pm .009$	$.375 \pm .010$
cpu-act	4000	2400	4192	$.673 \pm .014$	$.523 \pm .080$	$.568 \pm .018$
cpu-small	4000	2368	4192	$.682 \pm .053$	$.477 \pm .097$	$.408 \pm .071$
housing	300	116	206	$.509 \pm .049$	$.390 \pm .053$	$.482 \pm .042$
kin8nm	5000	2510	3192	$.594 \pm .008$	$.523 \pm .045$	$.574 \pm .018$
puma8NH	4499	2246	3693	$.685 \pm .013$	$.674 \pm .019$	$.641 \pm .012$

- Since we know exact cause of the (artificial) bias, we can compare to ideal re-weighting.

Summary

- Problem of sample selection bias occurs often in practice in important applications.
- We give general methods to measure effectiveness of re-weighting methods.
- Theory inspires new algorithms that work well in practice.
- Future work - Can we combine learning and

Thank You!

Proofs

- L1-convergence of count-based estimate:
  - Claim: with probability as least (1 - δ) simultaneously for all *x*,

$$\left|\Pr[s=1|x] - \frac{m_x}{n_x}\right| \le \sqrt{\frac{\log 2m' + \log \frac{1}{\delta}}{p_0 n}}$$

  - Proof: By Hoeffding's inequality for fixed x,

$$\Pr_U\left[\left|\Pr[s=1|x] - \frac{m_x}{n_x}\right| \ge \epsilon\right]$$

$$= \sum_{i=1}^{n} \Pr_x\left[\left|\Pr[s=1|x] - \frac{m_x}{i}\right| \ge \epsilon \mid n_x = i\right] \Pr[n_x = i]$$

$$\le \sum_{i=1}^{n} 2e^{-2i\epsilon^2} \Pr_U[n_x = i]$$

Proofs

- Note that n_x is a binomial r.v. with parameters $\Pr_U[x] = p_x$ and n.

$$2 \sum_{i=1}^{n} e^{-2i\epsilon^2} \Pr_U[n_x = i]$$

$$\leq 2 \sum_{i=0}^{n} e^{-2i\epsilon^2} \binom{n}{i} p_x^i (1 - p_x)^{n-i} = 2(p_x e^{-2\epsilon^2} + (1 - p_x))^n$$

$$= 2(1 - p_x(1 - e^{-2\epsilon^2}))^n \leq 2 \exp(-p_x n(1 - e^{-2\epsilon^2})).$$

- Using the fact $1 - e^{-x} \geq x/2$ for $x \in [0, 1]$ shows,

$$\Pr_U \left[\left| \Pr[s = 1|x] - \frac{m_x}{n_x} \right| \geq \epsilon \right] \leq 2e^{-p_x n \epsilon^2}$$

Proofs

- Taking a union bound over the m' distinct points in the training set completes the claim.
- Now to bound the L2 distance,

$$
\begin{aligned}
l_2^2(\mathcal{W}, \widehat{\mathcal{W}}) &= \frac{1}{m^2} \sum_{i=1}^{m} \left(\frac{1}{p(x_i)} - \frac{1}{\hat{p}(x_i)} \right)^2 \\
&= \frac{1}{m^2} \sum_{i=1}^{m} \left(\frac{p(x_i) - \hat{p}(x_i)}{p(x_i)\hat{p}(x_i)} \right)^2 \\
&\leq \frac{B^4}{m} \max_i (p(x_i) - \hat{p}(x_i))^2.
\end{aligned}
$$

- Using the previous claim completes the proof.