NeurIPS DistShift Workshop 2021 Dec. 13, 2021

Importance Weighting for Transfer Learning

Masashi Sugiyama

RIKEN Center for Advanced Intelligence Project/
The University of Tokyo

http://www.ms.k.u-tokyo.ac.jp/sugi/

RIKEN

東京大学
THE UNIVERSITY OF TOKYO

Transfer Learning

Given:

- Training data $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$

$\boldsymbol{x}$: Input

y : Output

Goal:

- Train a predictor $y = f(\boldsymbol{x})$
  that works well in the test domain
  (with some additional data from the test domain).

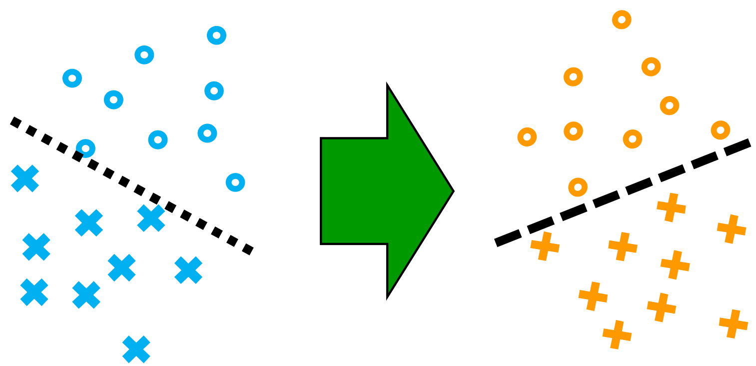
$$\min_f R(f) \qquad R(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)]$$

ℓ : loss function

Challenge:

- Overcome changing distributions!

$$p_{\mathrm{tr}}(\boldsymbol{x}, y) \neq p_{\mathrm{te}}(\boldsymbol{x}, y)$$



NIPS Workshop on Learning when Test and Training Inputs Have Different Distributions, Whistler 2006

Workshop

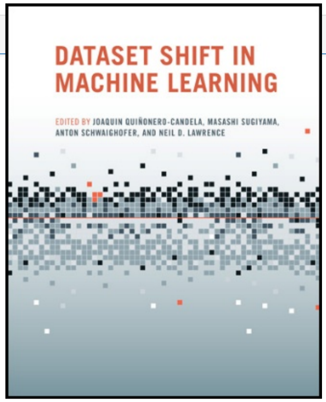
Learning when test and training inputs have different distributions

Joaquin Quiñonero Candela · Masashi Sugiyama · Anton Schwaighofer · Neil D Lawrence

Sat Dec 09 05:00 PM -- 05:00 PM (JST) @ Nordic

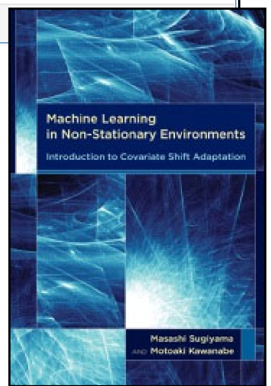
Event URL: http://ida.first.fraunhofer.de/projects/different06/ »

Many machine learning algorithms assume that the training and the test data are drawn from the same distribution. Indeed many of the proofs of statistical consistency, etc., rely on this assumption. However, in practice we are very often faced with the situation where the training and the test data both follow the same conditional distribution, p(y|x), but the input distributions, p(x), differ. For example, principles of experimental design dictate that training data is acquired in a specific manner that bears little resemblance to the way the test inputs may later be generated. The aim of this workshop will be to try and shed light on the kind of situations where explicitly addressing the difference in the input distributions is beneficial, and on what the most sensible ways of doing this are.



Quiñonero-Candela, Sugiyama, Schwaighofe & Lawrence (Eds.), Dataset Shift in Machine Learning, MIT Press, 2009.

Sugiyama & Kawanabe, Machine Learning in Non-Stationary Environments, MIT Press, 2012



Learning when Training and Test Inputs Have Different Distributions

Saturday December 9, 2006

Org: Joaquin Quiñonero-Candela, Anton Schwaighofer, Neil Lawrence & Masashi Sugiyama

Morning session: 7:30am–10:30am

7:30am **Opening**, *The organizers*

7:40am **When Training and Test Distributions are Different: Characterising Learning Transfer**, *Amos Storkey, University of Edinburgh*

8:10am **Can Adaptive Regularization Help?**, *Matthias Hein, Max Planck Institute for Biological Cybernetics*

8:40am *coffee break*

8:50am **Learning Classifiers in Distribution and Cost-sensitive Environments**, *Nitesh Chawla, University of Notre Dame*

9:20am **Optimality of Bayesian Transduction - Implications for Input Non-stationarity**, *Lars Kai Hansen, Technical University of Denmark*

9:50pm **Estimating the Joint AUC of Labelled and Unlabelled Data**, *Thomas Gärtner, Gemma Garriga, Thorsten Knopp, Peter Flach and Stefan Wrobel*

10:10am **A Domain Adaptation Formal Framework Addressing the Training/Test Distribution Gap**, *Shai Ben-David, University of Waterloo and John Blitzer, University of Pennsylvania*

Afternoon session: 3:30pm–6:30pm

3:30pm **Projection and Projectability**, *David Corfield, Max Planck Institute for Biological Cybernetics*

4:00pm **Using features of probability distributions to achieve covariate shift**, *Arthur Gretton, MPI for Biol. Cyb. and Alex Smola, National ICT Australia*

4:20pm **Active Learning, Model Selection and Covariate Shift**, *Masashi Sugiyama, Tokyo Institute of Technology*

4:50pm *coffee break*

5:00pm **Visualizing Pairwise Similarity via Semidefinite Programming**, *Amir Globerson, MIT, and Sam Roweis, University of Toronto*

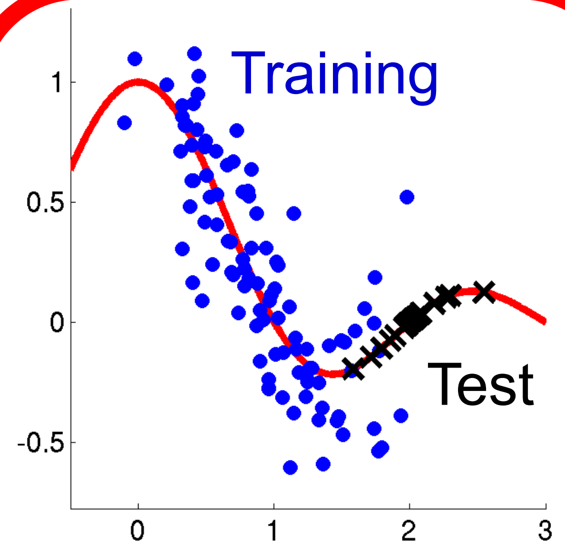
5:20pm **A Divergence Prior for Adaptive Learning**, *Xiao Li and Jeff Bilmes, University of Washington*

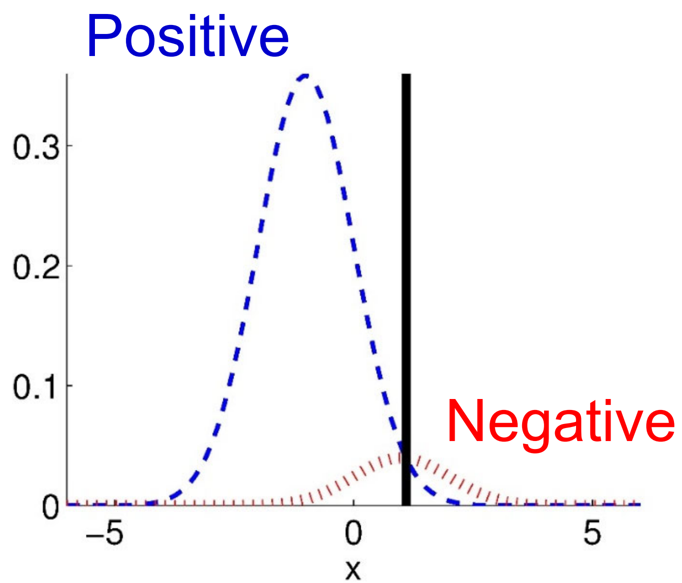
5:40pm **discussion**, *everyone*

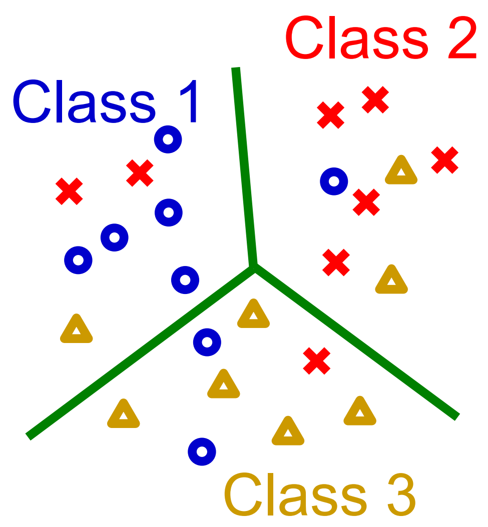
Various Scenarios

$\boldsymbol{x}$: Input y : Output

- Full-distribution shift: $p_{\mathrm{tr}}(\boldsymbol{x}, y) \neq p_{\mathrm{te}}(\boldsymbol{x}, y)$
- Covariate shift: $p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$
- Class-prior shift: $p_{\mathrm{tr}}(y) \neq p_{\mathrm{te}}(y)$
- Output noise: $p_{\mathrm{tr}}(y|\boldsymbol{x}) \neq p_{\mathrm{te}}(y|\boldsymbol{x})$
- Class-conditional shift: $p_{\mathrm{tr}}(\boldsymbol{x}|y) \neq p_{\mathrm{te}}(\boldsymbol{x}|y)$







Contents

1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

Regression under Covariate Shift

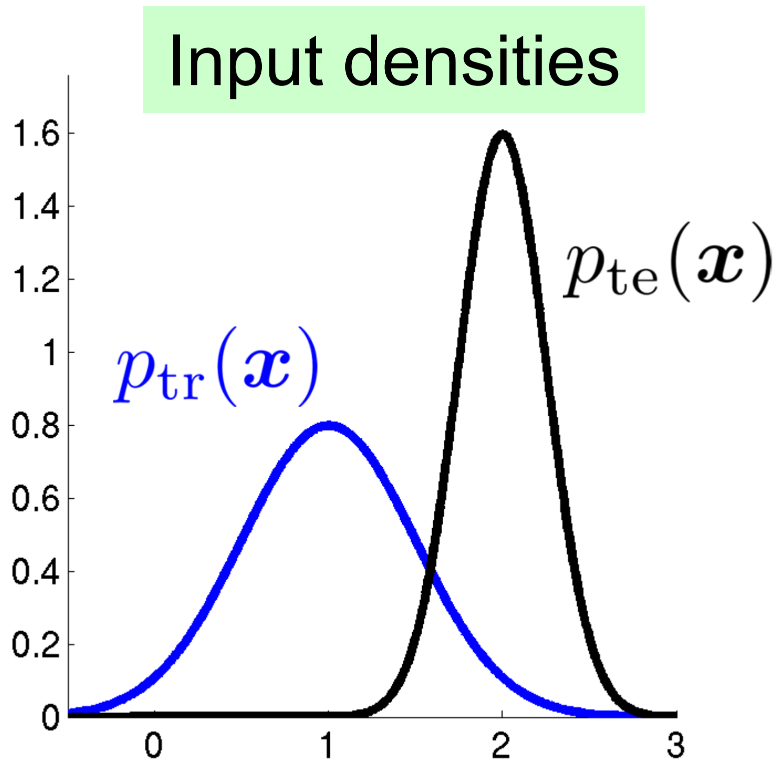
Covariate shift: Shimodaira (JSPI2000)

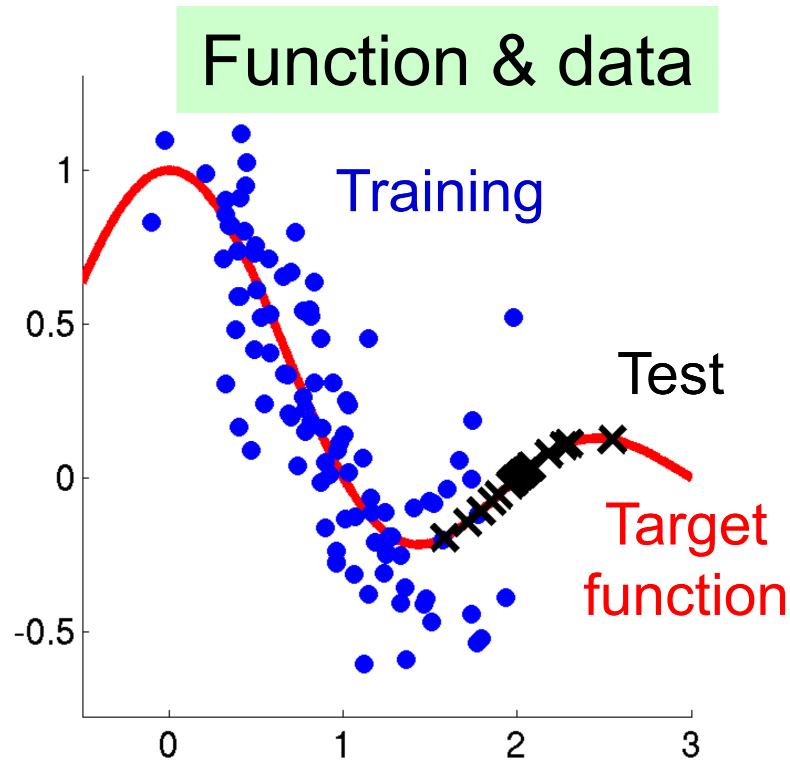
- Training and test input distributions are different:

$$p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$$

- But the output-given-input distribution remains unchanged:

$$p_{\mathrm{tr}}(y|\boldsymbol{x}) = p_{\mathrm{te}}(y|\boldsymbol{x}) = p(y|\boldsymbol{x})$$

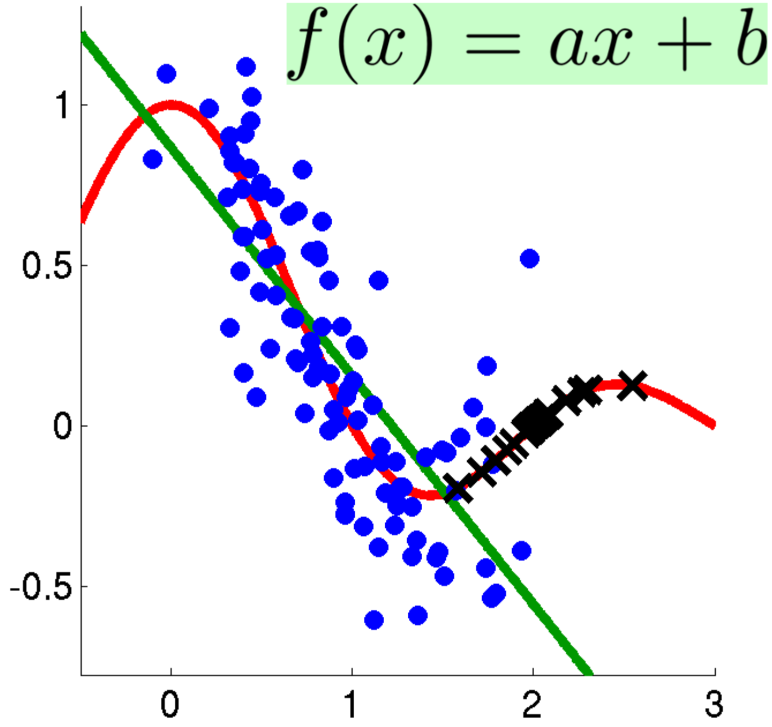




Empirical Risk Minimization (ERM)

$$\operatorname*{argmin}_{f \in \mathcal{F}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

$$\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$$



- Generally, ERM is consistent:
  - Learned function converges to the optimal solution when $n_{\mathrm{tr}} \to \infty$.
- However, covariate shift makes ERM inconsistent:

$$\operatorname*{argmin}_{f \in \mathcal{F}} \left[\frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right] \overset{n_{\mathrm{tr}} \to \infty}{\longrightarrow} \operatorname*{argmin}_{f \in \mathcal{F}} \left[\underbrace{\mathbb{E}_{p(y|\boldsymbol{x}) p_{\mathrm{tr}}(\boldsymbol{x})} [\ell(f(\boldsymbol{x}), y)]}_{\neq R(f)} \right]$$

Contents

1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

Importance-Weighted ERM (IWERM)

$$\underset{f\in\mathcal{F}}{\mathrm{argmin}}\left[\sum_{i=1}^{n_{\mathrm{tr}}}\underbrace{\frac{p_{\mathrm{te}}(\boldsymbol{x}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\boldsymbol{x}_i^{\mathrm{tr}})}}_{\text{Importance}}\ell(f(\boldsymbol{x}_i^{\mathrm{tr}}),y_i^{\mathrm{tr}})\right]$$

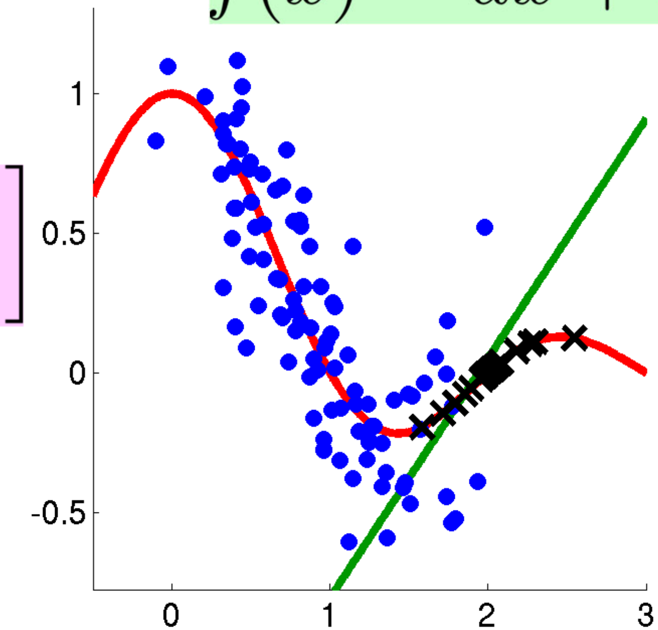
- IWERM is consistent even under covariate shift:

$$\underset{f\in\mathcal{F}}{\mathrm{argmin}}\left[\frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}}\frac{p_{\mathrm{te}}(y_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(y_i^{\mathrm{tr}})}\ell(f(\boldsymbol{x}_i^{\mathrm{tr}}),y_i^{\mathrm{tr}})\right]$$

$$\underset{n_{\mathrm{tr}}\to\infty}{\longrightarrow}\ \underset{f\in\mathcal{F}}{\mathrm{argmin}}\left[\mathbb{E}_{p(y|\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})}\left[\frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}\ell(f(\boldsymbol{x}),y)\right]\right]$$

$$=\underset{f\in\mathcal{F}}{\mathrm{argmin}}\left[\underbrace{\mathbb{E}_{p(y|\boldsymbol{x})p_{\mathrm{te}}(\boldsymbol{x})}[\ell(f(\boldsymbol{x}),y)]}_{=R(f)}\right]$$

$$f(x)=ax+b$$

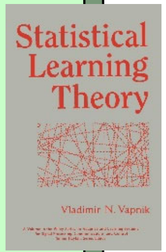


- How can we know the importance weight?

Importance Weight Estimation

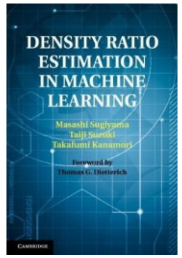
Vapnik's principle: Vapnik (Wiley, 1998)

When solving a problem of interest,
one should not solve a more general problem
as an intermediate step



Knowing densities		Knowing ratio
$p_{\mathrm{te}}(\boldsymbol{x}), p_{\mathrm{tr}}(\boldsymbol{x})$	$\Rightarrow$ / $\cancel{\Leftarrow}$	$r^*(\boldsymbol{x}) = \dfrac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$

- Estimating the density ratio is substantially easier than estimating both the densities!
- Various direct density-ratio estimators were developed.



Sugiyama, Suzuki & Kanamori,
Density Ratio Estimation
in Machine Learning
(Cambridge University Press, 2012)

Least-Squares Importance Fitting (LSIF)

Kanamori, Hido & Sugiyama (JMLR2009)

- Given training and test input data:

$$\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}) \qquad \{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x})$$

- Directly fit a model r to $r^*(\boldsymbol{x}) = \dfrac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$ by LS:

$$\min_r Q(r) \qquad Q(r) = \int \Big(r(\boldsymbol{x}) - r^*(\boldsymbol{x}) \Big)^2 p_{\mathrm{tr}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

  - Empirical approximation:

$$Q(r) = \int r(\boldsymbol{x})^2 p_{\mathrm{tr}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - 2 \int r(\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + C$$

$$\approx \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} r(\boldsymbol{x}_i^{\mathrm{tr}})^2 - \frac{2}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} r(\boldsymbol{x}_j^{\mathrm{te}}) + C$$

Contents

1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

Bias-Variance Trade-Off

- Importance-weighted empirical risk minimizer

$$\operatorname*{argmin}_{f \in \mathcal{F}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \frac{p_{\mathrm{te}}(\boldsymbol{x}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\boldsymbol{x}_i^{\mathrm{tr}})} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

  has no bias, but has large variance.

- The ordinary empirical risk minimizer

$$\operatorname*{argmin}_{f \in \mathcal{F}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

  has small variance (statistically efficient), but has large bias.

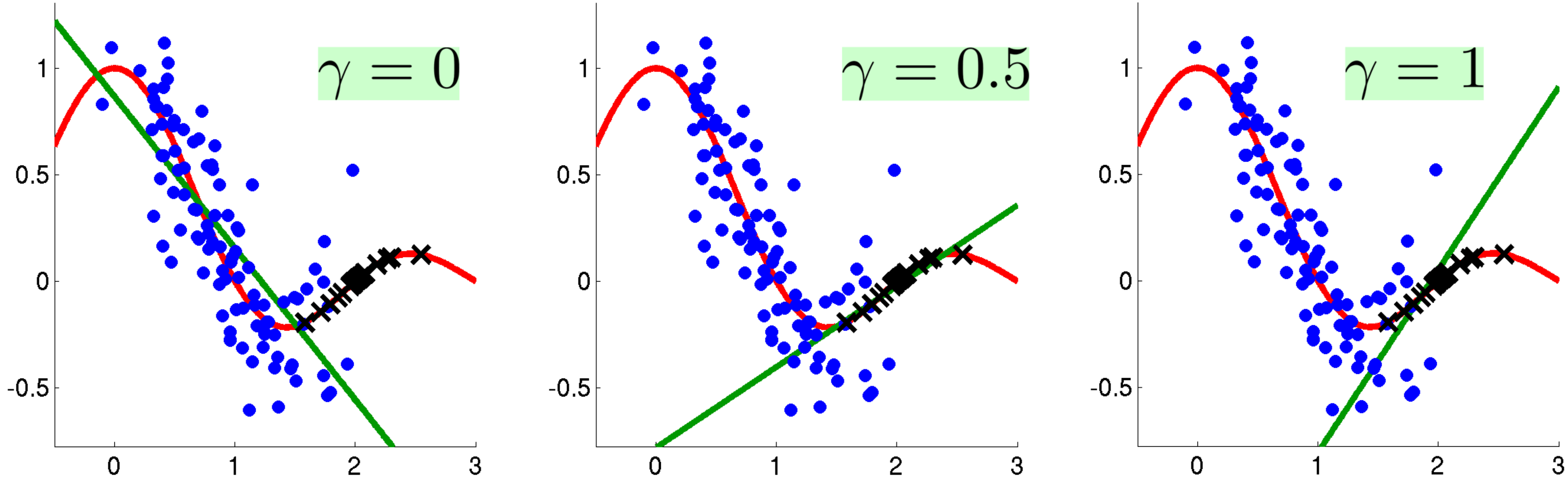
- How can we control the bias-variance trade-off?

Flattened Importance Weighting

$$\min_f \left[\sum_{i=1}^{n_{\mathrm{tr}}} \left(\frac{p_{\mathrm{te}}(\boldsymbol{x}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\boldsymbol{x}_i^{\mathrm{tr}})} \right)^{\gamma} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

Shimodaira (JSPI2000)

$$0 \leq \gamma \leq 1$$



Large bias, small variance (Intermediate) Small bias, large variance

- Flattening factor γ may be chosen by
  - Importance-weighted Akaike information criterion — Shimodaira (JSPI2000)
  - Importance-weighted cross-validation — Sugiyama, Krauledat & Müller (JMLR2007)

Relative Importance Weighting

- Even with direct methods, reliably estimating the importance weight is hard:
  - $r^*(\boldsymbol{x})$ could be highly fluctuated.

$$r^*(\boldsymbol{x}) = \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$$

- Then, flattening unreliable importance estimator $\widehat{r}(\boldsymbol{x})$ by power factor γ is also unreliable.

$$\min_f \left[\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{r}(\boldsymbol{x}_i^{\mathrm{tr}})^\gamma \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

- Let's use relative importance weight:

Yamada, Suzuki, Kanamori, Hachiya & Sugiyama (NIPS2011, NeCo2013)

$$r_\beta(\boldsymbol{x}) = \frac{p_{\mathrm{te}}(\boldsymbol{x})}{\beta p_{\mathrm{tr}}(\boldsymbol{x}) + (1-\beta) p_{\mathrm{te}}(\boldsymbol{x})} \qquad 0 \le \beta \le 1$$

  - Directly estimable for each β by relative LSIF.

Contents

1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

From Two-Step Adaptation to One-Step Adaptation

- The classical approaches are two steps:
  1. Weight estimation (e.g., LSIF):

$$\widehat{r} = \mathop{\mathrm{argmin}}_{r} \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x})}\left[(r(\boldsymbol{x}) - r^*(\boldsymbol{x}))^2\right]$$

  2. Weighted predictor training (e.g., IWERM):

$$\widehat{f} = \mathop{\mathrm{argmin}}_{f} \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x},y)}\left[\widehat{r}(\boldsymbol{x})\ell(f(\boldsymbol{x}), y)\right]$$

- Can we integrate these two steps?

Contents

1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

Joint Upper-Bound Minimization

Zhang et al. (ACML2020, SNCS2021)

- Suppose we are given
  - Labeled training data: $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$
  - Unlabeled test data: $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x})$

- Goal: We want to minimize the test risk.

$$R_\ell(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)]$$

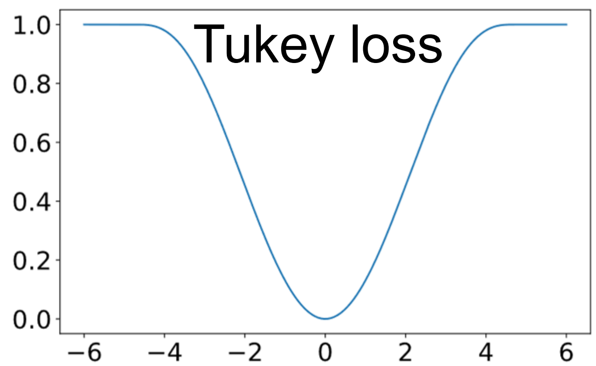
ℓ : evaluation loss

- We use two losses $\ell(\leq 1), \ell'(\geq \ell)$.

ℓ' : surrogate loss

For example:

- ℓ : 0/1, ℓ' : hinge or softmax cross-entropy (classification)
- ℓ : Tukey, ℓ': squared (regression)



Risk Upper-Bounding (cont.)

Zhang et al. (ACML2020, SNCS2021)

- For $\ell \leq 1, \ell' \geq \ell, r \geq 0$,
  the test risk is upper-bounded as

$$\frac{1}{2} R_\ell(f)^2 \leq J_{\ell'}(r, f)$$

$$R_\ell(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)]$$

$$J_{\ell'}(r, f) = (\mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x},y)}[r(\boldsymbol{x})\ell'(f(\boldsymbol{x}), y)])^2 \leftarrow \text{IWERM}$$

$$+\mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x})}[(r(\boldsymbol{x}) - r^*(\boldsymbol{x}))^2] \leftarrow \text{LSIF}$$

- In terms of this upper-bound minimization,
  2-step (LSIF followed by IWERM) is not optimal:
- Let's directly minimize the upper bound w.r.t. r, f !

Theoretical Analysis

- Under some mild conditions, the test risk of the empirical solution $\widehat{f} = \mathop{\mathrm{argmin}}_{f \in \mathcal{F}} \min_{r} \widehat{J}_{\ell'}(r, f)$ is upper-bounded as

$$R_{\ell}(\widehat{f}) \leq \sqrt{2} \min_{f \in \mathcal{F}} R_{\ell'}(f) + \mathcal{O}_p(n_{\mathrm{tr}}^{-1/4} + n_{\mathrm{te}}^{-1/4})$$

$$\widehat{J}_{\ell'}(r, f) = \left(\frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} r(\boldsymbol{x}_i^{\mathrm{tr}}) \ell'(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right)^2 + \left(\frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} r(\boldsymbol{x}_i^{\mathrm{tr}})^2 - \frac{2}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} r(\boldsymbol{x}_j^{\mathrm{tr}}) + C \right)$$

$$\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y) \qquad \{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x})$$

$$R_{\ell}(\widehat{f}) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x}, y)}[\ell(\widehat{f}(\boldsymbol{x}), y)]$$

$$R_{\ell'}(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x}, y)}[\ell'(f(\boldsymbol{x}), y)]$$

Contents

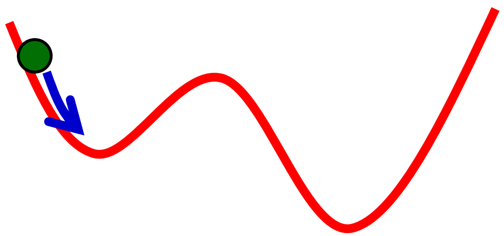
1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

Dynamic Importance Weighting

Fang et al. (NeurIPS2020)

- Deep learning adopts stochastic optimization:

$$f \leftarrow f - \eta \nabla \widehat{R}(f)$$ $\eta > 0$: Learning rate



- Let's learn
  - Importance weight r
  - predictor f

dynamically in the mini-batch-wise manner.

Mini-Batch-Wise Loss Matching

- Suppose we are given
  - (Large) labeled training data: $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$
  - (Small) labeled test data: $\{(\boldsymbol{x}_j^{\mathrm{te}}, y_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x}, y)$
- For each mini-batch $\{(\bar{\boldsymbol{x}}_i^{\mathrm{tr}}, \bar{y}_i^{\mathrm{tr}})\}_{i=1}^{\bar{n}_{\mathrm{tr}}}, \{(\bar{\boldsymbol{x}}_j^{\mathrm{te}}, \bar{y}_j^{\mathrm{te}})\}_{j=1}^{\bar{n}_{\mathrm{te}}}$ importance weights are estimated by kernel mean matching for loss values:

Huang, et al. (NeurIPS2007)

$$\frac{1}{\bar{n}_{\mathrm{tr}}} \sum_{i=1}^{\bar{n}_{\mathrm{tr}}} r_i \ell(f(\bar{\boldsymbol{x}}_i^{\mathrm{tr}}), \bar{y}_i^{\mathrm{tr}}) \approx \frac{1}{\bar{n}_{\mathrm{te}}} \sum_{j=1}^{\bar{n}_{\mathrm{te}}} \ell(f(\bar{\boldsymbol{x}}_j^{\mathrm{te}}), \bar{y}_j^{\mathrm{te}})$$

- No covariate shift assumption is needed!

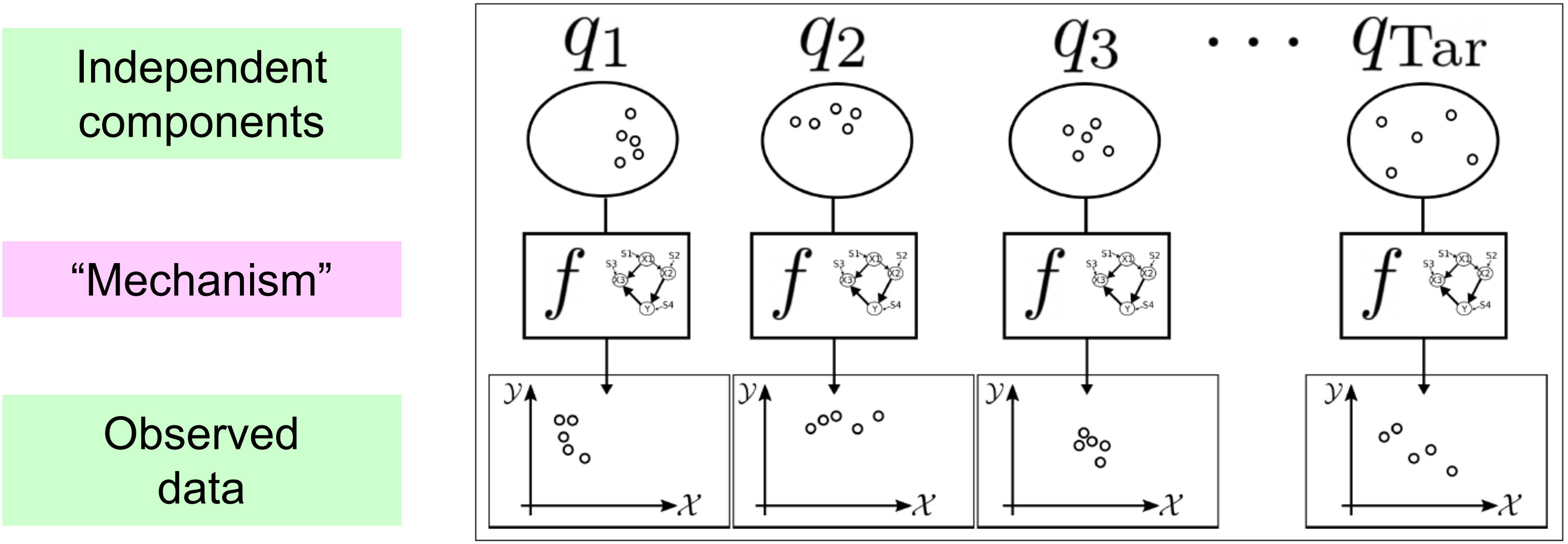
Contents

1. Introduction
2. Classical results
   A) Importance weighting
   B) Adaptive importance weighting
3. Recent results
   A) Joint upper-bound minimization
   B) Dynamic importance weighting
4. Future prospects

Conclusions

- In transfer learning with importance weighting, simultaneously performing importance estimation and predictor training is promising.
- What should we do if training and test distributions look very different?
  - Mechanism transfer!

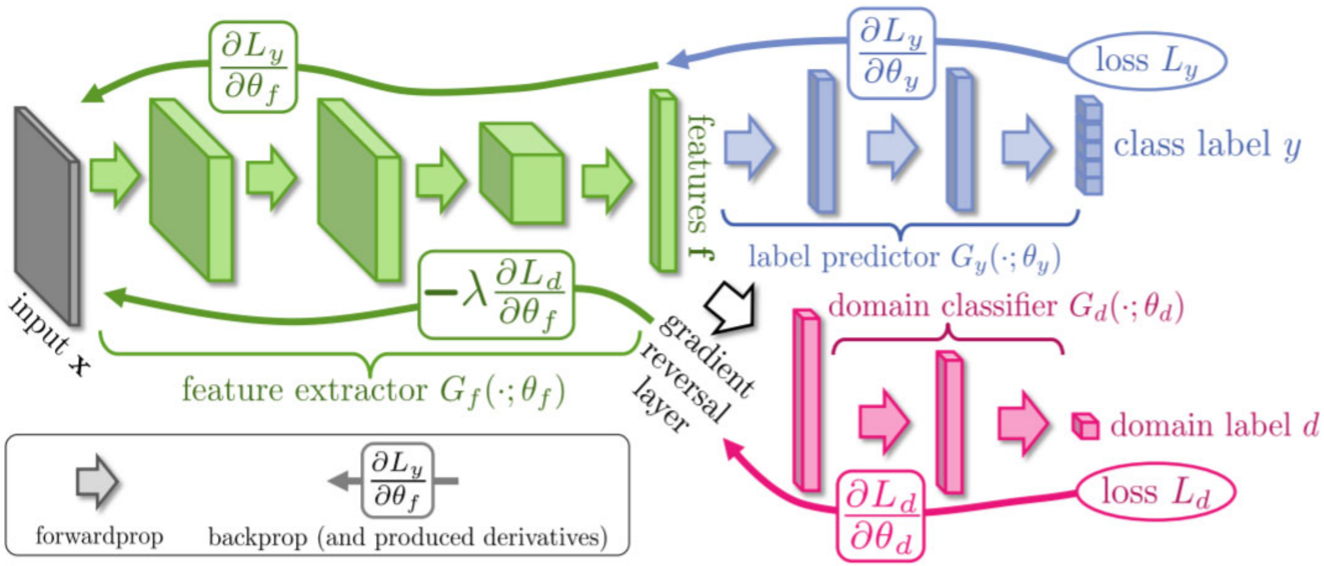
Teshima, Sato & Sugiyama (ICML2020)



Future Prospects: Domain Matching

- Domain matching would be another popular approach for transfer learning in deep learning:

Ben-David, Blitzer, Crammer & Pereira (NIPS2006)
Ganin & Lempitsky (ICML2015)



- Can we combine domain matching and importance weighting for better performance?

Future Prospects: Classification with Noisy Labels

- Output shift: $p_{\mathrm{tr}}(y|\boldsymbol{x}) \neq p_{\mathrm{te}}(y|\boldsymbol{x})$
  - Noise transition connects two distributions:

$$p_{\mathrm{tr}}(\bar{y}|\boldsymbol{x}) = \sum_{y} p(\bar{y}|y) p_{\mathrm{te}}(y|\boldsymbol{x})$$

$\boldsymbol{x}$: Input pattern

y : Class label

$\bar{y}$: Noisy class label

- Back/forward loss correction yields consistency.

Patrini, Rozza, Menon, Nock & Qu (CVPR2017)

- Estimation of noise transition only from noisy training data is the current challenge.

Xia et al. (NeurIPS2019), Yao et al. (NeurIPS2020), Xia et al. (NeurIPS2020), Zhang et al. (ICML2021), Li et al. (ICML2021), Berthon et al. (ICML2021)

- Can we use transfer learning techniques to better solve noisy label classification?