RIKEN AIP & PRAIRIE Joint Workshop
on Machine Learning and Artificial Intelligence

March 21, 2023

Transfer Learning

Masashi Sugiyama

RIKEN Center for Advanced Intelligence Project/
The University of Tokyo

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

Supervised Learning under Distribution Shift

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

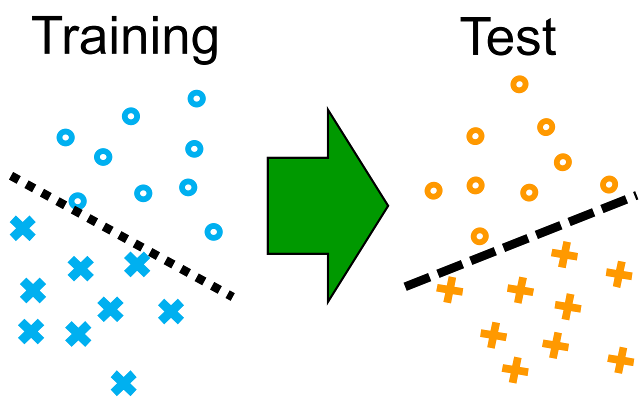
$$\min_f R(f) \qquad R(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)]$$

ℓ : loss

- **Challenge:**
  - Overcome changing distributions!

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

  - Non-stationary of the environments.
  - Sample selection bias due to privacy concerns.



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

Workshop

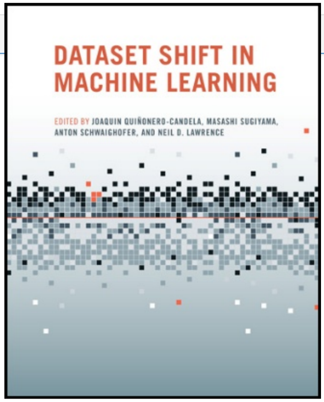
Learning when test and training inputs have different distributions

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, Schwaighofer & Lawrence (Eds.), Dataset Shift in Machine Learning, MIT Press, 2009.

NeurIPS DistShift Workshop in 2021/2022

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**, *son, MIT, and Sam Roweis, University of Toronto*
	e Prior for Adaptive Learning, *Jeff Bilmes, University of Washington*
5:40pm	**discussion**, *everyone*

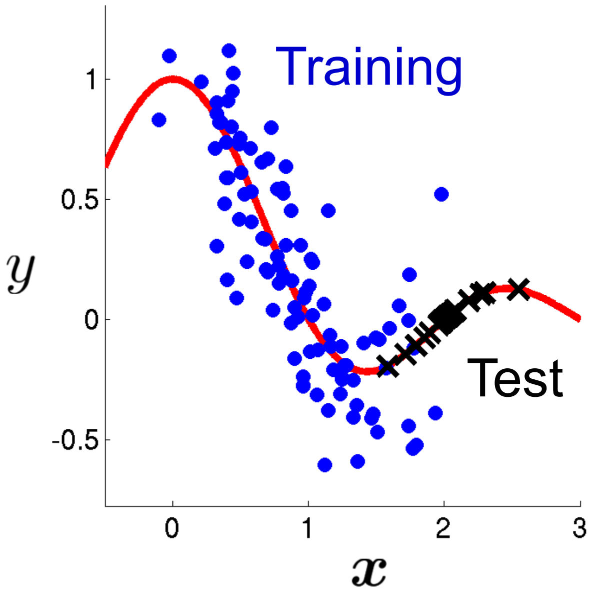
Contents

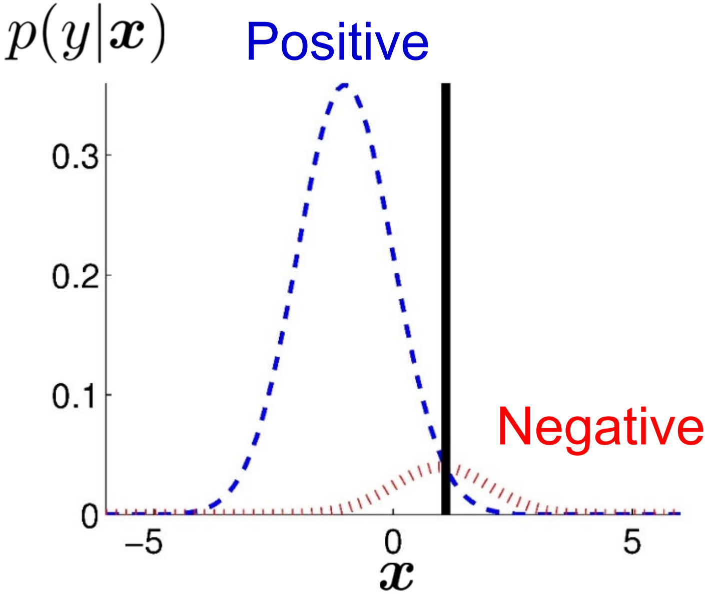
1. Basics
2. Recent approaches
3. Extension to continuous distribution shifts
4. Summary

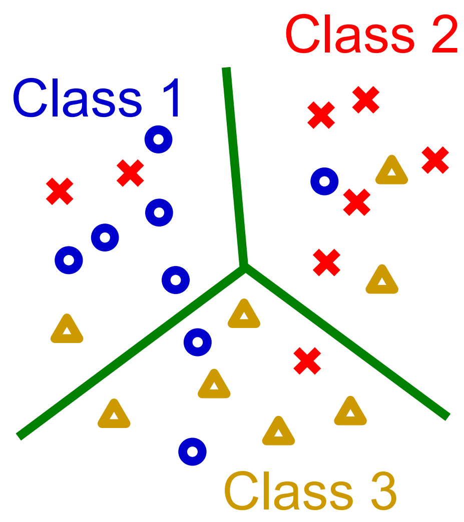
Types of Distribution Shifts

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

- Joint 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: $p_{\mathrm{tr}}(\boldsymbol{x}|y) \neq p_{\mathrm{te}}(\boldsymbol{x}|y)$







Basics: Importance-Weighted Training

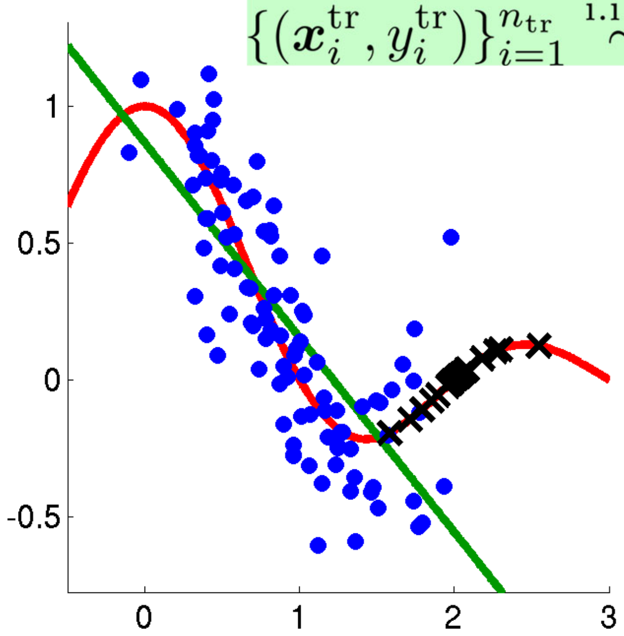
- **Covariate shift:** Only input distributions change. Shimodaira (JSPI2000)

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

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

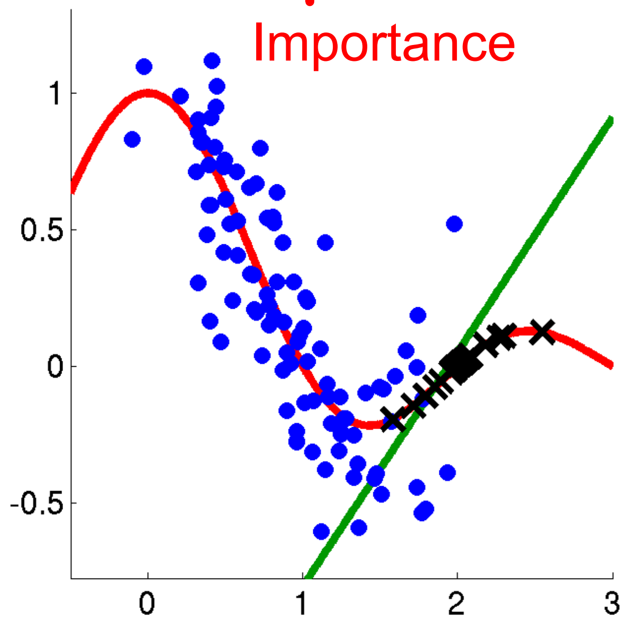
$$\underset{f}{\operatorname{argmin}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

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



Ordinary training is not consistent

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



Importance-weighted training is consistent

Direct Importance Estimation

- **Given**: training and test input data

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

- **Kernel mean matching:**

Huang, Smola, Gretton, Borgwardt & Schölkopf (NeurIPS2006)

  - Match the means of $w(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$ in characteristic reproducing kernel Hilbert space $\mathcal{H}$.

$$\min_{w\in\mathcal{H}} \left\| \int K(\boldsymbol{x},\cdot)p_{\mathrm{te}}(\boldsymbol{x})d\boldsymbol{x} - \int K(\boldsymbol{x},\cdot)w(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})d\boldsymbol{x} \right\|_{\mathcal{H}}^2$$

$K(\boldsymbol{x},\cdot)$: kernel

- **Least-squares importance fitting (LSIF)**:
  - Fit a model $w(\boldsymbol{x})$ to $\frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$ by least squares:

$$\underset{w}{\mathrm{argmin}} \left[\int \left(w(\boldsymbol{x}) - \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})} \right)^2 p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right]$$

Kanamori, Hido & Sugiyama (NeurIPS2008, JMLR2009)

$$= \underset{w}{\mathrm{argmin}} \left[\int w(\boldsymbol{x})^2 p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 2\int w(\boldsymbol{x})p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right]$$

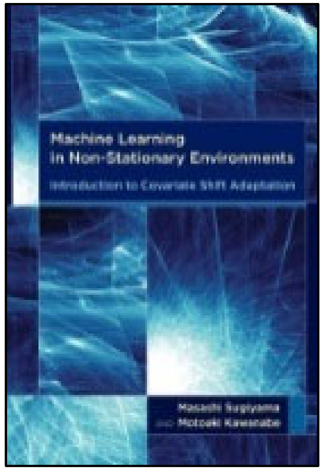
Classics: Two-Step Adaptation

1. Importance weight estimation (e.g., by LSIF):

$$\widehat{w} = \underset{w}{\operatorname{argmin}}\, \widehat{\mathbb{E}}_{p_{\mathrm{tr}}(\boldsymbol{x})}\left[\left(w(\boldsymbol{x}) - \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}\right)^2\right]$$

2. Importance-weighted predictor training:

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



Sugiyama & Kawanabe
(MIT Press 2012)

- However, estimation error in Step 1 is not taken into account in Step 2.
  - We want to integrate these two steps!

Contents

Joint Weight-Predictor Optimization

Zhang, Yamane, Lu & Sugiyama (ACML2020, SNCS2021)

- **Given**: Labeled training data and unlabeled test data

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

- **Risk upper bound**: $J_{\ell'}(f, w) \geq \frac{1}{2} R_\ell(f)^2$

$$R_\ell(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)] \qquad \ell \leq 1, \ell' \geq \ell$$

- **Joint minimization** $\displaystyle\min_{f \in \mathcal{F}, w \geq 0} J_{\ell'}(f, w)$

$$J_{\ell'}(f, w) = \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x})}\left[\left(w(\boldsymbol{x}) - \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}\right)^2\right] \quad \leftarrow \text{LSIF}$$

$$+ (\mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x},y)}[w(\boldsymbol{x})\ell'(f(\boldsymbol{x}), y)])^2 \quad \leftarrow \text{IW training}$$

  - Classic approach corresponds to 2-step minimization.

- Theoretical convergence guarantee: $\widehat{f} = \displaystyle\operatorname*{argmin}_{f \in \mathcal{F}} \min_{w \geq 0} \widehat{J}_{\ell'}(w, f)$

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

Experiments

Table 3 Mean test classification accuracy averaged over 5 trials on image datasets with neural networks. The numbers in the brackets are the standard deviations. For each dataset, the best method and comparable ones based on the *paired t-test* at the significance level 5% are described in bold face.

Dataset	Shift Level (a, b)	ERM	EIWERM	RIWERM	one-step
Fashion-MNIST	(2, 4)	81.71(0.17)	84.02(0.18)	84.12(0.06)	**85.07(0.08)**
	(2, 5)	72.52(0.54)	76.68(0.27)	77.43(0.29)	**78.83(0.20)**
	(2, 6)	60.10(0.34)	65.73(0.34)	66.73(0.55)	**69.23(0.25)**
Kuzushiji-MNIST	(2, 4)	77.09(0.18)	80.92(0.32)	81.17(0.24)	**82.45(0.12)**
	(2, 5)	65.06(0.26)	71.02(0.50)	72.16(0.19)	**74.03(0.16)**
	(2, 6)	51.24(0.30)	58.78(0.38)	60.14(0.93)	**62.70(0.55)**

$$\left(\frac{p_{\mathrm{te}}(\boldsymbol{x}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\boldsymbol{x}_i^{\mathrm{tr}})}\right)^{\gamma}$$

Shimodaira (JSPI2000)

$$\frac{p_{\mathrm{te}}(\boldsymbol{x})}{\beta p_{\mathrm{tr}}(\boldsymbol{x}) + (1-\beta) p_{\mathrm{te}}(\boldsymbol{x})}$$

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

- One-step method outperforms two-step methods!

Dynamic Importance Weighting

Fang, Lu, Niu & Sugiyama (NeurIPS2020)

- Full distribution shift: $p_{\mathrm{tr}}(\boldsymbol{x}, y) \neq p_{\mathrm{te}}(\boldsymbol{x}, y)$

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

- For each mini-batch $\{(\bar{\boldsymbol{x}}_i^{\mathrm{tr}}, \bar{y}_i^{\mathrm{tr}})\}_{i=1}^{\bar{n}_{\mathrm{tr}}}, \{(\bar{\boldsymbol{x}}_i^{\mathrm{te}}, \bar{y}_i^{\mathrm{te}})\}_{i=1}^{\bar{n}_{\mathrm{te}}}$, importance is estimated by kernel mean matching:

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

  - Simple, but highly flexible!

$$w_i \approx \frac{p_{\mathrm{te}}(\bar{\boldsymbol{x}}_i^{\mathrm{tr}}, \bar{y}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\bar{\boldsymbol{x}}_i^{\mathrm{tr}}, \bar{y}_i^{\mathrm{tr}})}$$

Experiments

Table 4: Mean accuracy (standard deviation) in percentage on Fashion-MNIST (F-MNIST for short), CIFAR-10/100 under label noise (5 trials). Best and comparable methods (paired t-test at significance level 5%) are highlighted in bold. p/s is short for pair/symmetric flip.

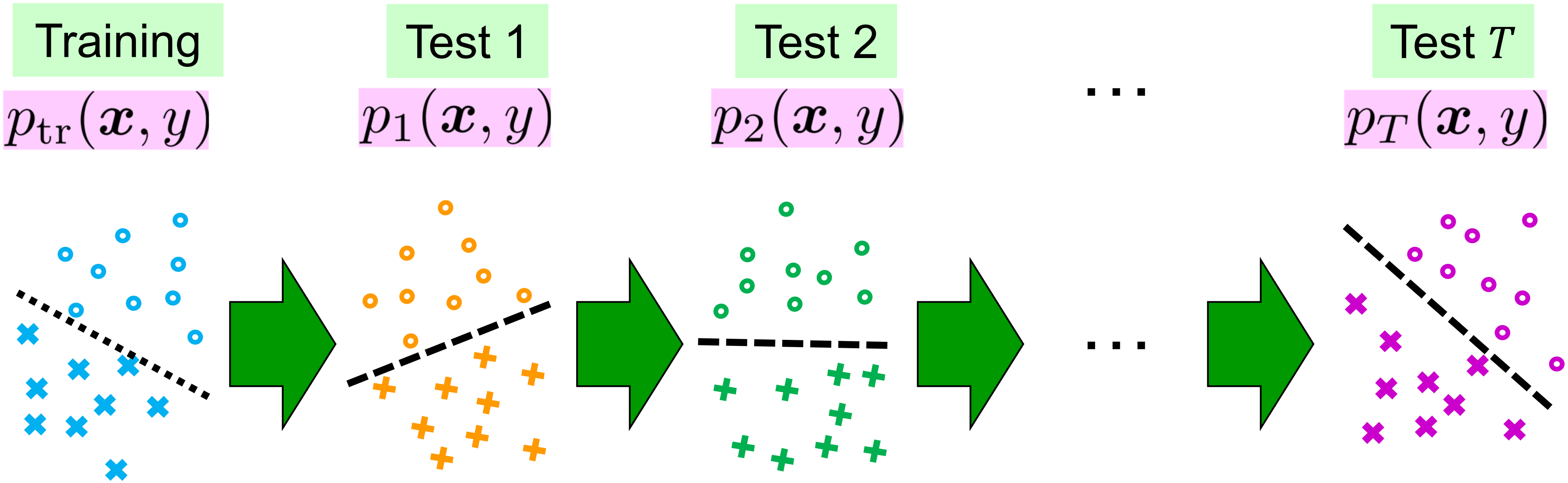
	Noise	Clean	Uniform	Random	IW	Reweight	DIW
F-MNIST	0.3 p	71.05 (1.03)	76.89 (1.06)	84.62 (0.68)	82.69 (0.38)	**88.74 (0.19)**	88.19 (0.43)
	0.4 s	73.55 (0.80)	77.13 (2.21)	84.58 (0.76)	80.54 (0.66)	85.94 (0.51)	**88.29 (0.18)**
	0.5 s	73.55 (0.80)	73.70 (1.83)	82.49 (1.29)	78.90 (0.97)	84.05 (0.51)	**87.67 (0.57)**
CIFAR-10	0.3 p	45.62 (1.66)	77.75 (3.27)	83.20 (0.62)	45.02 (2.25)	82.44 (1.00)	**84.44 (0.70)**
	0.4 s	45.61 (1.89)	69.59 (1.83)	76.90 (0.43)	44.31 (2.14)	76.69 (0.57)	**80.40 (0.69)**
	0.5 s	46.35 (1.24)	65.23 (1.11)	71.56 (1.31)	42.84 (2.35)	72.62 (0.74)	**76.26 (0.73)**
CIFAR-100	0.3 p	10.82 (0.44)	50.20 (0.53)	48.65 (1.16)	10.85 (0.59)	48.48 (1.52)	**53.94 (0.29)**
	0.4 s	10.82 (0.44)	46.34 (0.88)	42.17 (1.05)	10.61 (0.53)	42.15 (0.96)	**53.66 (0.28)**
	0.5 s	10.82 (0.44)	41.35 (0.59)	34.99 (1.19)	10.58 (0.17)	36.17 (1.74)	**49.13 (0.98)**

- Dynamic method outperforms other methods.

Contents

Continuous Distribution Shift

- So far, we focused on a fixed test domain:
  - We trained a predictor to match the test domain.
- However, test domains can change over time.



- Goal: Obtain classifier $\widehat{f}_t$ that works well for $p_t(\boldsymbol{x}, y)$.

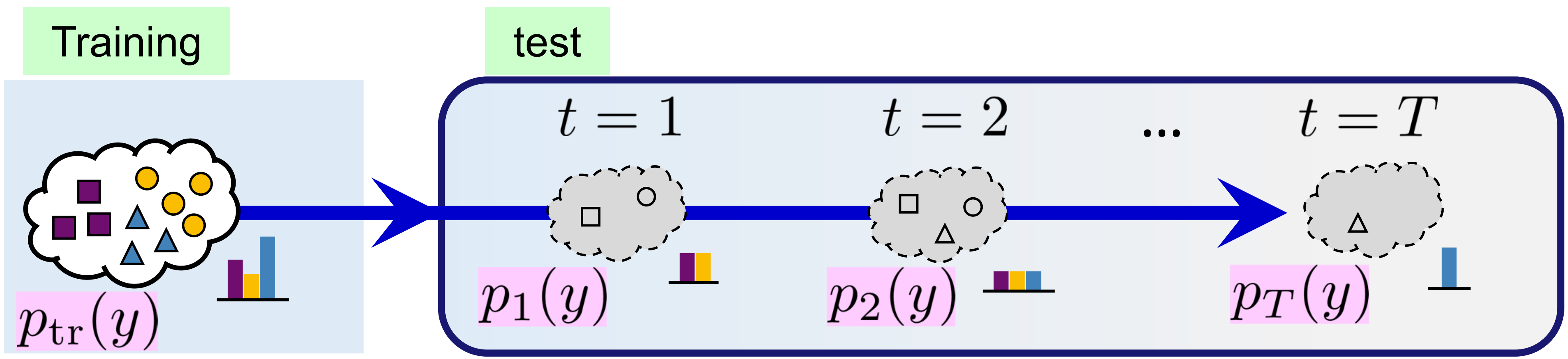
$$R_t(f) = \mathbb{E}_{p_t(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)] \quad t = 1, \ldots, T$$

Continuous Class-Prior Shift

- Class-priors $p_t(y)$ change arbitrarily over time, but class-conditionals stay unchanged:

$$p_{\mathrm{tr}}(\boldsymbol{x}|y) = p_t(\boldsymbol{x}|y) \qquad t = 1, \ldots, T$$

- Assume we are given
  - Labeled training data: $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$
  - Unlabeled test data: $\{\boldsymbol{x}_i^{(t)}\}_{i=1}^{n_t} \overset{\mathrm{i.i.d.}}{\sim} p_t(\boldsymbol{x})$



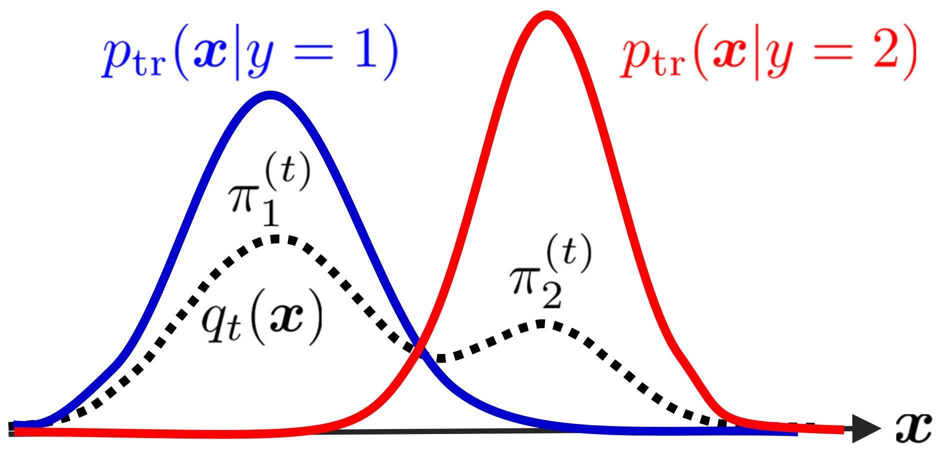
Batch Importance Weighting

du Plessis & Sugiyama (NN2014)

- At time step t, $p_t(y)$ can be estimated by implicit distribution matching (no density estimation is needed):

$$\min_{\boldsymbol{\pi}^{(t)} \in \Delta_{c-1}} \mathrm{Div}[p_{\mathrm{tr}}(\boldsymbol{x}) \| q_t(\boldsymbol{x})]$$

$$q_t(\boldsymbol{x}) = \sum_{y=1}^{c} \pi_y^{(t)} p_{\mathrm{tr}}(\boldsymbol{x}|y)$$



- Perform importance weighted training:

$$\min_f \widehat{R}_t(f) \qquad \widehat{R}_t(f) = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \frac{\widehat{p}_t(y_i^{\mathrm{tr}})}{\widehat{p}_{\mathrm{tr}}(y_i^{\mathrm{tr}})} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}})$$

ATLAS: (Adapting To LAbel Shift)

Bai, Zhang, Zhao, Sugiyama & Zhou (NeurIPS2022)

$$\min_f \widehat{R}_t(f) \qquad \widehat{R}_t(f) = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \frac{\widehat{p}_t(y_i^{\mathrm{tr}})}{\widehat{p}_{\mathrm{tr}}(y_i^{\mathrm{tr}})} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}})$$

- Batch importance weighing requires retraining in each time step.
- Can we make it computationally more efficient?
  - Online learning!

Hazan (2016)

- We use online convex optimization, assuming
  - convex loss ℓ (e.g., logistic),
  - linear model $f(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}, \quad \theta \in \Theta$.

$$\boldsymbol{\theta}_{t+1} = \Pi_\Theta \left[\boldsymbol{\theta}_t - \eta \nabla \widehat{R}_t(\boldsymbol{\theta}_t) \right]$$

Π_Θ : projection

$\eta > 0$: step size

- We use black box shift estimation for class priors.

Lipton, Wang & Smola (ICML2018)

Choice of Step Size η

$$\boldsymbol{\theta}_{t+1} = \Pi_{\Theta}\left[\boldsymbol{\theta}_t - \eta \nabla \widehat{R}_t(\boldsymbol{\theta}_t)\right] \qquad \widehat{R}_t(\boldsymbol{\theta}) = \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \frac{\widehat{p}_t(y_i^{\mathrm{tr}})}{\widehat{p}_{\mathrm{tr}}(y_i^{\mathrm{tr}})} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})$$

- If distribution shift is
  - slow, η should be small to keep the previous classifier.
  - fast, η should be large to quickly update the classifier.
- How do we choose η in practice?
  - Ensemble learning! Zhao, Zhang, Zhang & Zhou (NeurIPS2020)
- For $0 < \eta_1 < \cdots < \eta_M$, we run M learners:

$$\boldsymbol{\theta}_{t+1}^{(m)} = \Pi_{\Theta}\left[\boldsymbol{\theta}_t^{(m)} - \eta_m \nabla \widehat{R}_t(\boldsymbol{\theta}_t^{(m)})\right]$$

- Final output is the weighted average (cf. Hedge):

Freund & Schapire (JCSS1997)

$$\boldsymbol{\theta}_t = \sum_{m=1}^{M} p_t^{(m)} \boldsymbol{\theta}_t^{(m)} \qquad p_t^{(m)} \propto \exp\left(-\varepsilon \sum_{s=1}^{t-1} \widehat{R}_s(\boldsymbol{\theta}_s^{(m)})\right) \qquad \varepsilon = \Theta\left(\sqrt{\frac{\ln M}{T}}\right)$$

Theoretical Guarantee

- **Shift intensity**: $V_T = \sum_{t=2}^{T} \|p_t(y) - p_{t-1}(y)\|_1$

Suppose $V_T = \Theta(T^{-\frac{1}{2}})$ for simplicity

- When V_T is known:
  - Simple online learning with step size $\eta = \Theta(V_T^{\frac{1}{3}} T^{-\frac{1}{3}})$ achieves the optimal dynamic regret:

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t(\boldsymbol{\theta}_t) - \sum_{t=1}^{T} \min_{\boldsymbol{\theta}\in\Theta} R_t(\boldsymbol{\theta})\right] = \mathcal{O}\left(V_T^{\frac{1}{3}} T^{\frac{2}{3}}\right)$$

- Even when V_T is unknown:
  - ATLAS still achieves the optimal dynamic regret!
    - Number of learners: $M = 1 + \lceil \frac{1}{2} \log_2(1 + 2T) \rceil$
    - Step size: $\eta_m = 2^{m-1} G/\sqrt{T}, \quad m = 1, \ldots, M$

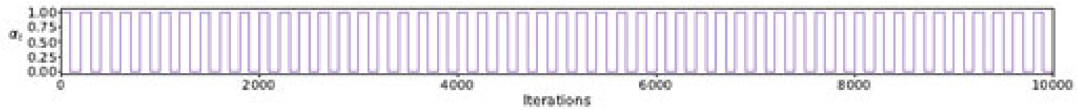
Experiments

Table 3: Average error (%) of different algorithms on various real-world datasets. We report the mean and standard deviation over five runs. The best algorithms are emphasized in bold. "•" indicates the algorithms that are significantly inferior to ATLAS-ADA by the paired t-test at a 5% significance level. Here AT-ADA represents ATLAS-ADA (with *OKM*). The online sample size is set as $N_t = 10$.

	Lin							Squ						
	FIX	**FTH**	**FTFWH**	**ROGD**	**UOGD**	**ATLAS**	**AT-ADA**	**FIX**	**FTH**	**FTFWH**	**ROGD**	**UOGD**	**ATLAS**	**AT-ADA**
ArXiv	• 30.28 ±0.07	• 28.18 ±0.28	• 25.74 ±0.21	• 23.09 ±0.20	**21.04** ±0.11	• 22.10 ±0.09	21.28 ±0.09	• 30.35 ±0.06	• 26.72 ±0.39	• 28.05 ±0.20	• 24.44 ±0.17	• 21.96 ±0.07	• 21.36 ±0.06	**20.80** ±0.06
EuroSAT	• 14.06 ±0.09	• 11.16 ±0.11	• 9.78 ±0.12	• 12.56 ±3.16	**7.04** ±0.11	• 7.19 ±0.10	7.13 ±0.11	• 14.15 ±0.11	• 10.22 ±0.08	• 10.26 ±0.06	• 8.91 ±0.05	• 7.30 ±0.07	• 6.97 ±0.08	**6.81** ±0.06
MNIST	• 1.79 ±0.02	• 1.38 ±0.03	• 1.20 ±0.02	• 1.25 ±0.02	**1.06** ±0.02	**1.06** ±0.02	**1.06** ±0.02	• 1.79 ±0.04	• 1.26 ±0.03	• 1.28 ±0.04	• 1.32 ±0.04	• 1.13 ±0.03	• 1.04 ±0.02	**1.01** ±0.04
Fashion	• 11.86 ±0.04	• 8.47 ±0.07	**7.84** ±0.06	8.18 ±0.07	7.95 ±0.08	• 8.36 ±0.07	8.04 ±0.08	• 11.92 ±0.09	• 8.24 ±0.09	• 8.35 ±0.07	• 8.63 ±0.07	• 8.42 ±0.04	• 8.05 ±0.07	**7.73** ±0.05
CIFAR10	• 20.77 ±0.12	• 17.36 ±0.14	15.77 ±0.12	• 18.45 ±0.47	**15.54** ±0.15	• 15.77 ±0.11	15.62 ±0.14	• 20.77 ±0.08	• 16.67 ±0.12	• 16.72 ±0.12	• 17.40 ±0.11	• 16.29 ±0.09	• 15.18 ±0.07	**14.84** ±0.05
CINIC10	• 33.98 ±0.22	• 28.85 ±0.10	• 26.87 ±0.13	• 32.54 ±2.59	**26.21** ±0.15	• 26.66 ±0.19	26.38 ±0.16	• 33.99 ±0.16	• 27.99 ±0.09	• 28.08 ±0.08	• 28.58 ±0.09	• 27.00 ±0.14	• 25.94 ±0.13	**25.56** ±0.12

- Lin: Nearly stationary

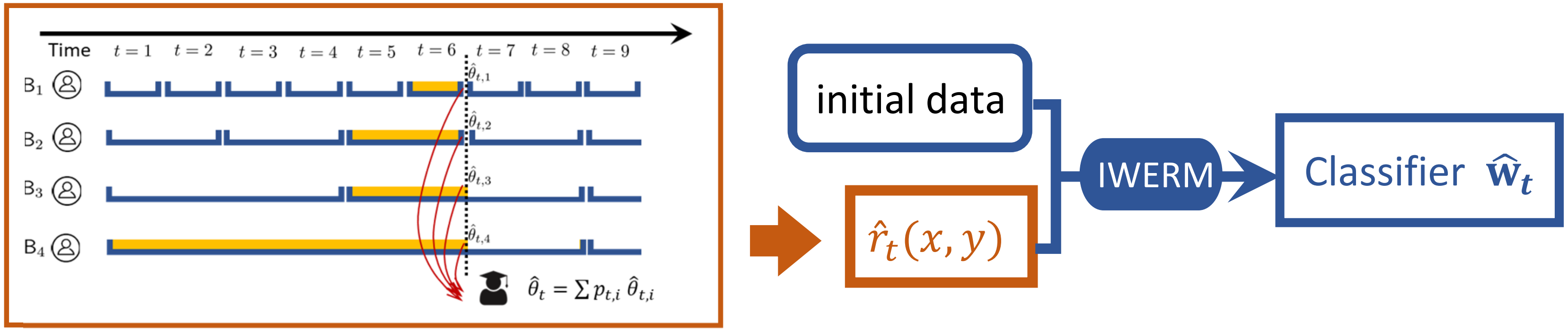
  - Comparable to methods designed for stationary environments.
- Squ: Highly non-stationary
  - Overperforms all baselines.



Online Covariate Shift Adaptation

Zhang, Zhang, Zhao & Sugiyama (arXiv2023)

A new method for continuous covariate shift via online density ratio estimation



online estimation of time-varying density ratio $r_t(\boldsymbol{x}) = p_t(\boldsymbol{x})/p_{\mathrm{tr}}(\boldsymbol{x})$

Importance-weighted (IW) ERM $\hat{\boldsymbol{w}}_t = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \sum_{i=1}^{n_{\mathrm{tr}}} \hat{r}_t(\boldsymbol{x}_i)\ell(\boldsymbol{w}^\top \boldsymbol{x}_i, y_i)$

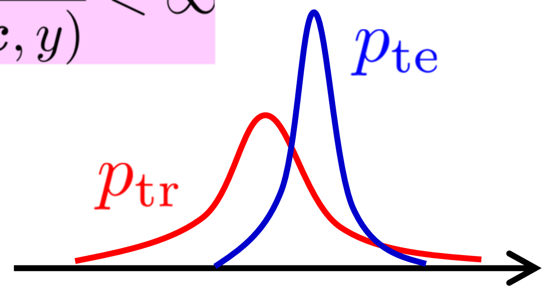
- To be presented soon!

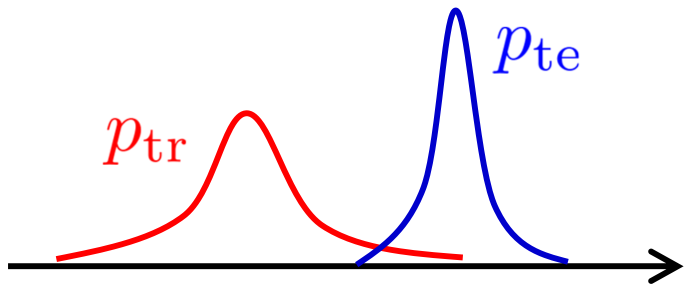
Contents

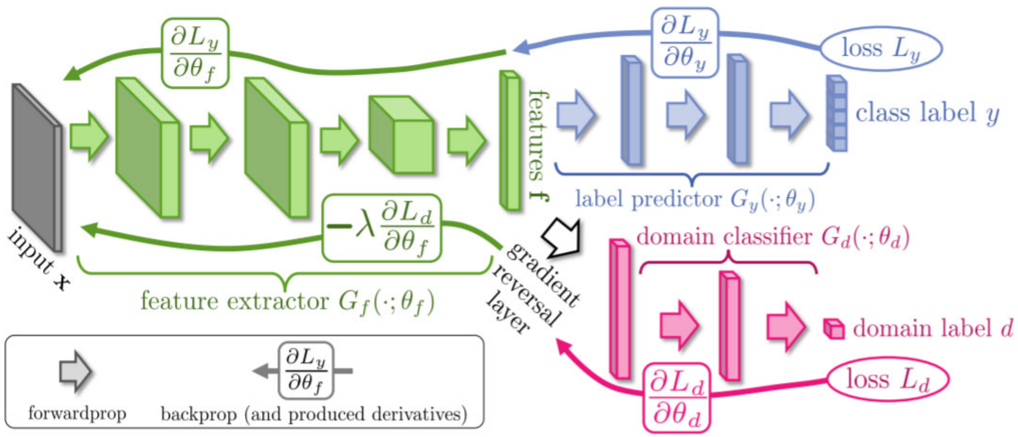
Summary

- Importance weighting: versatile for transfer learning
  - However, the training domain must cover the test domain.
- What if the test domain sticks out from the training domain?
  - Input domain matching
  - Mechanism transfer
- Further development needed!

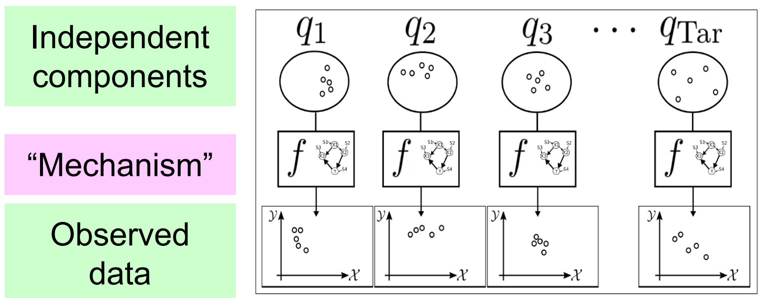
$$\frac{p_{\mathrm{te}}(\boldsymbol{x}, y)}{p_{\mathrm{tr}}(\boldsymbol{x}, y)} < \infty$$







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



Teshima, Sato & Sugiyama (ICML20

Online Adaptation in Practice

- In real-world application,
  - Updating the system immediately after receiving new data is dangerous since new data can be malicious.
  - The system may be updated periodically (daily, weekly, monthly, etc.).
  - The latest data may be incorporated in a temporary memory (e.g., 4000 tokens in ChatGPT).