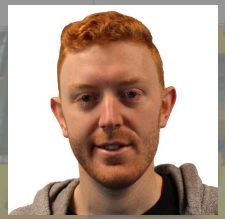
Transfer and Marginalize: Explaining Away Label Noise with Privileged Information (TRAM)

Mark Collier, Rodolphe Jenatton, Efi Kokiopoulou, Jesse Berent




Problem setup

Google AI
Brain Team

"Privileged information" / "auxiliary information"

- **At training time**: triplets $(\mathbf{x}_i, \mathbf{a}_i, y_i)$
- **At test time**: pairs $(\mathbf{x}_i, \cancel{\mathbf{a}_i}, y_i)$

Example:
- Annotator features
- Typically depend on **x**
  - E.g., time spent to annotate

- **Goal**: Help explain away otherwise irreductible aleatoric label noise

Natural solution

Google AI
Brain Team

Marginalized predictions

- **At training time**: Learn $p(y|\mathbf{x}, \mathbf{a})$

- **At test time**: Compute $\int p(y|\mathbf{x}, \mathbf{a})p(\mathbf{a}|\mathbf{x})d\mathbf{a}$

...but

Google AI
Brain Team

Several challenges

- Typically intractable to compute $\int p(y|\mathbf{x}, \mathbf{a})p(\mathbf{a}|\mathbf{x})d\mathbf{a}$
- Learning the density is a difficult problem $p(\mathbf{a}|\mathbf{x})$
  - PI often has mixed type features
- Even simplified (=ind. assumption) Monte Carlo estimate has a *O(S)* overhead

$$\int p(y|\mathbf{x}, \mathbf{a})p(\mathbf{a})d\mathbf{a} \approx \frac{1}{S}\sum_{s=1}^{S} p(y|\mathbf{x}, \mathbf{a}_s) \text{ with } \mathbf{a}_s \sim p(\mathbf{a})$$

Our approach

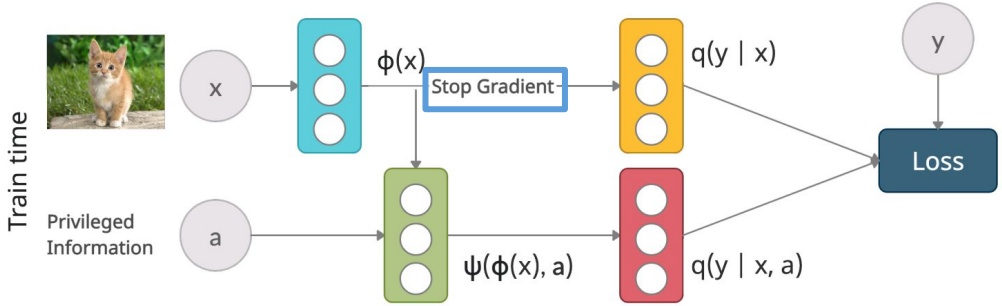
Google AI
Brain Team

Simple transfer-learning approach of representations learned with PI

Compared with previous work:

- Does not need $p(\mathbf{a}|\mathbf{x})$
- Single training (with negligible overhead)
- No overhead at evaluation time
- Guaranteed to approximate $p(y|\mathbf{x})$

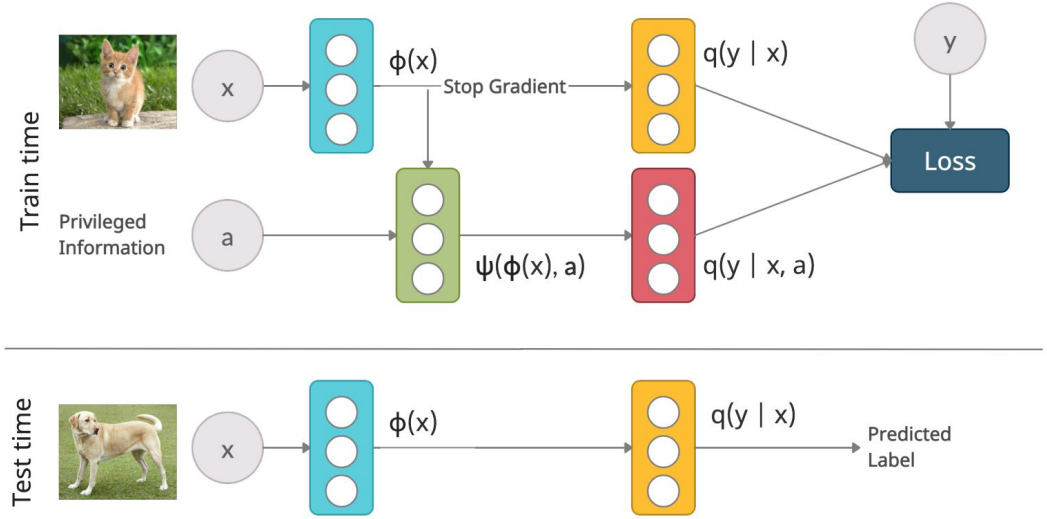
TRAM



$$\pi(y|\mathbf{x};\mathbf{w}) = q\big(y|\texttt{stop\_gradient}(\phi(\mathbf{x}));\mathbf{w}\big)$$

$$\min_{\mathbf{u},\mathbf{w},\phi,\psi} \mathbb{E}_{(\mathbf{x},\mathbf{a},y)\sim p(\mathbf{x},\mathbf{a},y)}\Big[\mathrm{CE}(y,\pi(y|\mathbf{x}))+\beta\mathcal{L}(y,q(y|\mathbf{x},\mathbf{a}))\Big]$$

TRAM



$$\pi(y|\mathbf{x};\mathbf{w}) = q\big(y|\texttt{stop\_gradient}(\phi(\mathbf{x}));\mathbf{w}\big)$$

$$\min_{\mathbf{u},\mathbf{w},\phi,\psi} \mathbb{E}_{(\mathbf{x},\mathbf{a},y)\sim p(\mathbf{x},\mathbf{a},y)}\Big[\texttt{CE}(y,\pi(y|\mathbf{x}))+\beta\mathcal{L}(y,q(y|\mathbf{x},\mathbf{a}))\Big]$$

Results

Google AI
Brain Team

Table 2: CIFAR-10 neg. log-likelihood & accuracy (trained on CIFAR-10H). Averaged over 20 runs $\pm$ 1 std. deviation.

METHOD	↓NLL	↑ACCURACY
NO PI	1.058 ± 0.050	67.0 ± 1.7
FULL MARGINALIZATION	1.119 ± 0.058	70.3 ± 2.5

Results

Google AI
Brain Team

Table 2: CIFAR-10 neg. log-likelihood & accuracy (trained on CIFAR-10H). Averaged over 20 runs ± 1 std. deviation.

METHOD	↓NLL	↑ACCURACY
NO PI	1.058 ± 0.050	67.0 ± 1.7
LAMBERT ET AL. (2018)	1.033 ± 0.044	67.1 ± 1.3
FULL MARGINALIZATION	1.119 ± 0.058	70.3 ± 2.5
DISTILLATION NO PI	1.118 ± 0.037	70.1 ± 1.4
LOPEZ-PAZ ET AL. (2015)	1.121 ± 0.040	70.2 ± 1.4

Based on dropout

Based on distillation

Results

Google AI
Brain Team

Table 2: CIFAR-10 neg. log-likelihood & accuracy (trained on CIFAR-10H). Averaged over 20 runs ± 1 std. deviation.

METHOD	↓NLL	↑ACCURACY
NO PI	1.058 ± 0.050	67.0 ± 1.7
ZERO IMPUTATION	1.009 ± 0.032	68.7 ± 1.4
MEAN IMPUTATION	$\mathbf{0.963} \pm 0.058$	70.1 ± 1.5
LAMBERT ET AL. (2018)	1.033 ± 0.044	67.1 ± 1.3
FULL MARGINALIZATION	1.119 ± 0.058	70.3 ± 2.5
DISTILLATION NO PI	1.118 ± 0.037	70.1 ± 1.4
LOPEZ-PAZ ET AL. (2015)	1.121 ± 0.040	70.2 ± 1.4

Not compared with in previous work...

Results

Google AI
Brain Team

Table 2: CIFAR-10 neg. log-likelihood & accuracy (trained on CIFAR-10H). Averaged over 20 runs ± 1 std. deviation.

METHOD	↓NLL	↑ACCURACY
NO PI	1.058 ± 0.050	67.0 ± 1.7
ZERO IMPUTATION	1.009 ± 0.032	68.7 ± 1.4
MEAN IMPUTATION	**0.963** ± 0.058	70.1 ± 1.5
LAMBERT ET AL. (2018)	1.033 ± 0.044	67.1 ± 1.3
FULL MARGINALIZATION	1.119 ± 0.058	70.3 ± 2.5
TRAM	0.980 ± 0.037	70.1 ± 1.4
HET-TRAM	0.972 ± 0.038	**70.4** ± 1.5
DISTILLATION NO PI	1.118 ± 0.037	70.1 ± 1.4
LOPEZ-PAZ ET AL. (2015)	1.121 ± 0.040	70.2 ± 1.4
DISTILLED-TRAM	**0.941** ± 0.039	**71.8** ± 1.4

Slightly different parametrization of the head $q(y|\mathbf{x})$

Results

Google AI
Brain Team

Table 2: CIFAR-10 neg. log-likelihood & accuracy (trained on CIFAR-10H). Averaged over 20 runs ± 1 std. deviation.

METHOD	↓NLL	↑ACCURACY
NO PI	1.058 ± 0.050	67.0 ± 1.7
ZERO IMPUTATION	1.009 ± 0.032	68.7 ± 1.4
MEAN IMPUTATION	**0.963** ± 0.058	70.1 ± 1.5
LAMBERT ET AL. (2018)	1.033 ± 0.044	67.1 ± 1.3
FULL MARGINALIZATION	1.119 ± 0.058	70.3 ± 2.5
TRAM	0.980 ± 0.037	70.1 ± 1.4
HET-TRAM	0.972 ± 0.038	**70.4** ± 1.5
DISTILLATION NO PI	1.118 ± 0.037	70.1 ± 1.4
LOPEZ-PAZ ET AL. (2015)	1.121 ± 0.040	70.2 ± 1.4
DISTILLED-TRAM	**0.941** ± 0.039	**71.8** ± 1.4

Table 3: ImageNet validation neg-log-likelihood and accuracy. Avg. over 10 seeds ± 1 std. deviation.

METHOD	↓NLL	↑ACCURACY
NO PI	1.264 ± 0.007	71.7 ± 0.2
ZERO IMPUTATION	1.895 ± 0.008	63.5 ± 0.2
MEAN IMPUTATION	1.619 ± 0.007	65.1 ± 0.3
LAMBERT ET AL. (2018)	1.264 ± 0.006	71.8 ± 0.1
FULL MARGINALIZATION	1.217 ± 0.004	72.6 ± 0.2
TRAM	1.225 ± 0.006	72.5 ± 0.2
HET-TRAM	**1.207** ± 0.008	**72.8** ± 0.2
DISTILLATION NO PI	1.207 ± 0.004	72.6 ± 0.2
LOPEZ-PAZ ET AL. (2015)	1.216 ± 0.003	72.7 ± 0.2
DISTILLED-TRAM	**1.154** ± 0.004	**73.8** ± 0.2

Conclusions

Google AI
Brain Team

- Simple approach to deal with PI based on transfer learning
- More material not covered here
  - Theoretical analysis in the case of linear models
  - NLP application (“Civil Comments” dataset)

METHOD	$p(\boldsymbol{a}\|\boldsymbol{x})$ REQUIRED	TRAINING	TEST COST	WEIGHT SHARING	APPROXIMATE $p(y\|\boldsymbol{x})$
IMPUTATION	×	1 MODEL, 1 STEP	= NO PI	✓	×
DISTILLATION (LOPEZ-PAZ ET AL., 2015)	×	2 MODELS, 2 STEPS	= NO PI	×	×
HET. DROPOUT (LAMBERT ET AL., 2018)	×	1 MODEL, 1 STEP	= NO PI	✓	✓
MIML-FCN+ (YANG ET AL., 2017)	×	1 MODEL, 1 STEP	= NO PI	×	×
FULL MARGINALIZATION	✓	1 MODEL, 1 STEP	$\mathcal{O}(S \times$ NO PI$)$	✓	✓
TRAM (OURS)	×	1 MODEL, 1 STEP	= NO PI	✓	✓
HET-TRAM (OURS)	×	1 MODEL, 1 STEP	= NO PI	✓	✓
DISTILLED-TRAM (OURS)	×	2 MODELS, 2 STEPS	= NO PI	✓	✓