Learning under Noisy Supervision, Part 5: Beyond Class-Conditional Noise

Gang Niu

Research Scientist
Imperfect Information Learning Team
RIKEN Center for Advanced Intelligence Project

ACML 2021 Tutorial
November 17, 2021

Outline

Class-conditional noise (CCN) model (Patrini+, CVPR 2017)

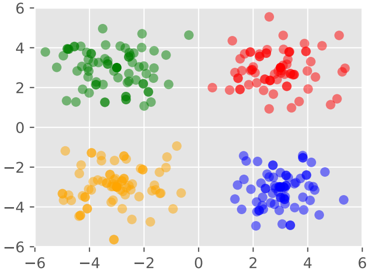
- All models are wrong, but some are useful (Box, "Science and Statistics", JASA 1976)
  - Following the law of total probability, $p(\tilde{y} \mid x) = \sum_y p(\tilde{y} \mid x, y) p(y \mid x)$
  - Assume $p(\tilde{y} \mid x, y) = p(\tilde{y} \mid y)$
    i.e., the corruption $y \to \tilde{y}$ is instance-independent and class-conditional
  - Equivalently, using transition matrix T where $[T]_{i,j} = p(\tilde{y} = j \mid y = i)$

$$\begin{pmatrix} p(\tilde{y}{=}1|x) \\ \vdots \\ p(\tilde{y}{=}c|x) \end{pmatrix} = \begin{pmatrix} p(\tilde{y}{=}1|y{=}1) & \ldots & p(\tilde{y}{=}c|y{=}1) \\ \vdots & \ddots & \vdots \\ p(\tilde{y}{=}1|y{=}c) & \ldots & p(\tilde{y}{=}c|y{=}c) \end{pmatrix}^\top \begin{pmatrix} p(y{=}1|x) \\ \vdots \\ p(y{=}c|x) \end{pmatrix}$$

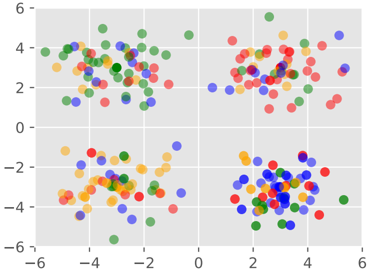
$$\Downarrow$$

$$\boldsymbol{p}_{\tilde{y}|x} = T^\top \boldsymbol{p}_{y|x}$$

What does CCN look like in 2D?



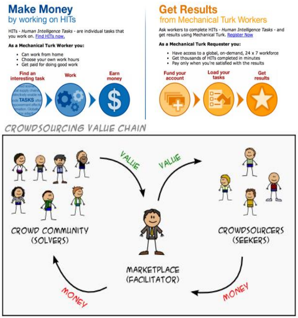
Clean training data
Easy to optimize
Easy to generalize



Noisy training data
Easy to optimize
Hard to generalize

Label noise is (almost) everywhere in industry

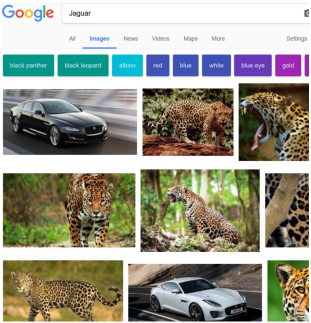
Active label collection



In crowdsourcing,
labels are from non-experts

(Credit to Amazon Mechanical Turk and IBM Crowdsourcing and Crowdfunding)

Passive label collection



In search engine,
labels are from users' clicks

(Credit to Google Images)

Going beyond CCN

However, CCN is not enough in expressing/modeling real-world label noise!

We need to go beyond it.

Outline

1. Motivation
2. Instance-dependent noise (IDN)
3. Mutually contaminated distributions (MCD)
4. Conclusions

Mainstream approaches to DL under CCN

- Loss correction
  - Design a corrected loss function such that

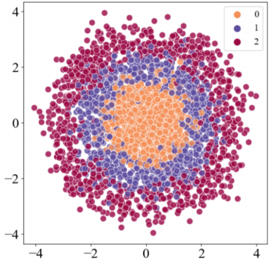
    minimize corrected loss on noisy data = minimize original loss on clean data
- Sample selection/reweighting
  - Selection: select data likely with correct labels and train only on those data
  - Reweighting: upweight/downweight data likely with correct/incorrect labels
- Label correction
  - Direct: correct the given labels using predicted labels
  - Indirect: sample selection + semi-supervised learning

Loss correction may fail under IDN

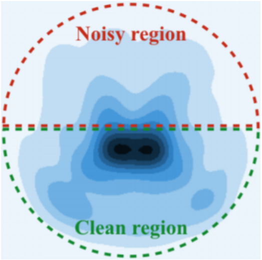
- CCN assumes $p(\tilde{y} \mid x, y) = p(\tilde{y} \mid y)$
  - It holds that $\boldsymbol{p}_{\tilde{y}|x} = T^\top \boldsymbol{p}_{y|x}$ where T is a matrix independent of x
  - Possible to estimate T from data since all instances share the same T
- IDN does not assume $p(\tilde{y} \mid x, y) = p(\tilde{y} \mid y)$
  - It becomes $\boldsymbol{p}_{\tilde{y}|x} = T(x)^\top \boldsymbol{p}_{y|x}$ where T is a matrix-valued function
  - Impossible to estimate T from data since each instance x has its $T(x)$
    i.e., IDN is mathematically unidentifiable, regardless of the size of data
  - Hence, without additional assumption/information, loss correction fails

How about sample selection?

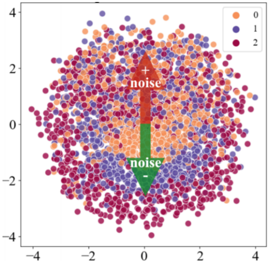
- Sample selection may also fail (Berthon+, ICML 2021; Zhu+, CVPR 2021)
  - The memorization effect is weakened—learn mislabeled data in low-noise regions first
  - Even if sample selection is perfect, a covariate shift exists between clean distributions



Clean training data



Density of selected small-loss data



Noisy training data

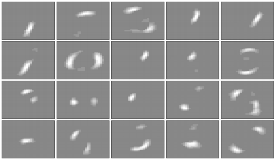
Wait a minute, can we approximate IDN?

- With additional assumption/information, we can obtain some approximations of IDN (list is not comprehensive)
  - Boundary consistent noise (for binary classification) (Menon+, MLJ 2018)
  - Bounded IDN (for binary classification) (Cheng+, ICML 2020)
  - Part-dependent noise (Xia+, NeurIPS 2020)
  - Difficulty-dependent noise (Wang+, AAAI 2021; Zhu+, CVPR 2021; Zhang+, arXiv 2021)
  - Confidence-scored IDN (Berthon+, ICML 2021)
- After we approximate IDN, we will perform loss correction

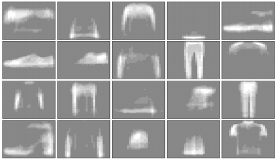
Part-dependent noise (PDN) (Xia+, NeurIPS 2020)

- PDN is naturally motivated
  - Humans perceive instances based on the parts, physiologically and psychologically
  - More likely to annotate an instance based on its parts but not the whole instance
  - A wrong mapping from parts to classes would cause PDN (a special case of IDN)
- 3 key assumptions of PDN
  - Every instance can be decomposed into r parts (a convex combination of r parts)
  - For every class, there are at least r anchor points
  - For every x, $T(x)$ is a convex combination of r matrices (with the same weights)

Parts for MNIST

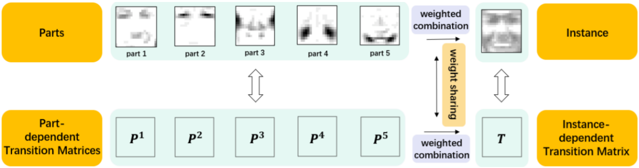


Parts for Fashion-MNIST

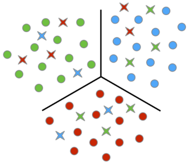


Effective learning under PDN (Xia+, NeurIPS 2020)

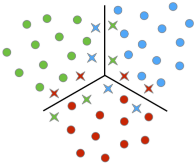
1. Learn the parts and the combination weights
   1.1. Estimate $p(\tilde{y} \mid x)$ from noisy data, and extract latent representations of instances
   1.2. Learn the parts and the combination weights by non-negative matrix factorization
2. Estimate the rows of $T(x)$ for anchor points
   2.1. When x is an anchor point for class i, we obtain that $\forall j$, $[T(x)]_{i,j} = p(\tilde{y} = j \mid x)$
3. Recover $T(x)$ for all training data (including non-anchor points)
   3.1. Estimate $P^1, \ldots, P^r$ given the weights and those rows of $T(x)$ for anchor points
   3.2. Recover $T(x)$ for every training instance x based on the weights and $P^1, \ldots, P^r$



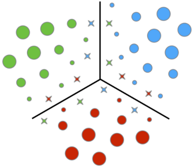
Confidence-scored IDN (CSIDN) (Berthon+, ICML 2021)



CCN



IDN (here, multi-class boundary consistent noise)



CSIDN

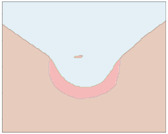
- CSIDN $\geq$ boundary consistent noise + difficulty-dependent noise
  - Binary boundary consistent noise: noise gets higher if $p(y = 1 \mid x)$ is closer to 0.5
  - Difficulty-dependent noise: x influences the noise magnitude but not its dynamics

    $$p(\tilde{y} \mid \tilde{y} \neq y, x, y) = p(\tilde{y} \mid \tilde{y} \neq y, y) \iff [T(x)]_{i,j|j\neq i} = (1 - [T(x)]_{i,i})[E]_{i,j}$$

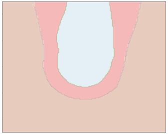
    where $1 - [T(x)]_{i,i} = p(\tilde{y} \neq y \mid y = i, x)$ controls the magnitude of the noise and $[E]_{i,j} = p(\tilde{y} = j \mid \tilde{y} \neq y, y = i)$ is CCN and controls the dynamics of the noise
  - CSIDN assumes that the confidence information $r_{x_i} = p(y = \tilde{y}_i \mid x_i, \tilde{y}_i)$ is available which can indicate both of the boundary information and the difficulty information

Instance-level forward correction (ILFC) (Berthon+, ICML 2021)

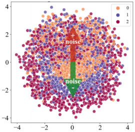
- ILFC minimizes $\ell(T(x_i)^\top g(x_i), \tilde{y}_i)$ for each $(x_i, \tilde{y}_i, r_i)$
  - Without loss of generality, assume that ℓ is the cross-entropy loss
  - Hence, we need the $\tilde{y}_i$-th column of $T(x_i)$ for computing the loss
- How to effectively estimate $[T(x_i)]_{:,\tilde{y}_i}$?
  1. The matrix E is CCN and thus can be estimated from anchor points and $\hat{p}(\tilde{y} \mid x)$
  2. $[T(x_i)]_{\tilde{y}_i,\tilde{y}_i}$ can be estimated as $r_i \hat{p}(\tilde{y} = \tilde{y}_i \mid x_i)/\hat{p}(y = \tilde{y}_i \mid x_i)$ in an iterative way
  3. Note that for $j \neq \tilde{y}_i$, $r_i = p(y = \tilde{y}_i \mid x_i, \tilde{y}_i)$ is uninformative to estimate $[T(x_i)]_{j,j}$
     We heuristically set $[\widehat{T}(x_i)]_{j,j}$ as the empirical average of $[\widehat{T}(x_k)]_{j,j}$ where $\tilde{y}_k = j$
  4. Finally, $[T(x_i)]_{j,\tilde{y}_i | j \neq \tilde{y}_i}$ can be estimated as $(1 - [\widehat{T}(x_i)]_{j,j})[\widehat{E}]_{j,\tilde{y}_i}$



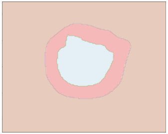
GCE, 0.5 IDN



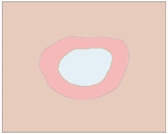
GCE, 0.4 IDN



Noisy data



ILFC, 0.4 IDN



ILFC, 0.5 IDN

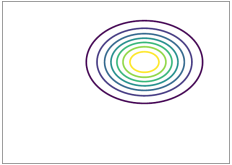
A summary of IDN settings and methods

- IDN strictly generalizes CCN
  - Transition matrix $T \Longrightarrow$ Matrix-valued function $T(x)$
- IDN is notably more challenging than CCN
  - The memorization effect acts differently in regions with different $T(x)$
  - $T(x)$ is not identifiable unless we (roughly or nicely) approximate IDN
  - Rely on part-dependent noise if we can decompose our data into parts
  - Rely on confidence-scored IDN if we collected or can assign the scores
  - Otherwise, try boundary consistent noise or difficulty-dependent noise

Outline

When $p(x \mid y)$ instead of $p(y \mid x)$ is corrupted (Scott+, COLT 2013)

An illustrative example of MCD (Lu+, ICLR 2019)

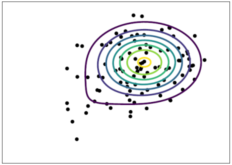


Clean P component

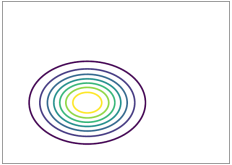
$\Longrightarrow\Longrightarrow\Longrightarrow$

corrupt into

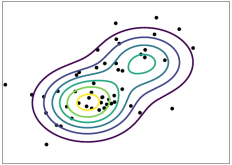
$\Longrightarrow\Longrightarrow\Longrightarrow$



Noisy P mixture (0.9P+0.1N)



Clean N component



Noisy N mixture (0.4P+0.6N)

Is it still a problem of noisy supervision? Yes!

Does it belong to CCN or IDN? No...

MCD also (strictly) generalizes CCN (Menon+, ICML 2015)

- In common: $\{(x_1, \tilde{y}_1), \ldots, (x_n, \tilde{y}_n)\}$ drawn from $p(x, \tilde{y})$
- CCN corrupts class-posterior probability: $\boldsymbol{p}_{\tilde{y}|x} = T^\top \boldsymbol{p}_{y|x}$
  - T is a label transition matrix such that $[T]_{i,j} = p(\tilde{y} = j \mid y = i)$
  - It is a label-noise model for the corruption of the labeling process
  - $p(x)$ remains the same so that the memorization effect is reliable
  - $p(\tilde{y})$ is determined once $p(\tilde{y} \mid x)$ or T is fixed
- MCD corrupts class-conditional density: $\boldsymbol{p}_{x|\tilde{y}} = S\boldsymbol{p}_{x|y}$
  - S is a mixture proportion matrix such that $[S]_{i,j} = p(y = j \mid \tilde{y} = i)$
  - It is a "label-noise" model for the corruption of the sampling process

    It is often not viewed as label noise, since instances are also "wrong"
  - $p(\tilde{y})$ is totally free after $p(x \mid \tilde{y})$ or S is fixed
  - Depending on $p(\tilde{y})$, $p(x)$ may notably change (with probability one)

    The only chance of the same $p(x)$ is when MCD is reduced to CCN

    Thus, just the memorization effect can be practically very unreliable

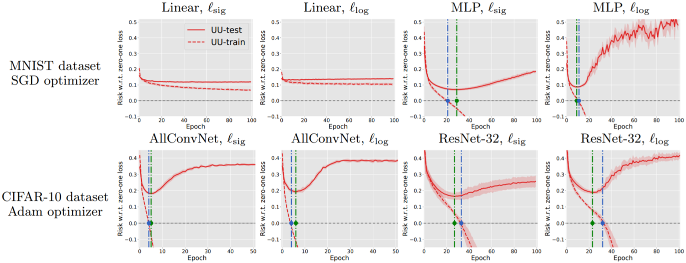
Backward correction for MCD: an overview

- We are going to rewrite the risk $R(g) = \mathbb{E}_{p(x,y)}[\ell(g(x), y)]$
- Specifically, $R(g)$ could be decomposed into c partial risks
- We create a loss ℓ^b, such that $\mathbb{E}_{p(x,\tilde{y})}[\ell^b(g(x), \tilde{y})] = R(g)$
- It could be achieved by solving a set of c^2 linear equations
- The solution is simple: $\ell^b(\cdot, j) = \sum_{k=1}^{c} \frac{[S^{-1}]_{k,j} p(y=k)}{p(\tilde{y}=j)} \ell(\cdot, k)$

Consistent risk correction (Kiryo+, NeurIPS 2017; Lu+, AISTATS 2020)

- However, BC for MCD tends to overfit the training data
  - $\ell^b(\cdot, j)$ is a linear combination but not convex combination of $\{\ell(\cdot, k)\}$
  - We may suffer from that $[U]_{k,j} = [S^{-1}]_{k,j}\pi_k/\tilde{\pi}_j < 0$ for some j and k

    As a result, ℓ^b is not lower bounded, whenever ℓ is not upper bounded



- Aggressive ideas: enforce $[U]_{k,j} \geq 0$ or $\ell^b(g(x_i), \tilde{y}_i) \geq 0$
- Least aggressive idea: just enforce $\widehat{\mathbb{E}}_{p_j(x)}[\ell(g(x), j)] \geq 0$

Connection to learning from unlabeled data

- Binary classification (based on empirical risk minimization)
  - Classifier training is impossible given a single set of U data (Lu+, ICLR 2019)
  - This becomes possible given two sets of U data with different class priors by assuming/forcing $p(y = +1) = \frac{1}{2}$ (du Plessis+, TAAI 2013; Menon+, ICML 2015)
  - $p(y)$ becomes free (Lu+, ICLR 2019), and practical solution (Lu+, AISTATS 2020)
  - Able to train from ≥ 3 different-class-prior U datasets (Lu+, ICML 2021)
- Multi-class classification (based on empirical risk minimization)
  - Should be possible if the number of U datasets = the number of classes
  - However, mapping U datasets to right corrupted classes is combinatorial

Outline

Two ways to go beyond CCN

- Instance-dependent noise (IDN)
  - $\boldsymbol{p}_{\tilde{y}|x} = T(x)^\top \boldsymbol{p}_{y|x}$, the best model for the labeling-process corruption
  - When we confirm/believe $p(x)$ does not change, apply IDN methods
  - Very hard to estimate $T(x)$:

    Rely on part-dependent noise if we can decompose our data into parts

    Rely on confidence-scored IDN if we collected or can assign the scores
- Mutually contaminated distributions (MCD)
  - $\boldsymbol{p}_{x|\tilde{y}} = S\boldsymbol{p}_{x|y}$, the best model for the sampling-process corruption
  - When we confirm/believe $p(x)$ may change, apply MCD methods
  - Very hard to estimate S: Best to (re)label a small subset of data
  - Don't forget learning rate decay and/or consistent risk correction

Future directions

- IDN and MCD are huge future directions of noisy supervisions
  - How to adjust/modify the sample selection/label correction methods for them
- Within IDN
  - What assumptions, besides part-dependent noise, can make $T(x)$ identifiable
  - What information, besides confidence scores, can also help to estimate $T(x)$
- Within MCD
  - How to better mitigate the overfitting of its backward corrections
  - How to accurately estimate S, i.e., the mixture proportion matrix
- Even beyond IDN and MCD
  - A partial label for x_i is a set Y_i of candidate labels, including the true label y_i
  - It belongs to inexact supervision rather than inaccurate/noisy supervision but the key ideas here can be applied (Lv+, ICML 2020; Feng+, ICML 2020 & NeurIPS 2020)

Thanks

Q & A