Label Noisy Representation Learning

Wentao Mo[1]

[1]Department of Machine Intelligence
Peking University

2021 年 4 月 23 日

Outline

LNRL in Chart

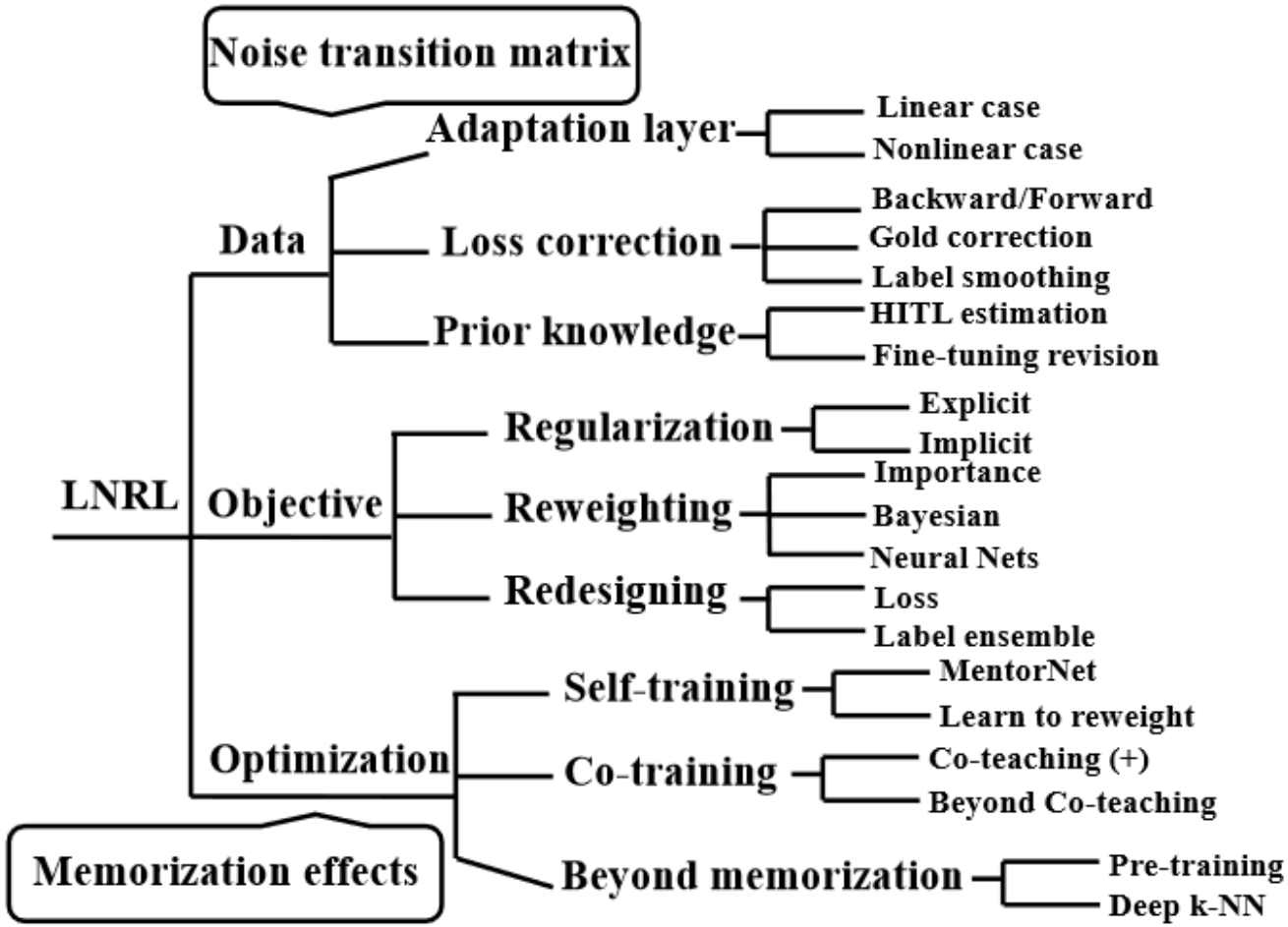

Fig. 2. A taxonomy of LNRL based on the focus of each method. For each technique branch, we list a few representative works here.

Noise Transition Matrix, Forward/Backward Correction

定义

(Noise transition matrix) Suppose that the observed noisy label $\bar{y}$ is drawn independently from a corrupted distribution $p(X, \bar{Y})$, where features are intact. Meanwhile, there exists a corruption process, transition from the latent clean label y to the observed noisy label $\bar{y}$. Such a corruption process can be approximately modeled via a noise transition matrix T, where $T_{ij} = p\left(\bar{y} = e_j \mid y = e_i\right)$

两种经典的 (合成) 噪声转移矩阵, 对称 flipping/配对 flipping

$$\begin{bmatrix} 1-\tau & \frac{\tau}{n-1} & \cdots & \frac{\tau}{n-1} \\ \frac{\tau}{n-1} & 1-\tau & & \frac{\tau}{n-1} \\ \vdots & & \ddots & \vdots \\ \frac{\tau}{n-1} & \frac{\tau}{n-1} & \cdots & 1-\tau \end{bmatrix} \quad \begin{bmatrix} 1-\tau & \tau & 0 & 0 \\ 0 & 1-\tau & \tau & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & & \tau \\ \tau & 0 & \cdots & 1-\tau \end{bmatrix}$$

实际中噪声不一定形式这么好/对称.

Remark 在合成噪声和实际噪声之间存在 domain gap.

Estimate Noise Transition Matrix T

定义

后向矫正

$$\ell^{\leftarrow}(f(x), \bar{y}) = \left[T^{-1}\ell_{y|f(x)}\right]_{\bar{y}} \tag{1}$$

可以证明后向矫正 loss 是 clean label loss 的无偏估计.

定义

前向矫正

$$\ell^{\rightarrow}(f(x), \bar{y}) = \left[\ell_{y|T^{\top}f(x)}\right]_{\bar{y}} \tag{2}$$

可以证明前向矫正 loss 和 clean label loss 上有相同的极小值.

Estimate Noise Transition Matrix T

1. (Patrini et al., 2017) 提出一个两阶段训练. 首先使用 noisy data 训练网络, 再获得一个 T 的估计, 再重新训练网络, 使用 T 校正的 loss.
2. (Hendrycks et al., 2018) 提出了 Gold 校正来处理严重噪声. 关键思路是, 假设一部分数据是可信的且可用的, 比如有一些专家来得出的 trusted set D. 他们使用 D 来估计 T, 再用前向矫正来训练 DNN, 这就是 GLC.
3. 使用 Label Smoothing. 本质上是后向矫正, 且 $T^{-1} = (1-\alpha)I + \frac{\alpha E}{L}$

Estimate Noise by Adaptation Layer

(Sukhbattar, 2015) 提出了在网络输出之后增加一个参数化 T 的 adapt. layer. 单独用 CE 优化两个不同的模块并不能达到 optimal 的 T. 他们又增加了一个 T 的正则化项 trace norm.
(Goldberger et al., 2017) 使用了 base model param. by ω, 以及噪声模型 param. by θ. 既然 base model 的输出是 hidden 的, 那么他们用 EM 算法来估计隐藏输出 (E-step), 以及当前的参数 (M-step). EM 也会导致局部最优和可伸缩性的问题.

Regularization: Explicit

1. (Azadi et al., 2016) 提出了一种正则化项 $\Omega_{\text{aux}}(w) = \|Fw\|_{\text{g}}$ 其中 $\|\cdot\|_g$ 是 group norm, $F^\top = [X_1, \ldots, X_n], X_i = \text{Diag}(\boldsymbol{x}_i)$, 鼓励稀疏性. 这会鼓励一小部分 clean data 来 control model.
2. (Berthelot et al., 2019) 提出了 MixMatch 来进行 SSL. 其中的一个关键部分是 Minimum Ent. Reg.(MER), 也是一种显式正则化. MER 提出于 (Grandvalet & Bengio, 2005), 关键 idea 是把 CE 加入一个正则项, 鼓励在 unlabeled data 上给出 high-confidence 的输出, 具体地, 最小化在 unlabeled 数据上的熵.
3. 类似于 MER, psedo-label 方法 (D.-H. Lee,, 2013)(i.e. lebel guessing) 进行隐式的 ent. 最小化. 具体上讲, 首先计算模型 (通过各种 augmentation) 预测的类型分布, 再通过 temperature sharpening func. 来最小化 label dist. 的熵.

Regularization: Explicit

(Miyato et al., 2018) 提出了一个 virtual adversarial loss, 使用 VA direction, 一种无标签的对抗样本生成法, 类似 FGSM/PGD 但利用了二阶梯度的估计.

定义 $D\left(r, x_*, \theta\right) := D\left[p\left(y \mid x_*, \hat{\theta}\right), p\left(y \mid x_* + r, \theta\right)\right]$ 由于在 $r = 0$ 时, $\nabla_r D(r, x, \hat{\theta})\Big|_{r=0} = 0$, 那么有二阶估计

$$D(r, x, \hat{\theta}) \approx \frac{1}{2} r^T H(x, \hat{\theta}) r \tag{3}$$

那么 r_{vadv} 可以是 Hessian 的第一个 dominant eigenvector 具有长度 ϵ

$$\begin{aligned} r_{\text{vadv}} &\approx \arg\max_r \left\{ r^T H(x, \hat{\theta}) r; \|r\|_2 \leq \epsilon \right\} \\ &= \epsilon \overline{u(x, \hat{\theta})}, \end{aligned} \tag{4}$$

Regularization: Explicit

为了避免直接计算 H, 使用有限差分法来估计这个乘积, 随机采样一个 unit vector$\boldsymbol{d}$, 迭代计算 mat-vec prod. $d \leftarrow \overline{Hd}$.

$$\begin{aligned} Hd &\approx \frac{\nabla_r D(r, x, \hat{\theta})\Big|_{r=\xi d} - \nabla_r D(r, x, \hat{\theta})\Big|_{r=0}}{\xi} \\ &= \frac{\nabla_r D(r, x, \hat{\theta})\Big|_{r=\xi d}}{\xi} \end{aligned} \tag{5}$$

i.e.,

$$d \leftarrow \overline{\nabla_r D(r, x, \hat{\theta})\Big|_{r=\xi d}} \tag{6}$$

他们发现一步迭代就能达到类似 FGSM 里的估计精度.

Regularization: Implicit

(Reed et al., 2015) 的 Bootsrapping. 学习器和自己 bootstrap, 使用 label 和模型目前的 prediction 的凸组合来生成训练目标. 直觉上, 随着 learner 学习, predictions 也变得可信. 进而避免对 noise 的直接建模. 具体地, 有 soft/hard 两种 bootstrapping. 对于 soft bootstrapping, 使用预测的类概率 q 来得到回归目标.

$$\ell_{\text{soft}}(q, t) = \sum_{k=1}^{L} [\beta t_k + (1-\beta) q_k] \log(q_k) \tag{7}$$

这等价于 softmax regression + MER.
对于 hard bootstrapping, 使用 q 的 MAP 估计.

$$\ell_{\text{hard}}(q, t) = \sum_{k=1}^{L} [\beta t_k + (1-\beta) z_k] \log(q_k) \tag{8}$$

其中 $z_k = \mathbf{1}[k = \arg\max_{i=1,\ldots,L} q_i]$ 为了能够优化 hard 版本, 需要使用 EM-like 算法. E-step 中计算凸组合的 targets, M-step 根据 targets 进行优化参数.

Regularization: Implicit

(Zhang et al., 2018) 的 Mixup. 这显然也是一种 label regularization.
(Han et al., 2020) 的 SIGUA(data-agnostic). 注意到随着网络容量的提升, 网络能逐渐地 overfit noisy data. 所以他们提出了 Stochastic Integrated Gradient Underweighted Ascent(SIGUA) 的一种**训练策略**, 在一个 mini-batch 中, 线照常使用 SGD, 再在 bad-data 上使用 (lr 递减的) 梯度递增. 在训练哲学上, SIGUA 让网络忘记不想要的记忆, 来更好的加强想要的记忆.

Objective Reweighting: Importance Reweighting, Bayesian Methods

(Liu and Tao, 2015) 使用 importance reweighting 来 LNL. 将 noisy data 作为 source domain, clean data 作为目标 domain. Idea 是重写经验风险 w.r.t. clean data, 可以得到

$$\beta(X, \bar{Y}) = p_D(\bar{Y} = i \mid X = x) / p_{\bar{D}}(\bar{Y} = i \mid X = x) \tag{9}$$

就是 IW. 这可以通过转移矩阵 T 或者使用小数据集的 clean data(like GLC) 来学到.

(Wang et al., 2017) 的 reweighted prob. models(RPM) 来应对 label noise. Idea 在于, 降低 bad labels 的权重且增加 clean labels 的权重. 具体地,

- 定义概率模型 $p_\beta(\beta) = \prod_{n=1}^{N} \ell\left(y_n \mid \beta\right)$
- 给出 latent weight 的先验分布 $p_w(w), w = (w_1, \ldots, w_N)$

$$p(y, \beta, w) = 1/z \cdot p_\beta(\beta) p_w(w) \prod_{n=1}^{N} \ell\left(y_n \mid \beta\right)^{w_n} \tag{10}$$

- 推理 β, w, 通过后验分布 $p(\beta, w|y)$. 先验分布 $p_w(w)$ 可以是 Beta 分布, scaled Dirichlet 分布, Gamma 分布. 不同的选择 trade off 小概率

Objective Reweighting: Bayesian Methods

(Arazo et al., 2019) 使用了两组分 beta mixture model(BMM), 视为 clean-noisy 的混合, 使用了一个 bootstrapping loss. 具体地, 使用 dynamic weighted boostrapping loss. 数学上, 定义 loss 上的 pdf

$$p(\ell) = \sum_{k=1}^{K} \lambda_k p(\ell \mid k) \tag{11}$$

并且 $p(\ell \mid k)$ 可以使用 Beta 分布建模.

$$p(\ell \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \ell^{\alpha-1}(1-\ell)^{\beta-1} \tag{12}$$

上述问题可以使用 EM 算法来解决.
更具体地, 引入 $\gamma_k(\ell) = p(k \mid \ell)$, E-step 中固定 λ, α, β, 计算 γ. M-step 中固定 γ, 使用带权动量估计 α, β, 动态权重则使用简单的方法来得到 $\lambda_k = \frac{1}{N}\sum_{i=1}^{N} \gamma_k(\ell_i)$. 基于这个 BMM 模型, 他们还提出了动态 hard/soft bootstrapping loss, 其中每个 sample 的 weight 动态的设置为 $p(k=1 \mid \ell_i)$(sample 为 clean 的概率).

Objective Reweighting: NNs

(Shu et al., 2019) 使用 Meta-Weight-Net(MW-Net) 来学习显示的 weighting function. w. func. 是一个单层 MLP, 从 loss 到 weight. 数学上

$$w^*(\theta) = \arg\min_w \ell^{\text{tr}}(w;\theta) = 1/N \sum_{i=1}^{N} \mathcal{V}\left(t_i^{\text{tr}}(w);\theta\right) \ell_i^{\text{tr}}(w) \tag{13}$$

这里, 可以使用元学习来优化 MW-Net: 给出一些 clean, balanced 元数据 $\left\{x_i^{(\text{meta})}, y_i^{(\text{meta})}\right\}_{i=1}^{M}$, 最小化 meta-loss

$$\theta^* = \arg\min_\theta \ell^{\text{meta}}\ (w^*(\theta)) = 1/M \sum_{i=1}^{M} \ell_i^{\text{meta}}\ (w^*(\theta))\ . \tag{14}$$

使用 SGD 迭代的分别更新 w 和 θ