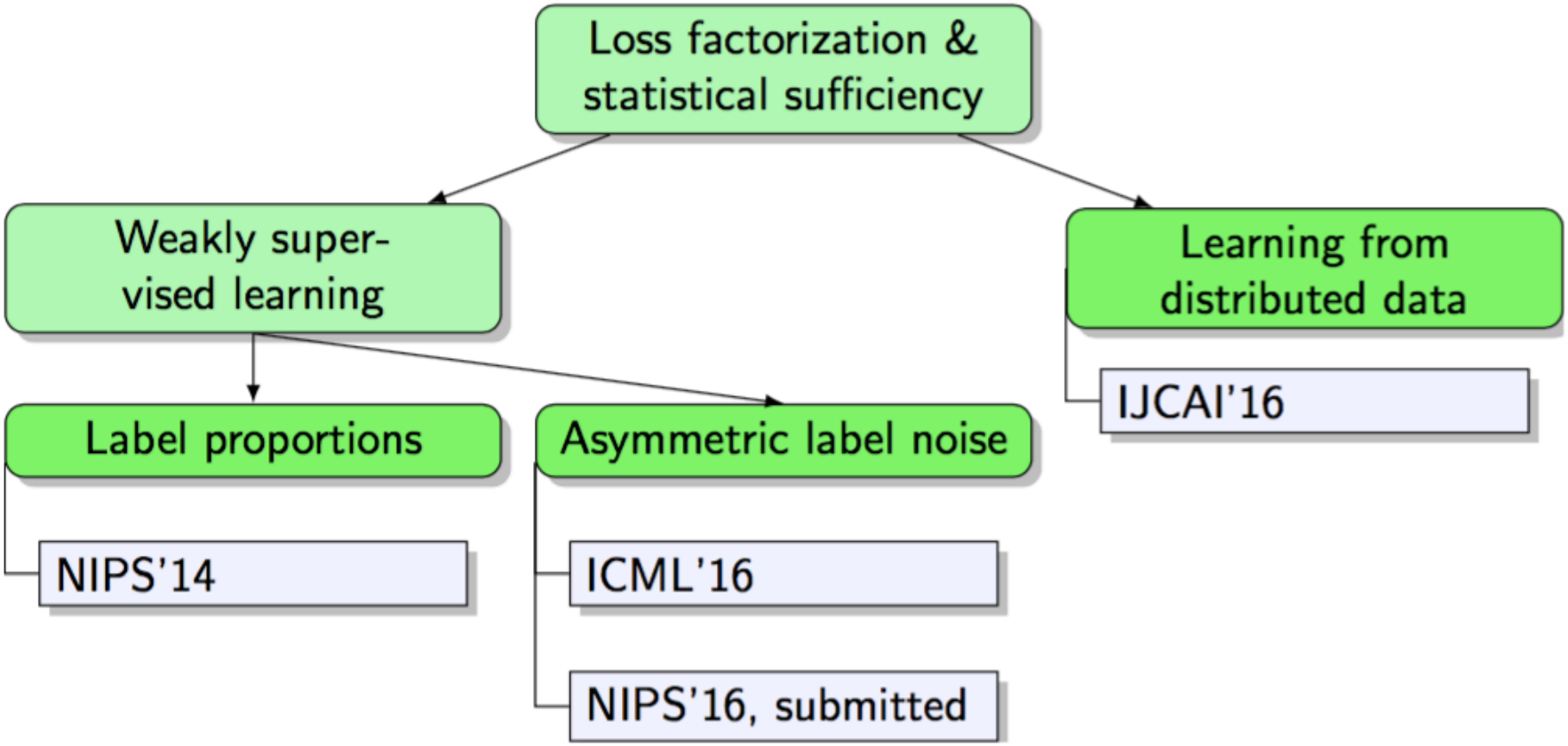
Weakly supervised learning via statistical sufficiency

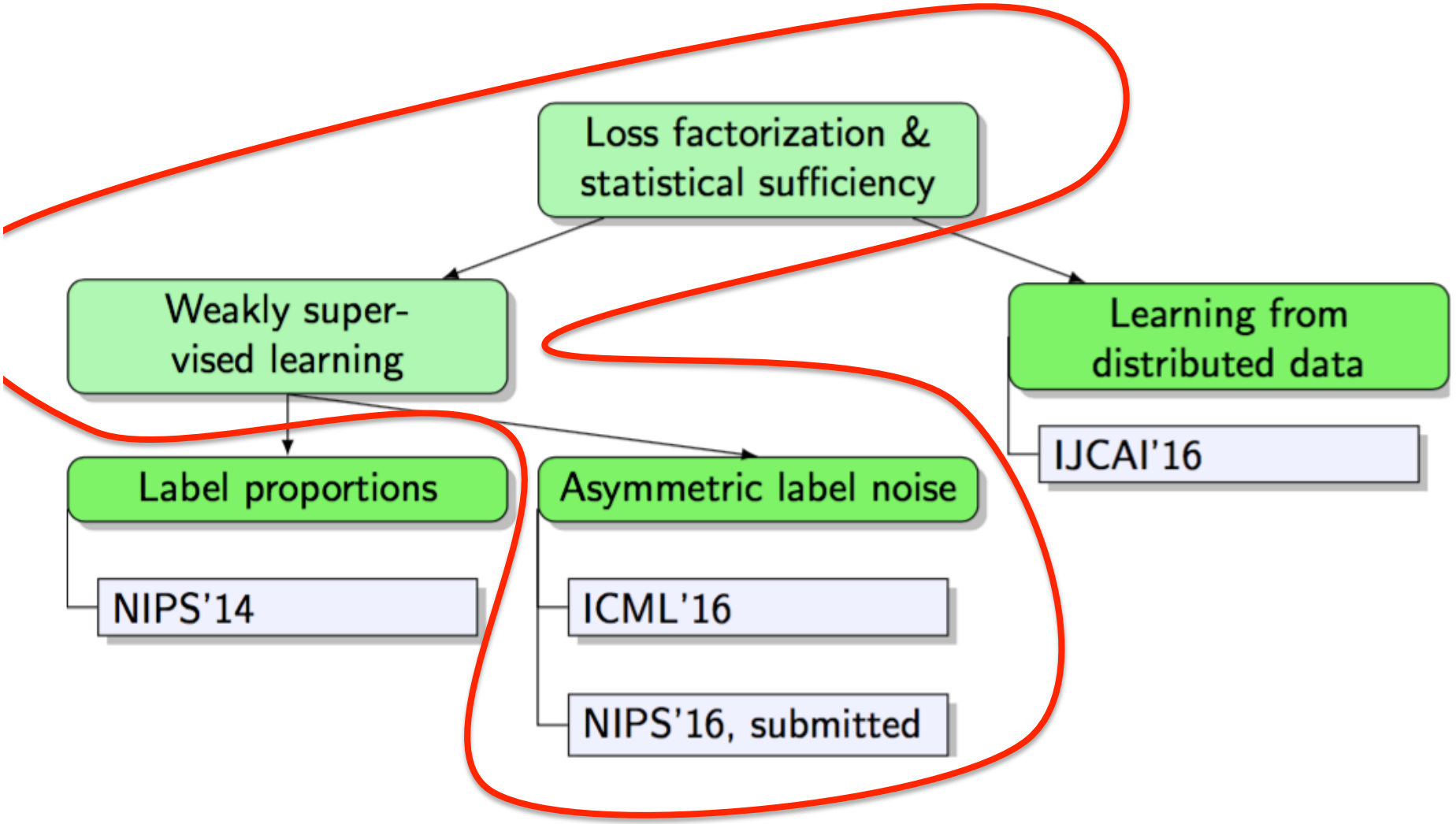
Giorgio Patrini

4 August 2016

Content organization



Content organization



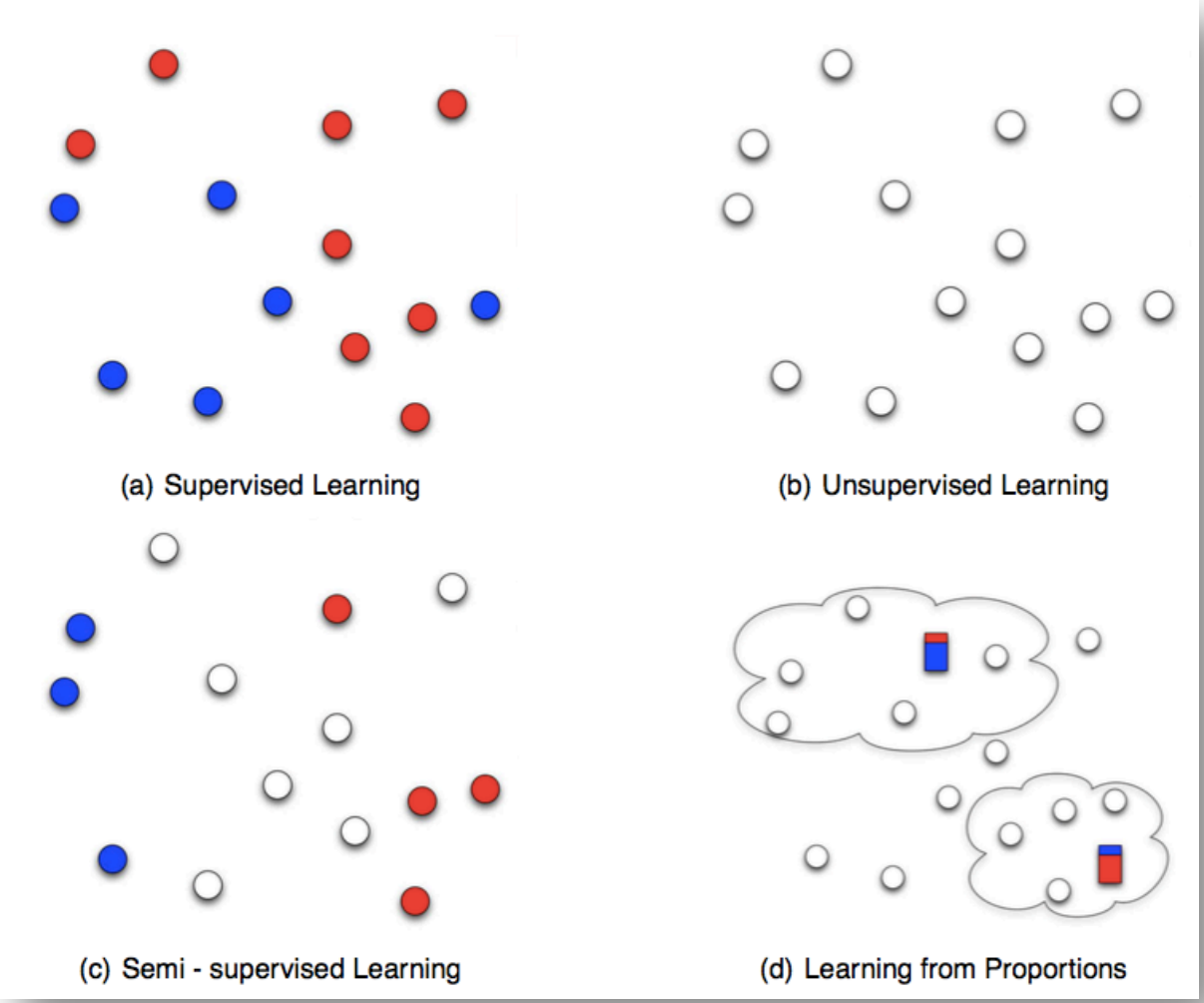
Outline

- Motivations
- Statistical sufficiency & loss factorization
- Asymmetric label noise I: theory with linear models
- Asymmetric label noise II: neural nets
- Other learning settings and conclusion

Outline

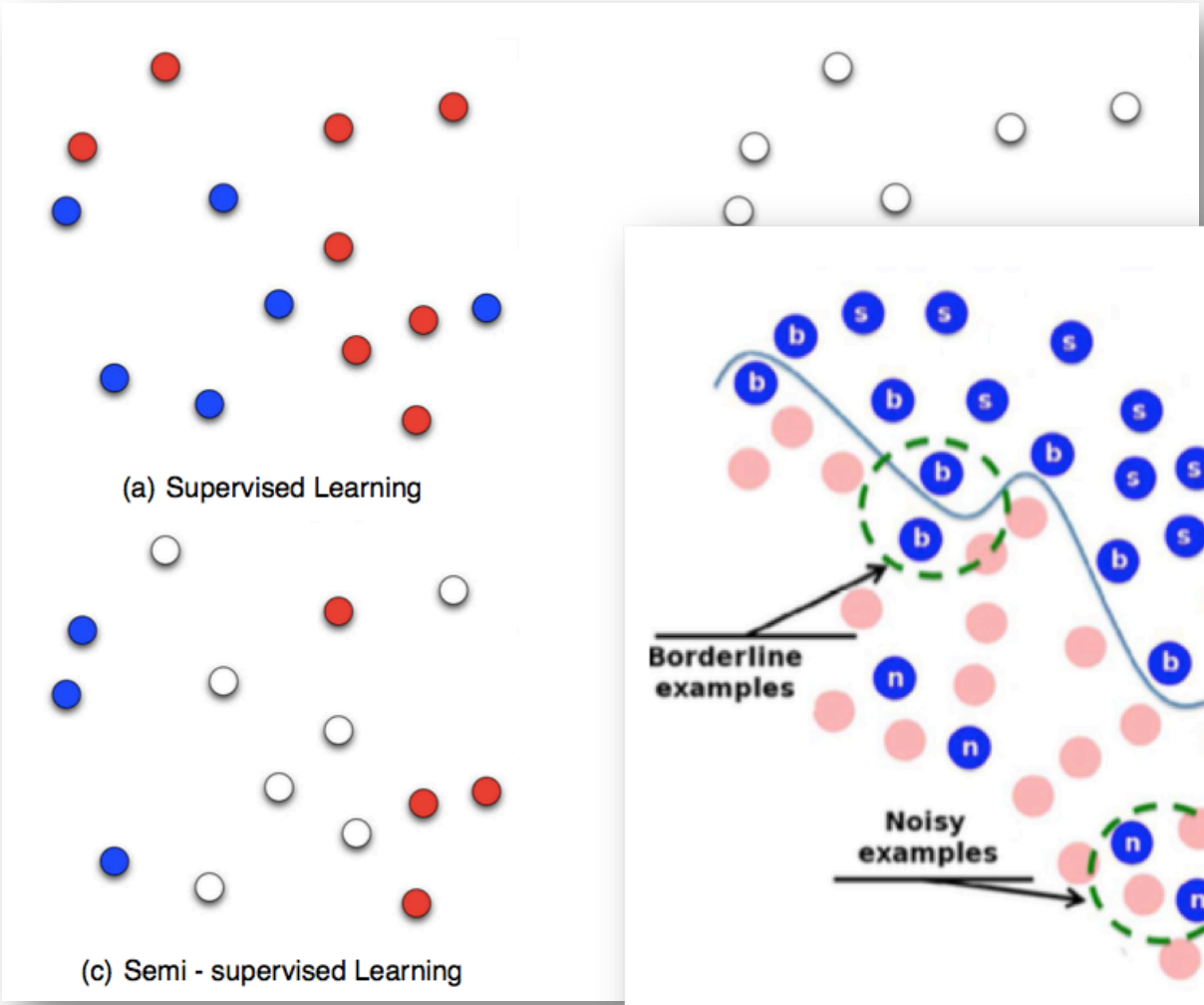
- **Motivations**
- Statistical sufficiency & loss factorization
- Asymmetric label noise I: theory with linear models
- Asymmetric label noise II: neural nets
- Other learning settings and conclusion

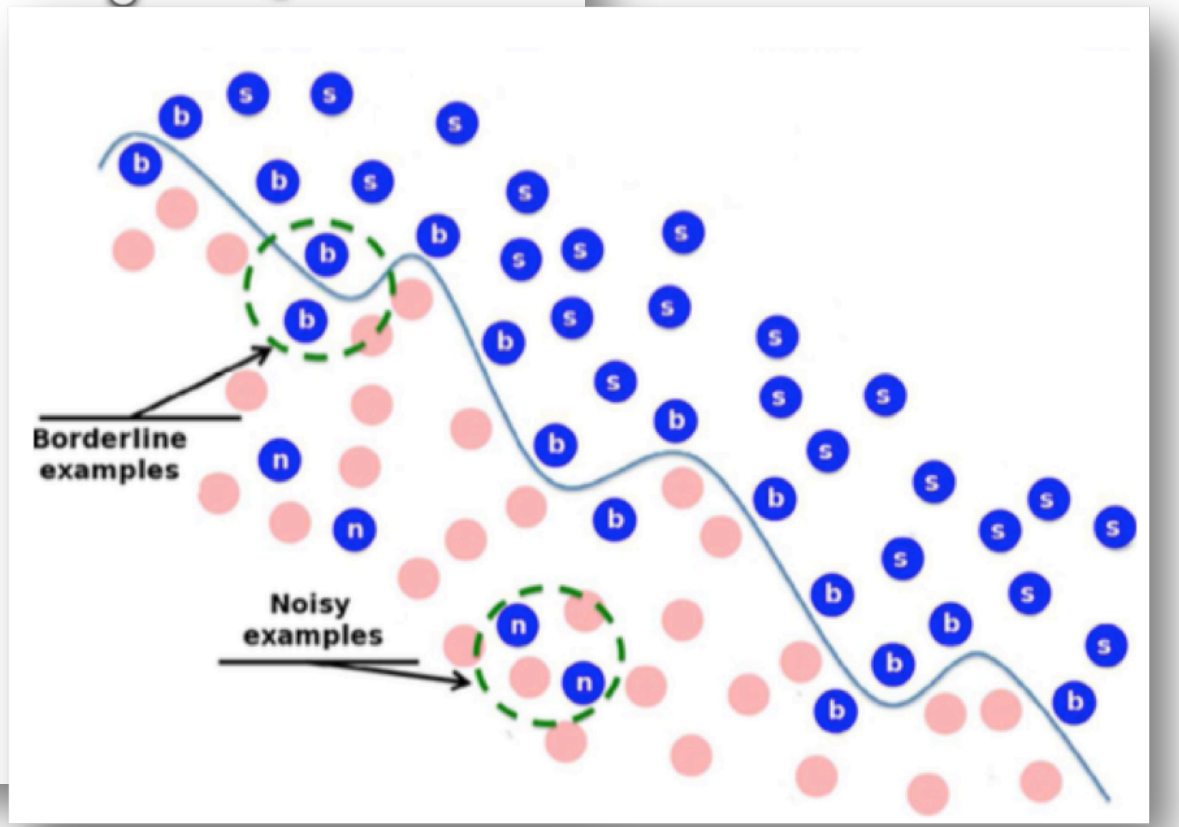
Motivations



[Quadrianto et al.'09]

Motivations





[sci2s.ugr.es/noisydata]

(Fully) supervised learning

- Binary classification

  $\mathcal{S} = \{(\boldsymbol{x}_i, y_i), i \in [m]\}$ sampled from $\mathcal{D}$ over $\mathbb{R}^d \times \{-1, 1\}$

  $\{1, \ldots, m\}$

- Learn a linear model $h \in \mathcal{H}$
- Minimize the empirical risk associated with a surrogate loss $\ell(x)$

$$\operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{S}} \left[\ell(yh(\boldsymbol{x}))\right] = \operatorname*{argmin}_{h \in \mathcal{H}} R_{\mathcal{S},\ell}(h)$$

Weakly supervised learning

$$\mathcal{D} \xrightarrow{corrupt} \tilde{\mathcal{D}} \xrightarrow{sample} \tilde{\mathcal{S}}$$

- **Weak** labels may be wrong, missing, multi-instance label constraints, etc.
- Marginal of X is the same
- The goal is unchanged: generalize to $\mathcal{D}$

Risk with weak labels is non-sense

$$\mathbb{E}_{\tilde{S}}\left[\ell(yh(\boldsymbol{x}))\right] \qquad \textbf{?}$$

A sample of proposed solutions

- Individual labels are corrupted: engineer a robust loss

$$\operatorname*{argmin}_{h} \mathbb{E}_{\tilde{S}}[\tilde{\ell}(yh(\boldsymbol{x}))]$$

A sample of proposed solutions

- Individual labels are corrupted: engineer a robust loss

$$\operatorname*{argmin}_{h} \mathbb{E}_{\tilde{S}}[\tilde{\ell}(yh(\boldsymbol{x}))]$$

- Some labels are missing: regularize with unlabelled data

$$\operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_{\tilde{S}}[\ell(yh(\boldsymbol{x}))] + \lambda \ \textsc{Reg}(\{\boldsymbol{x}\}, h)$$

A sample of proposed solutions

- Individual labels are corrupted: engineer a robust loss

$$\operatorname*{argmin}_{h} \mathbb{E}_{\tilde{S}}[\tilde{\ell}(yh(\boldsymbol{x}))]$$

- Some labels are missing: regularize with unlabelled data

$$\operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_{\tilde{S}}[\ell(yh(\boldsymbol{x}))] + \lambda \ \textsc{Reg}(\{\boldsymbol{x}\}, h)$$

- Label knowledge on sets: enforce the constraints

$$\operatorname*{argmin}_{h \in \mathcal{H}, y \in \{\pm 1\}} \mathbb{E}_{\tilde{S}}[\ell(yh(\boldsymbol{x}))] + \lambda \ \textsc{Constr}(\{(\boldsymbol{x}, y)\}, h)$$

Drawbacks

A mix of

- Need to dream up new:
  - losses (possibly non-convex)
  - regularizers/constraints
  - optimization algorithms
- Label as latent variable => non-convex objective
- No unified approach

Solution principles

- Divide and conquer (decoupling)

  Treat labels issues and learning separately

- Do not reinvent the wheel (modularity)

  Let's reuse well-known algorithms for risk minimization

- Computational laziness

  "One should solve the problem directly and never solve a more general problem as an intermediate step" [V. Vapnik'98]

- Statistical sufficiency

  If we know what is **sufficient** for learning, poor labels are not a problem

A 2-step framework

(1) Estimate a sufficient statistic $\boldsymbol{\mu}$ for the labels from weak supervision

$$\tilde{\mathcal{S}} \to \boldsymbol{\mu}$$

(2) Plug it into a standard loss ℓ and call any algorithm for empirical risk minimization

$$\operatorname*{argmin}_{h \in \mathcal{H}} R_{\tilde{S},\ell}(h, \boldsymbol{\mu})$$

A unifying approach

Learning from label proportions with
- logistic loss [Quadrianto et al. '09]
- symmetric proper loss [Patrini et al. '14]

Learning with noisy labels with
- logistic loss [Gao et al. '16]

"Indirect supervision" with
- logistic loss [Raghunathan et al. '16]

Outline

- Motivations
- **Statistical sufficiency & loss factorization**
- Asymmetric label noise I: theory with linear models
- Asymmetric label noise II: neural nets
- Other learning settings and conclusion

Statistical sufficiency

- Intuition: a sufficient statistic aggregates from data all information about the model parameters

Statistical sufficiency

- Intuition: a sufficient statistic aggregates from data all information about the model parameters
- Definition: μ is sufficient for θ wrt. Y when for each pair of outcomes y, y' we have

$$\frac{P(\theta|\mu(y))}{P(\theta|\mu(y'))} \text{ does not depend on } Y \iff \mu(y) = \mu(y')$$

Exponential family ⇔ logistic loss

- Conditional exponential family

$$p(y|\boldsymbol{x}) = \exp\{\langle \boldsymbol{\theta}, y\boldsymbol{x}\rangle \ - \log \textstyle\sum_y \exp\langle \boldsymbol{\theta}, y\boldsymbol{x}\rangle\}$$

label independent

Exponential family ⇔ logistic loss

- Conditional exponential family

$$p(y|\boldsymbol{x}) = \exp\{\langle \boldsymbol{\theta}, y\boldsymbol{x}\rangle \ - \log \textstyle\sum_y \exp\langle \boldsymbol{\theta}, y\boldsymbol{x}\rangle\}$$

label independent

- Log-likelihood leads to logistic loss

$$\sum_{i=1}^{m} \log \sum_y e^{y\langle \boldsymbol{\theta}, \boldsymbol{x}_i\rangle} - \langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle = \sum_{i=1}^{m} \log\left(1 + e^{-2y_i\langle \boldsymbol{\theta}, \boldsymbol{x}_i\rangle}\right)$$

sufficiency of $\boldsymbol{\mu}$ for y

Mean operator & linear-odd losses

- Mean operator

$$\boldsymbol{\mu}_S \doteq \mathbb{E}_S[y\boldsymbol{x}] = \frac{1}{m}\sum_{i=1}^{m} y_i \boldsymbol{x}_i \in \mathbb{R}^d$$

Mean operator & linear-odd losses

- Mean operator

$$\boldsymbol{\mu}_S \doteq \mathbb{E}_S[y\boldsymbol{x}] = \frac{1}{m}\sum_{i=1}^{m} y_i \boldsymbol{x}_i \in \mathbb{R}^d$$

- Linear-odd loss, a-LOL

$$\exists a \in \mathbb{R},\ \tfrac{1}{2}\ (\ell(x) - \ell(-x)) = \ell_o(x) = ax$$

generic x argument

Loss factorization

- Linear model h
- Linear-odd loss $\frac{1}{2}(\ell(x) - \ell(-x)) = \ell_o(x) = ax$
- Given a sample $\mathcal{S}$, define a "double sample"

$$\mathcal{S}_{2x} \doteq \{(\boldsymbol{x}_i, \sigma), i \in [m], \sigma \in \{\pm 1\}\}$$

Loss factorization

- Linear model h
- Linear-odd loss $\frac{1}{2}(\ell(x) - \ell(-x)) = \ell_o(x) = ax$
- Given a sample $\mathcal{S}$, define a "double sample"

$$\mathcal{S}_{2x} \doteq \{(\boldsymbol{x}_i, \sigma), i \in [m], \sigma \in \{\pm 1\}\}$$

Then:

$$R_{\mathcal{S},\ell}(h) = \frac{1}{2} R_{\mathcal{S}_{2x},\ell}(h) + a \cdot h(\boldsymbol{\mu}_{\mathcal{S}})$$

label independent

smoothness nor convexity of ℓ required

Loss factorization: proof

$$
\begin{aligned}
&R_{\mathcal{S},\ell}(h) = \\
&= \mathbb{E}_{\mathcal{S}}\Big[\ell(yh(\boldsymbol{x}))\Big] \\
&= \frac{1}{2}\mathbb{E}_{\mathcal{S}}\Big[\ell(yh(\boldsymbol{x})) + \ell(-yh(\boldsymbol{x})) + \ell(yh(\boldsymbol{x})) - \ell(-yh(\boldsymbol{x}))\Big] \\
&= \frac{1}{2}\mathbb{E}_{\mathcal{S}_{2x}}\Big[\ell(\sigma h(\boldsymbol{x}))\Big] + \mathbb{E}_{\mathcal{S}}\Big[\ell_o(h(y\boldsymbol{x}))\Big] \\
&= \frac{1}{2}R_{\mathcal{S}_{2x},\ell}(h) + a \cdot h(\boldsymbol{\mu}_{\mathcal{S}})
\end{aligned}
$$

even + odd

linear ℓ_o and h

sufficiency of $\boldsymbol{\mu}$ for y

Sufficiency with losses

$\boldsymbol{\mu}$ is sufficient for y when $\forall \ell \in \mathcal{L}, \forall h \in \mathcal{H}$
and for each pair of learning samples $\mathcal{S}, \mathcal{S}'$:

$$R_{\mathcal{S},\ell}(h) - R_{\mathcal{S}',\ell}(h) \text{ does not depend on } y \iff \boldsymbol{\mu}_{\mathcal{S}} = \boldsymbol{\mu}_{\mathcal{S}'}$$

Sufficiency with losses

$\boldsymbol{\mu}$ is sufficient for y when $\forall \ell \in \mathcal{L}, \forall h \in \mathcal{H}$
and for each pair of learning samples $\mathcal{S}, \mathcal{S}'$:

$$R_{\mathcal{S},\ell}(h) - R_{\mathcal{S}',\ell}(h) \text{ does not depend on } y \iff \boldsymbol{\mu}_{\mathcal{S}} = \boldsymbol{\mu}_{\mathcal{S}'}$$

Sufficiency as a consequence of factorization

$$R_{\mathcal{S},\ell}(h) = \frac{1}{2} R_{\mathcal{S}_{2x},\ell}(h) + a \cdot h(\boldsymbol{\mu}_{\mathcal{S}})$$

Linear-odd losses: examples

	loss ℓ	odd term ℓ_o
LOL	$\ell(x)$	ax
ρ-loss	$\rho\lvert x\rvert - \rho x + 1$	$-\rho x \ \ (\rho \geq 0)$
unhinged	$1 - x$	$-x$
perceptron	$\max(0, -x)$	$-x$
double-hinge	$\max(-x, 1/2 \max(0, 1 - x))$	$-x$
SPL	$a_\ell + \ell^\star(-x)/b_\ell$	$-x/(2b_\ell)$
logistic	$\log(1 + e^{-x})$	$-x/2$
square	$(1 - x)^2$	$-2x$
Matsushita	$\sqrt{1 + x^2} - x$	$-x$

Linear-odd losses: examples

	loss ℓ	odd term ℓ_o
LOL	$\ell(x)$	ax
ρ-loss	$\rho\lvert x\rvert - \rho x + 1$	$-\rho x \ \ (\rho \geq 0)$
unhinged	$1 - x$	$-x$
perceptron	$\max(0, -x)$	$-x$
double-hinge	$\max(-x, 1/2 \max(0, 1 - x))$	$-x$
SPL	$a_\ell + \ell^\star(-x)/b_\ell$	$-x/(2b_\ell)$
logistic	$\log(1 + e^{-x})$	$-x/2$
square	$(1 - x)^2$	$-2x$
Matsushita	$\sqrt{1 + x^2} - x$	$-x$

- Bonus: convex LOLs are calibrated when $a < 0$

Generalization bound

- Loss is a-LOL and Lipschitz
- Bounded feature and model spaces
- Bounded loss $C \doteq \max_x \ell(x)$
- Let $\hat{\boldsymbol{\theta}} \doteq \operatorname{argmin}_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{S},\ell}(\boldsymbol{\theta})$

Generalization bound

- Loss is a-LOL and Lipschitz
- Bounded feature and model spaces
- Bounded loss $C \doteq \max_x \ell(x)$
- Let $\hat{\boldsymbol{\theta}} \doteq \operatorname{argmin}_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{S},\ell}(\boldsymbol{\theta})$

Then for any $\delta > 0$, with probability at least $1 - \delta$:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - \inf_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \leq O\left(\frac{1}{\sqrt{m}}\right) +$$

$$C \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right) + |a| \cdot O\left(\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2\right)$$

Generalization bound

- Loss is a-LOL and Lipschitz
- Bounded feature and model spaces
- Bounded loss $C \doteq \max_x \ell(x)$
- Let $\hat{\boldsymbol{\theta}} \doteq \operatorname{argmin}_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{S},\ell}(\boldsymbol{\theta})$

Then for any $\delta > 0$, with probability at least $1 - \delta$:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - \inf_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \leq O\left(\frac{1}{\sqrt{m}}\right) +$$

complexity of space $\mathcal{H}$

$$C \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right) + |a| \cdot O\left(\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2\right)$$

loss non-linearity

$$|a| \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right)$$

Example: SGD (step 2)

Algorithm SGD

Input: $\mathcal{S}$, ℓ;

$m \leftarrow |\mathcal{S}|$

$\boldsymbol{\theta}^0 \leftarrow \mathbf{0}$

For any $t = 1, 2, \ldots$:

Sample $i \sim U([m])$

$\eta^t \leftarrow 1/t$

Pick any $\boldsymbol{v} \in \partial \ell(y_i \langle \boldsymbol{\theta}^t, \boldsymbol{x}_i \rangle)$

$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta^t \boldsymbol{v}$

Output: $\boldsymbol{\theta}^{t+1}$

Example: SGD (step 2)

Algorithm $\boldsymbol{\mu}$SGD

Input: $\mathcal{S}_{2x}, \boldsymbol{\mu}$, ℓ is a-LOL;

$m \leftarrow |\mathcal{S}_{2x}|$

$\boldsymbol{\theta}^0 \leftarrow \mathbf{0}$

For any $t = 1, 2, \ldots$:

Sample $i \sim U([m])$

$\eta^t \leftarrow 1/t$

Pick any $\boldsymbol{v} \in \partial \ell(y_i \langle \boldsymbol{\theta}^t, \boldsymbol{x}_i \rangle)$

$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta^t(\boldsymbol{v} + a\boldsymbol{\mu}/2)$

Output: $\boldsymbol{\theta}^{t+1}$

only changes wrt SGD

Example: SGD (step 2)

Algorithm $\boldsymbol{\mu}$SGD

Input: $\mathcal{S}_{2x}, \boldsymbol{\mu}$, ℓ is a-LOL;

$m \leftarrow |\mathcal{S}_{2x}|$

$\boldsymbol{\theta}^0 \leftarrow \mathbf{0}$

For any $t = 1, 2, \ldots$:

Sample $i \sim U([m])$

$\eta^t \leftarrow 1/t$

Pick any $\boldsymbol{v} \in \partial \ell(y_i \langle \boldsymbol{\theta}^t, \boldsymbol{x}_i \rangle)$

$\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta^t(\boldsymbol{v} + a\boldsymbol{\mu}/2)$

Output: $\boldsymbol{\theta}^{t+1}$

only changes wrt SGD

- Similar with proximal algorithms

Outline

- Motivations
- Statistical sufficiency & loss factorization
- **Asymmetric label noise I: theory with linear models**
- Asymmetric label noise II: neural nets
- Other learning settings and conclusion

Asymmetric label noise

Sample $\tilde{\mathcal{S}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^m$ corrupted by **asymmetric** noise rates p_+, p_-

Asymmetric label noise

Sample $\tilde{\mathcal{S}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^m$ corrupted by **asymmetric** noise rates p_+, p_-

By the method of [Natarajan et al. '13] an **unbiased** estimator of $\boldsymbol{\mu}_{\mathcal{S}}$ is

$$\hat{\boldsymbol{\mu}}_{\mathcal{S}} \doteq \mathbb{E}_{\tilde{\mathcal{S}}}\left[\frac{y - (p_- - p_+)}{1 - p_- - p_+}\boldsymbol{x}\right], \quad \mathbb{E}_{\tilde{D}}[\hat{\boldsymbol{\mu}}_{\mathcal{S}}] = \boldsymbol{\mu}_{\mathcal{S}}$$

Asymmetric label noise

Sample $\tilde{\mathcal{S}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^m$ corrupted by **asymmetric** noise rates p_+, p_-

By the method of [Natarajan et al. '13] an **unbiased** estimator of $\boldsymbol{\mu}_{\mathcal{S}}$ is

$$\hat{\boldsymbol{\mu}}_{\mathcal{S}} \doteq \mathbb{E}_{\tilde{\mathcal{S}}}\left[\frac{y - (p_- - p_+)}{1 - p_- - p_+}\boldsymbol{x}\right], \quad \mathbb{E}_{\tilde{D}}[\hat{\boldsymbol{\mu}}_{\mathcal{S}}] = \boldsymbol{\mu}_{\mathcal{S}}$$

This is step (1), then run $\boldsymbol{\mu}$-SGD for (2).

Generalization bound under noise

Same as before, except that now we learn with $\hat{\boldsymbol{\mu}}_S$

Then for any $\delta > 0$, with probability at least $1 - \delta$:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - \inf_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \leq O\left(\frac{1}{\sqrt{m}}\right) +$$

$$C \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right) + \frac{|a|}{1 - p_- - p_+} \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right)$$

Generalization bound under noise

Same as before, except that now we learn with $\hat{\boldsymbol{\mu}}_S$

Then for any $\delta > 0$, with probability at least $1 - \delta$:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - \inf_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \leq O\left(\frac{1}{\sqrt{m}}\right) +$$

complexity untouched

$$C \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right) + \frac{|a|}{1 - p_- - p_+} \cdot O\left(\frac{1}{\sqrt{m}}, \log \frac{1}{\delta}\right)$$

noise affects the linear term only

Empirics

- Artificially corrupted data. Noise rates up to ~50%
- SGD vs. μ -SGD with the same parameters
- Test error average difference over 25 runs

$(p_-, p_+) \rightarrow$	(.00, .00)		(.20, .00)		(.20, .10)		(.20, .20)		(.20, .30)		(.20, .40)		(.20, .49)	
dataset	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD
australian	0.13	+.01	0.15	**−.01**	0.14	±.00	0.14	+.01	0.16	**−.01**	0.26	**−.09**	0.45	**−.25**
breast-can.	0.02	+.01	0.03	±.00	0.03	±.00	0.03	±.00	0.05	**−.01**	0.11	**−.06**	0.17	**−.08**
diabetes	0.28	**−.03**	0.29	**−.03**	0.29	**−.03**	0.27	**−.02**	0.28	**−.02**	0.39	**−.13**	0.59	**−.22**
german	0.27	**−.02**	0.26	±.00	0.27	**−.02**	0.29	**−.02**	0.31	**−.01**	0.31	±.00	0.31	±.00
heart	0.15	+.01	0.17	**−.01**	0.16	±.00	0.17	±.00	0.18	**−.01**	0.26	**−.08**	0.35	**−.15**
housing	0.17	**−.03**	0.23	**−.05**	0.22	**−.04**	0.20	**−.02**	0.22	**−.03**	0.34	**−.12**	0.41	**−.13**
ionosphere	0.14	+05	0.19	**−.05**	0.20	**−.05**	0.20	**−.03**	0.21	**−.03**	0.35	**−.13**	0.54	**−.29**
sonar	0.27	±.00	0.29	+.02	0.29	+.01	0.34	**−.04**	0.36	**−.03**	0.43	**−.10**	0.45	**−.05**

Empirics

- Artificially corrupted data. Noise rates up to ~50%
- SGD vs. μ -SGD with the same parameters
- Test error average difference over 25 runs

$(p_-, p_+) \rightarrow$	(.00, .00)		(.20, .00)		(.20, .10)		(.20, .20)		(.20, .30)		(.20, .40)		(.20, .49)	
dataset	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD
australian	0.13	+.01	0.15	**−.01**	0.14	±.00	0.14	+.01	0.16	**−.01**	0.26	**−.09**	0.45	**−.25**
breast-can.	0.02	+.01	0.03	±.00	0.03	±.00	0.03	±.00	0.05	**−.01**	0.11	**−.06**	0.17	**−.08**
diabetes	0.28	**−.03**	0.29	**−.03**	0.29	**−.03**	0.27	**−.02**	0.28	**−.02**	0.39	**−.13**	0.59	**−.22**
german	0.27	**−.02**	0.26	±.00	0.27	**−.02**	0.29	**−.02**	0.31	**−.01**	0.31	±.00	0.31	±.00
heart	0.15	+.01	0.17	**−.01**	0.16	±.00	0.17	±.00	0.18	**−.01**	0.26	**−.08**	0.35	**−.15**
housing	0.17	**−.03**	0.23	**−.05**	0.22	**−.04**	0.20	**−.02**	0.22	**−.03**	0.34	**−.12**	0.41	**−.13**
ionosphere	0.14	+05	0.19	**−.05**	0.20	**−.05**	0.20	**−.03**	0.21	**−.03**	0.35	**−.13**	0.54	**−.29**
sonar	0.27	±.00	0.29	+.02	0.29	+.01	0.34	**−.04**	0.36	**−.03**	0.43	**−.10**	0.45	**−.05**

=> Still able to learn with one label ~ random

Outline

- Motivations
- Statistical sufficiency & loss factorization
- Asymmetric label noise I: theory with linear models
- **Asymmetric label noise II: neural nets**
- Other learning settings and conclusion

Extended setting

- Multi-class (one-hot): $\boldsymbol{y} \in \{\boldsymbol{e}^i : i \in [c]\}$

=> Noise is a transition matrix T

$$T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j | \boldsymbol{y} = \boldsymbol{e}^i)$$

- Estimation of the matrix from noisy data

$$\tilde{\mathcal{S}} \rightarrow \hat{T}$$

- Neural networks

$$\boldsymbol{h}(\boldsymbol{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^c$$

Neural network

- For n layers ($\boldsymbol{z}$ generic input)

$e.g.$ ReLU, $\sigma = \max(0, z)$

$$(\forall i \in [n-1])\, \boldsymbol{h}^{(i)}(\boldsymbol{z}) = \boldsymbol{\sigma}(W^{(i)}\boldsymbol{z} + \boldsymbol{b}^{(i)}) \ ,$$

$$\boldsymbol{h}^{(n)}(\boldsymbol{z}) = W^{(n)}\boldsymbol{z} + \boldsymbol{b}^{(n)}$$

$$\Rightarrow \ \boldsymbol{h} = (\boldsymbol{h}^{(n)} \circ \boldsymbol{h}^{(n-1)} \circ \cdots \circ \boldsymbol{h}^{(1)})$$

- Cross-entropy loss

$$\boldsymbol{y}^\top \boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = -\boldsymbol{y}^\top \log \mathrm{softmax}(\boldsymbol{h}(\boldsymbol{x}))$$

No trivial decoupling

- Cross-entropy (linear-odd) factorization

$$\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = -\left(W^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) + \boldsymbol{b}^n\right) + \mathbf{1} \cdot \log \sum_{j \in [c]} \exp\{W_j^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) + \boldsymbol{b}_j^n\}$$

last layer = linear model

label-independent

No trivial decoupling

- Cross-entropy (linear-odd) factorization

$$\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = -\left(W^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) + \boldsymbol{b}^n\right) + \mathbf{1} \cdot \log \sum_{j \in [c]} \exp\{W_j^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) + \boldsymbol{b}_j^n\}$$

last layer = linear model

label-independent

- The "sufficient statistic" for the labels is a function of all layers but the last: $\mathbb{E}_S[\boldsymbol{h}^{n-1}(\boldsymbol{x})]$

No trivial decoupling

- Cross-entropy (linear-odd) factorization

$$\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = -\left(W^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) + \boldsymbol{b}^n\right) + \mathbf{1} \cdot \log \sum_{j \in [c]} \exp\{W_j^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) + \boldsymbol{b}_j^n\}$$

last layer = linear model

label-independent

- The "sufficient statistic" for the labels is a function of all layers but the last: $\mathbb{E}_S[\boldsymbol{h}^{n-1}(\boldsymbol{x})]$
- But: if we know the noise T, we can make the statistic unbiased **while training**. By [Natarajan et al. '13] :

$$\mathbb{E}_S[T^{-1}(-W^n \boldsymbol{h}^{n-1}(\boldsymbol{x}) - \boldsymbol{b}^n)]$$

is unbiased and so is the whole loss

Noise estimation [Menon et al. '15]

- Train on noisy data and obtain $p(\tilde{y}|\boldsymbol{x})$
- Then estimate $\hat{T}$ by

$$\forall i, j \left[\begin{array}{l} \bar{\boldsymbol{x}}^i = \underset{(\boldsymbol{x},\cdot)\in\tilde{S}}{\operatorname{argmax}}\, p(\tilde{y} = \boldsymbol{e}^i|\boldsymbol{x}) \\ \\ T_{ij} = p(\tilde{y} = \boldsymbol{e}^j|\bar{\boldsymbol{x}}^i) \end{array} \right.$$

- **Hp**: there are some "perfect examples", and the net models $p(\tilde{y}|\boldsymbol{x})$ perfectly

Noise estimation [Menon et al. '15]

- Train on noisy data and obtain $p(\tilde{y}|\boldsymbol{x})$
- Then estimate $\hat{T}$ by

$$\forall i, j \left[\begin{array}{l} \bar{\boldsymbol{x}}^i = \underset{(\boldsymbol{x},\cdot)\in\tilde{S}}{\operatorname{argmax}}\, p(\tilde{y} = \boldsymbol{e}^i|\boldsymbol{x}) \\ \\ T_{ij} = p(\tilde{y} = \boldsymbol{e}^j|\bar{\boldsymbol{x}}^i) \end{array} \right.$$

- **Hp**: there are some "perfect examples", and the net models $p(\tilde{y}|\boldsymbol{x})$ perfectly
- **Rational**: mistakes on "perfect examples" must be due to the noise

2-step solution with neural nets

(1) Train the network on $\tilde{\mathcal{S}}$ to obtain $\hat{T}$

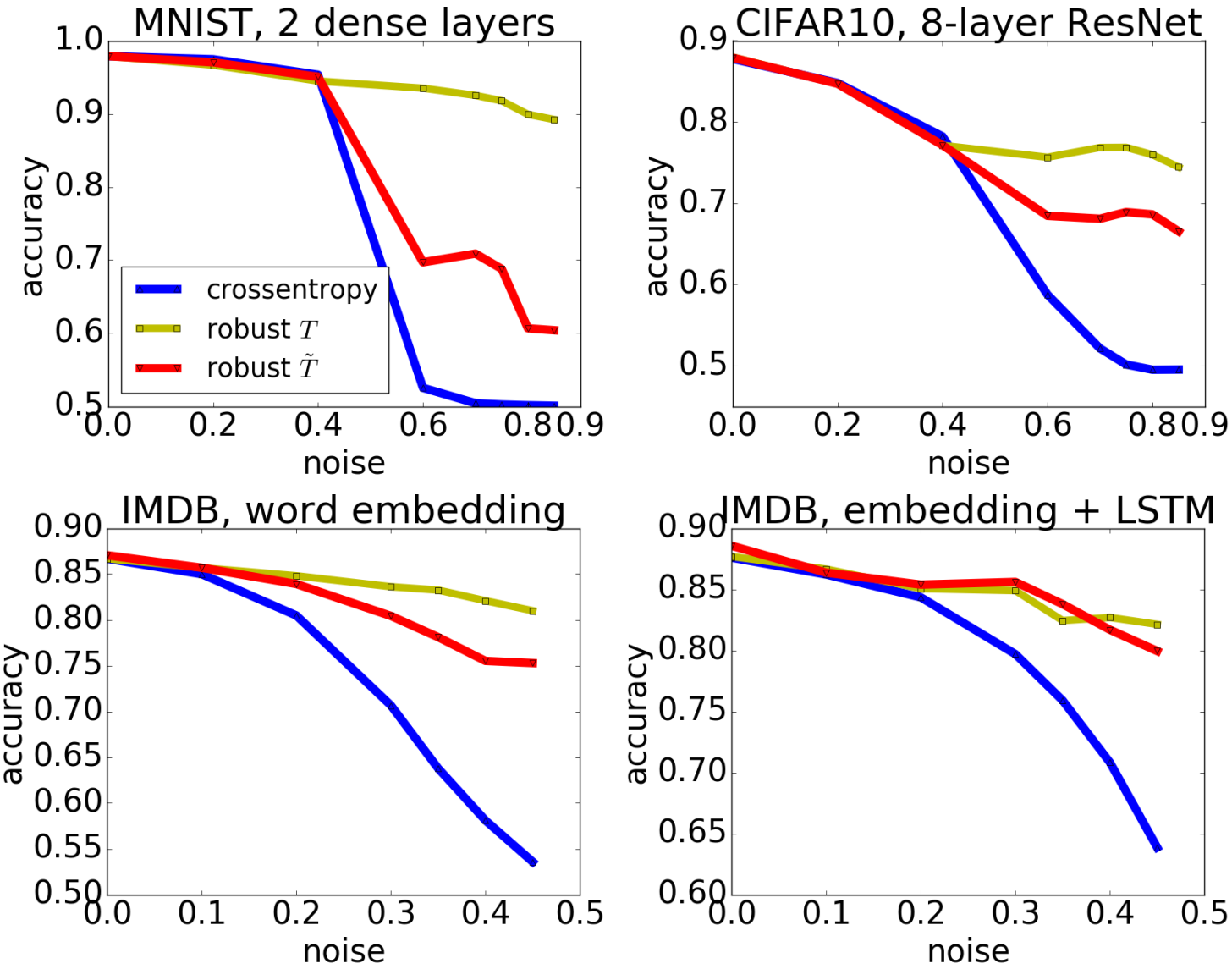
$$\underset{\boldsymbol{h}}{\operatorname{argmin}} R_{\tilde{\mathcal{S}},\ell}(\boldsymbol{h}) \rightarrow p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) \rightarrow \hat{T}$$

(2) Re-train the network correcting the sufficient statistic by $\tilde{T}^{-1}$

$$\boldsymbol{h}^* = \underset{\boldsymbol{h}}{\operatorname{argmin}} R_{\tilde{\mathcal{S}},\ell}(\boldsymbol{h}, \hat{T}^{-1})$$

no change in back-propagation

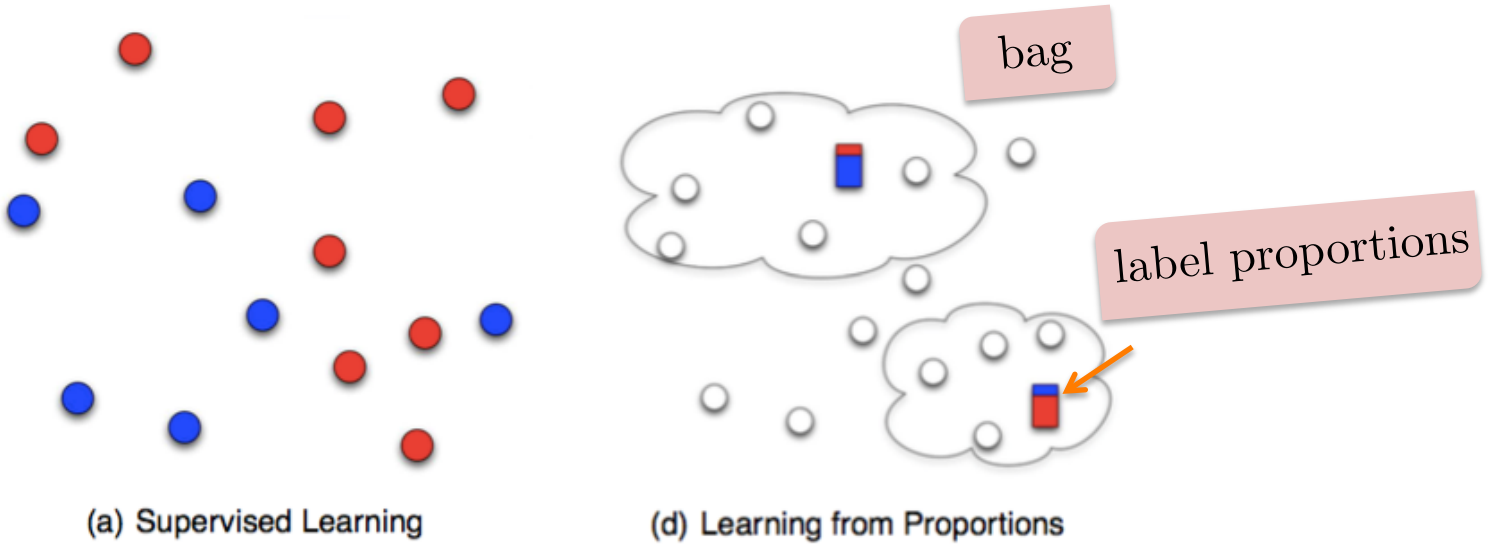
Empirics: inject sparse, asymmetric T



Outline

- Motivations
- Statistical sufficiency & loss factorization
- Asymmetric label noise I: theory with linear models
- Asymmetric label noise II: neural nets
- **Other learning settings and conclusion**

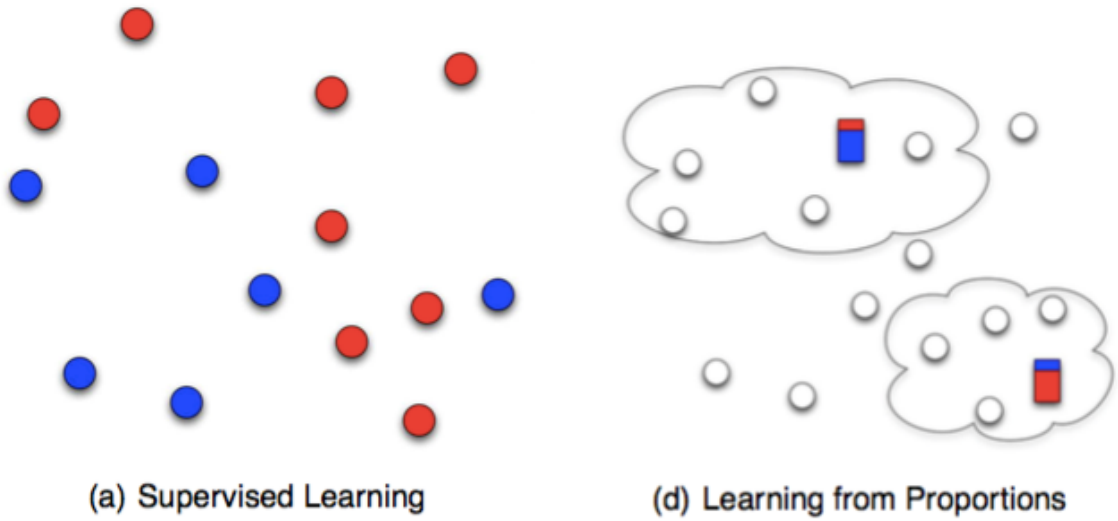
Learning from label proportions



(a) Supervised Learning

(d) Learning from Proportions

Learning from label proportions



bag centroids

label proportions

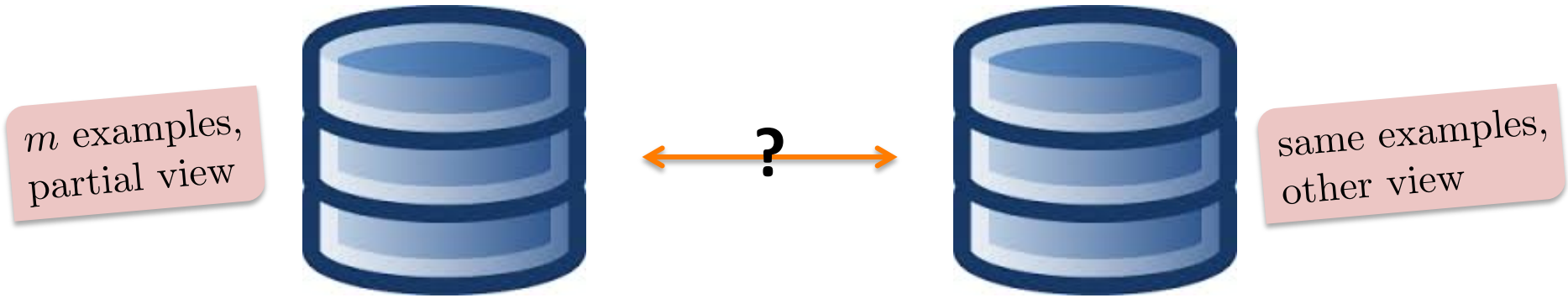
Laplacian regularizer

Step (1)

$$\hat{U} = \underset{U}{\operatorname{argmin}} \operatorname{tr}\left((B - \Pi U)^\top (B - \Pi U)\right) + \gamma \operatorname{tr}\left(U^\top L U\right)$$

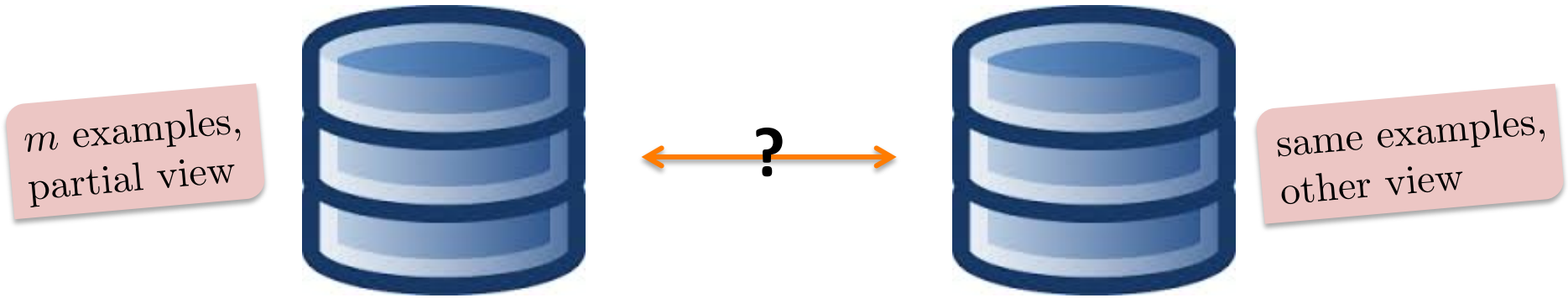
$$\hat{U} \to \hat{\boldsymbol{\mu}} \quad \text{(by linear algebra)}$$

Learning from distributed datasets



- Data is **vertically** partitioned: same examples, different features
- "who-is-who" not know: **no shared IDs**
- Goal: learn in the product feature space

Learning from distributed datasets



- Data is **vertically** partitioned: same examples, different features
- "who-is-who" not know: **no shared IDs**
- Goal: learn in the product feature space

- NO: entity matching
- YES: compute sufficient statistics, Rademacher observations

Conclusion

- Sufficiency is a powerful tool
  - decoupling & modularity
  - abstraction
  - computational saving (by compression)
- Toward ML
  - less like a bag of tricks
  - more like engineering

  (see also J. Langford's learning reductions)

References

- G. Patrini, R. Nock, P. Rivera, T. Caetano, **(Almost) no label no cry**, NIPS'14
- G. Patrini, F. Nielsen, R. Nock, M. Carioni, **Loss factorization, weakly supervised learning and label noise robustness**, ICML'16
- G. Patrini, R. Nock, S. Hardy, T. Caetano, **Fast learning from distributed datasets without entity matching**, IJCAI'16
- G. Patrini, A. Rozza, R. Nock, A. Menon, L. Qu, **Making Neural Networks Robust to Label Noise: a Loss Correction Approach**, NIPS'16 (submitted)

Co-authored

- R. Nock, G. Patrini, A. Friedman, **Rademacher observations, private data and boosting**, ICML'15

Patent applications

- R. Nock, G. Patrini, T. Caetano, **Learning with transformed data**, 2015
- R. Nock, G. Patrini, **Learning with distributed data**, 2016