A Mixture Model for Grouping Annotations in Learning from Crowds

CIARP 2019

Francisco Mena and Ricardo Ñanculef

Departamento de Informática
Universidad Técnica Federico Santa María
Chile

October 2019

Summary

Motivation

- Machine learning methods has been widely spread to different areas.
- Supervised learning relies on correctly labelled data to learn some task.
- Unfortunately, human annotators are imprecise.
    - In some cases it can be very difficult or infeasible to obtain accurate labels.
    - Subjective task: *Sentiment Analysis*, *Product Rating* or *Medical Judgment*.

Label	Dog	Cat
Ground Truth	?	?

Crowdsourcing Solution

- We can collect multiple subjective and possible innacurate labels and try to infer the *ground truth* from these annotations.

  - More feasible and cheaper through *crowdsourcing* platforms.
    - As Amazon Mechanical Turk (AMT) and CrowdFlower.
  - Annotators together generate one or more annotations per item.

What is the difficulty of *crowdsourcing*?

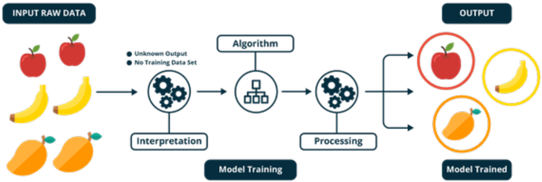
- Annotators can have varying levels of expertise and ability.
- There could be bad scenarios where the annotations obtained were generated by
  - Inaccurate
  - Spammers
  - Malicious

Base of the Problem Definition

Supervised scenario

- Consider an input pattern x, with $x \sim p(x)$ unknown, and a *ground truth* label $z \in \{1, 2, \ldots, K\}$, with $z \sim p(z|x)$ unknown.
- Given a sample $\{(x_i, z_i)\}_{i=1}^{N}$ drawn from $p(x, z) = p(z|x)p(x)$.

Objective: Learn $p(z|x)$ or its properties (mode, expected value or others).



Problem Definition I

Crowdsourcing scenario:

- Annotations y are produced by a labelling process $y \sim p(y|x,z)$, that depends on the **ability** of the annotator to detect the ground truth.

Individual scenario

- Consider T annotators and T_i annotations per item x_i.
- Given the sample $\{(x_i, \{y_i^{(\ell)}\}_{\ell=1}^{T_i})\}_{i=1}^{N}$ from $p(x,y)$.

Objective: (i) Learn $p(z|x)$ and (ii) learn the ability of each annotator.

Problem Definition II

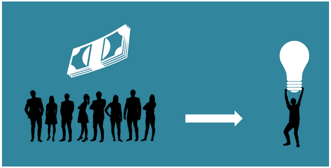
Crowdsourcing scenario:

- Annotations y are produced by a labelling process $y \sim p(y|x,z)$, that depends on the **ability** of the annotator to detect the ground truth.

Global scenario

- It is not known or care who provided the annotations.
- Given the sample $\{(x_i, r_i)\}_{i=1}^{N}$ to learn *crowdsourcing* task (i).
- $r_{ij} \in \{0, 1, \ldots, T_i\}$ is the number of annotations j given for item i.

State of the Art: Simple Aggregation



- Methods that use summary statistics to reduce annotations into a single label.
- Most used and simple technique: *Majority Voting* (MV)
  - *hard-MV*: $z_i = mode\{y_i^{(1)}, \ldots, y_i^{(T_i)}\} = \arg\max_j r_{ij}$
  - *soft-MV*: $z_{ij} = \frac{1}{T_i} \sum_{\ell}^{T_i} y_{ij}^{(\ell)} = \frac{1}{T_i} \cdot r_{ij}$
- The MV has a limited performance in some cases:
  - Quite different ability among annotators.
  - Few annotations by data.

State of the Art: **Without** Predictive Model

Setting

- **Objective**: Learn *ground truth* z_i through annotations $\{y_i^{(t)}\}_t^T$.
- *Assumptions*:
  - Input pattern is not available: $p(z|x) = p(z)$.
  - Annotator labels every data x_i: $T_i = T$ (**dense** labels).
- It needs a second step to learn a predictive model over z: $f(x)$.

- Dawid and Skene 1979 (DS) pioneer work that deal with annotators of varying expertise.
- DS models annotator ability as a confusion matrix, $p(y^{(t)}|z)$, and infers the *ground truth* with EM algorithm.
  - Zhang et al. 2016 proposed another way to initialize EM that allows to speed up the convergence.

State of the Art: **With** Predictive Model

Setting

- **Objective**: Include the predictive model into the learning process.
- Predictive model: binary problems, usually logistic regression (*LR*).
- Same *assumption* of dense labeling: $T_i = T$.

DS extension:

- Raykar et al. 2010
  - At the M step, learn the predictive model with confusion matrices.
  - At the E step, infer the ground truth to use on the M step.
- Annotator ability as a learning model (*LR*): Yan et al. 2010 and Kajino 2012 (convex).
- Annotator reliability as binary latent variable: Rodrigues et al. 2013.

State of the Art: Deep Learning

- The LR model is replaced by a deep learning (DL) model.
- Albarqouni et al. 2016 applied Raykar's model in a cancer detection, replacing LR by a CNN.
- Rodrigues et al. 2018 extended Albarqouni to multiple classes.
  - They also proposed to encode the confusion matrix into the DL model.
- Patrini et al. 2017 faced the *label noise* problem (1 annotator) with neural net assuming known the confusion matrix.

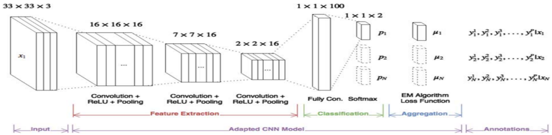


Figure 1: *AggNet* of Albarqouni et al.

Proposal

Nowadays, there is no method that is superior to the others in all the cases (Zheng et al. 2017).
Different assumptions have to be fulfilled to achieve good results.

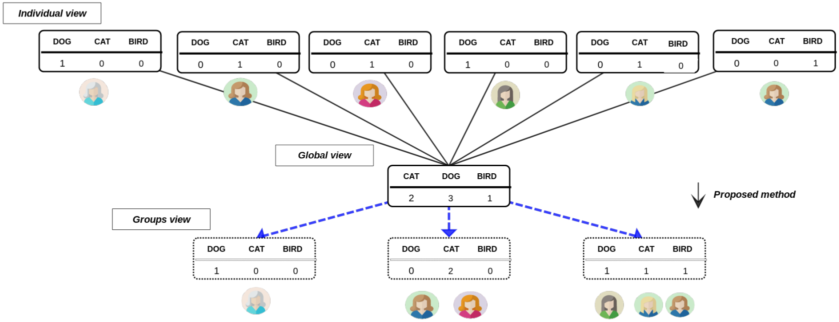
We focus on:

- Large scale scenarios in terms of annotators ($T >> 0$).
  - Avoid having an explicit model per annotator (*computational efficiency*).
- Scenarios with a small number of annotations per annotator.
  - Group similar annotations (*statistical efficiency*).

Model: Global

Setting

- **Global** scenario: we do not known who gives the annotations.
- Sparse annotations: variable number by input pattern and annotator.
- *Assumption*: there exist groups of annotations with similar ability.



Model: Mixture Model and Ground Truth

Finite Mixture Model (groups)

$$p(y \mid x) = \sum_{m=1}^{M} p(y \mid x, g = m) \cdot p(g = m) \tag{1}$$

We model the groups g with an *a priori* distribution: $p(g|x) = p(g)$.

Labeling Pattern and *Ground Truth*

$$p(y = j \mid x, g = m) = \sum_{k=1}^{K} p(y = j \mid g = m, z = k) \cdot p(z = k \mid x) \tag{2}$$

We model the observed labels using a confusion matrix by group. We also assume that the *ground truth* depends only on x.

Model: Parametrization

Crowd Mixture Model (**CMM**)

$$p(y \mid x) = \sum_{k}^{K} \sum_{m}^{M} p(y \mid g = m, z = k) \cdot p(z = k \mid x) \cdot p(g = m) \tag{3}$$

Three components to model:

Term	Model	# Parameters
$p(y\|g, z)$	Confusion matrix $\beta^{(m)}$ for group m $p(y = j\|g = m, z = k)$	$MK(K-1)$
$p(z\|x)$	DL model $f(x; \theta)$	Indep. of T
$p(g)$	Mixing coefficients $\alpha^{(m)} = p(g = m)$	$M - 1$

Model: Optimization

Observed annotations $\{y_i^{(\ell)}\}_{\ell=1}^{T_i}$ are drawn from a Multinomial distribution

$$r_i \sim Mu(T_i,\ p(y|x_i)) \quad .$$

The conditional log-likelihood is thus given by

$$\begin{aligned} \ell(\Theta) = \log p(r \mid x, \Theta) &= \log\left(C \cdot \prod_j^K p(y = j \mid x)^{r_{\cdot j}}\right), \\ &= \mathcal{C} + \sum_j^K r_{\cdot j} \log\left(\sum_{m,k} \beta_{k,j}^{(m)} \cdot f_k(x;\theta) \cdot \alpha^{(m)}\right) \qquad (4) \end{aligned}$$

The hidden variable model is optimized with the use of the Jensen inequality and the EM algorithm.

Model: Training

Using the EM algorithm to optimize the lower bound:

- **E-step**. For grouping the annotations based on ground-truth estimation:

$$q_{ij}(m,k) = \frac{1}{N_{ij}} \beta_{k,j}^{(m)} f_k(x^{(i)};\theta)\alpha_m, \text{ with } N_{ij} = \sum_{m',k'} \beta_{k',j}^{(m')} f_{k'}(x^{(i)};\theta)\alpha_{m'}.$$

- **M-step**. For the mixing coefficients and confusion matrices, we obtain

$$\alpha_m = \frac{\sum_{ij} r_j^{(i)} \cdot q_{ij}(m,\cdot)}{\sum_{ij} r_j^{(i)}}, \quad \beta_{k,j}^{(m)} = \frac{\sum_i q_{ij}(m,k) \cdot r_j^{(i)}}{\sum_{ij'} q_{ij'}(m,k) \cdot r_{j'}^{(i)}}.$$

For the DL model, the objective is to minimize:

$$J(\theta) = \sum_{i,k} -\left(\sum_j q_{ij}(\cdot,k) r_j^{(i)}\right) \cdot \log f_k(x^{(i)};\theta) = \sum_{i,k} -\bar{r}_k^{(i)} \cdot \log f_k(x^{(i)};\theta)$$

Where $q_{ij}(m,\cdot) = \sum_k q_{ij}(m,k)$ and $q_{ij}(\cdot,k) = \sum_m q_{ij}(m,k)$.

Methods and Optimization

We compare:

- *Ideal*: DL model trained with the *ground truth* (as upper bound).
- DL model trained on: *hardMV* and *softMV* (Rodrigues et al. 2013).
- *DL-DS*: DL model trained over DS inferred labels (D&S 1979).
- *DL-EM*: DL model inside the EM algorithm of *DL-DS* (Albarqouni et al. 2016/Rodrigues et al. 2018).

Training details

- Methods are trained until convergence up to a maximum of 50 iterations.
- EM algorithm initialization is done with *softMV*.
- We perform 20 runs of each experiment and average the results.

Evaluation

Evaluation metrics

- To evaluate the predictive model (test set):
  - **Accuracy** over the *ground truth*.
- To evaluate the confusion matrices estimation (train set):
  - **I-JS** (Individual Jensen-Shannon divergence): average divergence between the real and the predicted matrices of each annotator.
  - **G-JS** (Global Jensen-Shannon divergence): divergence between the real and predicted global matrices.

Simulation

Simulation process as in previous work:

1. Train a neural network model over the *ground truth*.
2. Randomly perturb the model weights with M different noise levels.
3. Create the confusion matrix (ability) of each perturbed model.
4. Create T annotators by selecting one of the M ability levels based on $p(g)$.
5. Each data point is labelled by a random subset T_i of all the annotators T.
   - In average, we obtain $\bar{T}_i$ annotations per data.
6. Each annotator provides a label based on the *ground truth* and her ability.

Datasets

- Fully synthetic data (Setup (1)):
  - Three **Gaussians**, 1000 data points each, $K = 3$
  - Set $\bar{T}_i = 5$
  - Set $M = 3$ (experts, inexperts, spammers)
  - Set $p(g) = (0.25; 0.55; 0.20)$
- Semi synthetic data (Setup (2)):
  - **CIFAR-10** dataset, 60000 real images, $K = 10$
  - Set $\bar{T}_i = 3$
  - Set $M = 4$ (experts, inexperts, highly inexpert, spammers)
  - Set $p(g) = (0.20; 0.45; 0.15; 0.20)$.
- Real data (from AMT):
  - **LabelMe**, 2688 real images, $K = 8$
  - $T = 59$ annotators
  - $\bar{T}_i = 2.6$ annotations by image in average

Results on Synthetic Data

Table 1: Test accuracy of the different methods on a simulated crowd-sourcing scenario for values of T (columns) ranging from $T = 100$ to $T = 10000$. Marker † represents that the method could not be executed due to insufficient memory (16GB available).

	Setup (1)					
Method	100	500	1500	3500	6000	10000
softMV	69.34	66.21	66.87	68.48	67.00	66.49
hardMV	79.57	82.49	80.51	81.57	74.30	79.07
DL-DS	**94.66**	93.89	**92.28**	90.00	89.69	85.13
DL-EM	93.97	**93.99**	92.18	88.27	76.47	67.01
CMM	90.53	91.07	91.66	**90.45**	**90.26**	**90.46**
Ideal	94.75					

	Setup (2)					
Method	100	500	1500	3500	6000	10000
softMV	63.35	65.90	63.59	60.07	63.21	64.20
hardMV	71.09	69.50	68.48	69.09	70.08	66.01
DL-DS	71.33	68.49	68.08	66.86	†	†
DL-EM	**81.38**	**80.42**	77.81	69.81	†	†
CMM	78.83	78.36	**79.35**	**77.92**	**78.45**	**78.96**
Ideal	83.77					

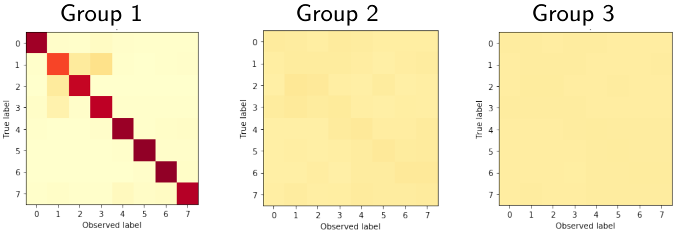
Results on Real Data

Table 2: Performance of the different methods in a real crowd-sourcing scenario (LabelMe). Marker ⋄ represents no change with respect to the setting. Acc. stands for Accuracy. Iters stands for iterations to converge.

	Individual Setting					Global Setting			
Method	Iters	Train Acc.	Test Acc.	I-JS	G-JS	Iters	Train Acc.	Test Acc.	G-JS
softMV	9.2	83.32	81.69	0.216	**0.024**		⋄	⋄	⋄
hardMV	11.8	80.34	79.95	0.225	0.035		⋄	⋄	⋄
DL-DS	10.6	84.30	**83.57**	**0.153**	0.036	4.1	12.63	14.08	0.473
DL-EM	3.9	**85.18**	83.07	0.295	0.259	3.0	78.02	75.92	0.467
CMM	7.2	84.58	83.10	0.234	0.054		⋄	⋄	⋄
Ideal	8	97.90	92.09				⋄	⋄	

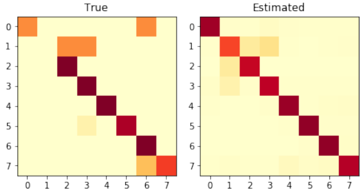
Groups found by the Method I

- Confusion matrices found on the LabelMe dataset
- $M = 3$

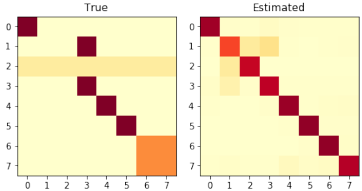


Group	$\alpha^{(m)}$	I_{sim}	$\mathbb{H}$
1	0.99	0.91	0.48
2	0.01	0.02	2.08
3	0.00	0.03	2.08

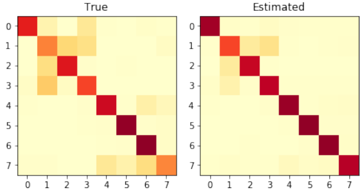
Groups found by the Method II



a) I-JS $= 0.150$, $p(g|t) = (1; 0; 0)$.



b) I-JS $= 0.232$, $p(g|t) = (1; 0; 0)$.

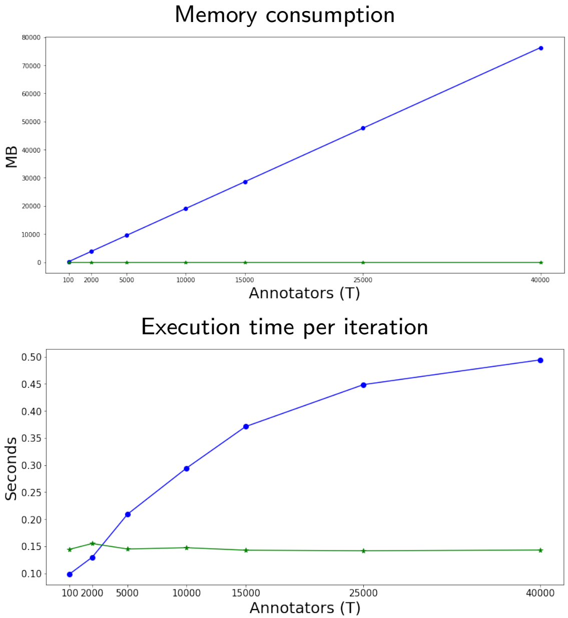


c) G-JS $= 0.047$, $p(g) = (0.99; 0.01; 0.00)$

Figure 1: Examples of confusion matrices (True vs Estimated) on the LabelMe dataset.

Computational Efficiency

- Increasing T on simulated data setup (1).
- *DL-EM*: blue
- *CMM*: green



Conclusion

- We presented a model with a fixed number of components into which annotations can be grouped together.
- Grouping annotations has have some advantages:
  - The method is more scalable in case with large number of annotators.
  - The method adapts naturally when we do not know which annotations are given by which annotator (**Global** setting).
- Our results on synthetic and real data show that *CMM* can outperform the baselines when the labels are sparse or many annotators are present.
- **Future work**: Generate an extension that avoid the use of EM algorithm.

A Mixture Model for Grouping Annotations in Learning from Crowds

CIARP 2019

Francisco Mena and Ricardo Ñanculef

Departamento de Informática
Universidad Técnica Federico Santa María
Chile

October 2019

Model: Group Assignment

- When we need to estimate the group corresponding to an annotator, we used her annotations $\mathcal{L} = \{x_i, y_i\}_{i \in N_c}$ to a group and compute:

$$p\left(g = m | \mathcal{L}, X\right) = \frac{p\left(\mathcal{L} | g = m, X\right) p(g = m | X)}{\sum_{m'} p\left(\mathcal{L} | g = m', X\right) p(g = m' | X)}$$
$$= \frac{p(y_i^{(\ell)} | x^{(i)}, g = m) \cdot \alpha^{(m)}}{\sum_{m'} p(y_i^{(\ell)} | x^{(i)}, g = m') \cdot \alpha^{(m')}}$$

- The confusion matrix of an annotator a is estimated as:

$$\beta^{(a)} = \sum_m p(g = m | a) \cdot \beta^{(m)}$$

Additional Optimization Details

EM optimization details:

- MV methods are deterministic.
- DS inference with closed equations is also deterministic.
- DL-EM and our method (CMM) are stochastic (due to neural net's optimization).
  - (M step) The DL models are trained one epoch using the Adam optimizer.
  - Multiple restarts (20) of the EM algorithm were applied.

Theoretical framework - EM

EM algorithm

Optimize iteratively the parameters Θ model over all the variables. An auxiliary model $q(\cdot)$ over the latent variable is used.

- **E**-xpectation step: Infer some latent variable c distribution, through the auxiliary model $q(\cdot)$, with Θ fixed. Initialization required.
- **M**-aximization step: Learn the model parameters Θ, maximizing a lower-bound of the log-likelihood, with $q(c)$ fixed.

Mixture Models (MM)	$p(y\|x) = \sum_k \alpha_k p_k(y\|x)$	The mixture coefficients $\{\alpha_k\}$ are hidden values
Mixture of Experts (MoE)	$p(y\|x) = \sum_i \alpha_k(x) p_k(y\|x)$	The mixture coefficients are function of some variable

Table 3: Examples of hidden variable models