Tri-Training

A Review of Zhou, Li (2005)[1]

Prepared for Metrowest Boston Machine Learning Meetup

Chris Winsor
5/15/2019

[1] Z. Zhou and M. Li, "Tri-Training: Exploiting Unlabeled Data Using Three Classifiers" in IEEE Transactions on Knowledge & Data Engineering, vol. 17, no. 11, pp. 1529-1541, 2005.

Tri-Training

What is it? Why?

"Semi-Supervised Learning" (general class)

- Prime models using Labeled Data
- Then consume unlabeled data
- Does require some labeled data (vs "unsupervised")

Motivation:

- Unlabeled data is easy to get. Labeled is expensive.
- CNNs require a LOT of data and they're popular
- We want to use our unlabeled data!

- Example:
  - Amazon wake-word - presented paper at *ICASSP 2019* 12-May-2019

Shi, Bowen & Sun, Ming & Kao, Chieh-Chi & Rozgic, Viktor & Matsoukas, Spyros & Wang, Chao. (2019). Semi-supervised Acoustic Event Detection based on tri-training

Co–Training(1)

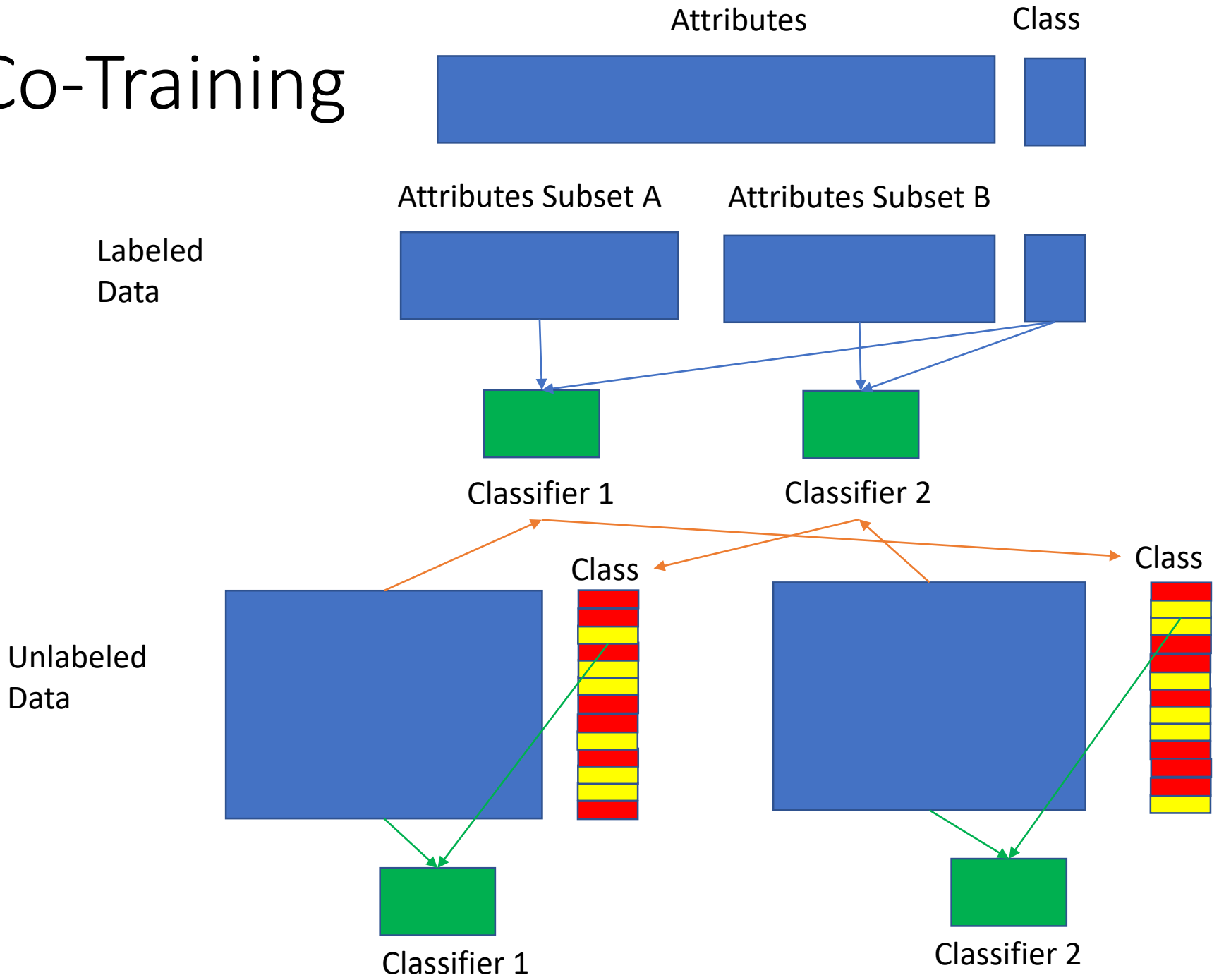
The first Semi-Supervised Algorithm

- Two classifiers
- Attributes partitioned into two sets – each sufficient, and independent of one-another (“sufficient and redundant views”)
- Classifiers trained using the labeled data and their chosen attributes
- Using unlabeled data each classifier makes prediction using their attribute set
- Each prediction is measured for confidence
- “Confident” predictions are used to re-train the peer

(1) Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann, 1998, p. 92-100.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.487.2431&rep=rep1&type=pdf

Co-Training



1) Subset the attributes

2) Train two classifiers using Labeled Data

3) Use classifiers to predict class to be used by peer

4) Measure confidence of predictions. Select data instances based on confidence.

5) Re-train using confident peer-labeled examples

Co-Training

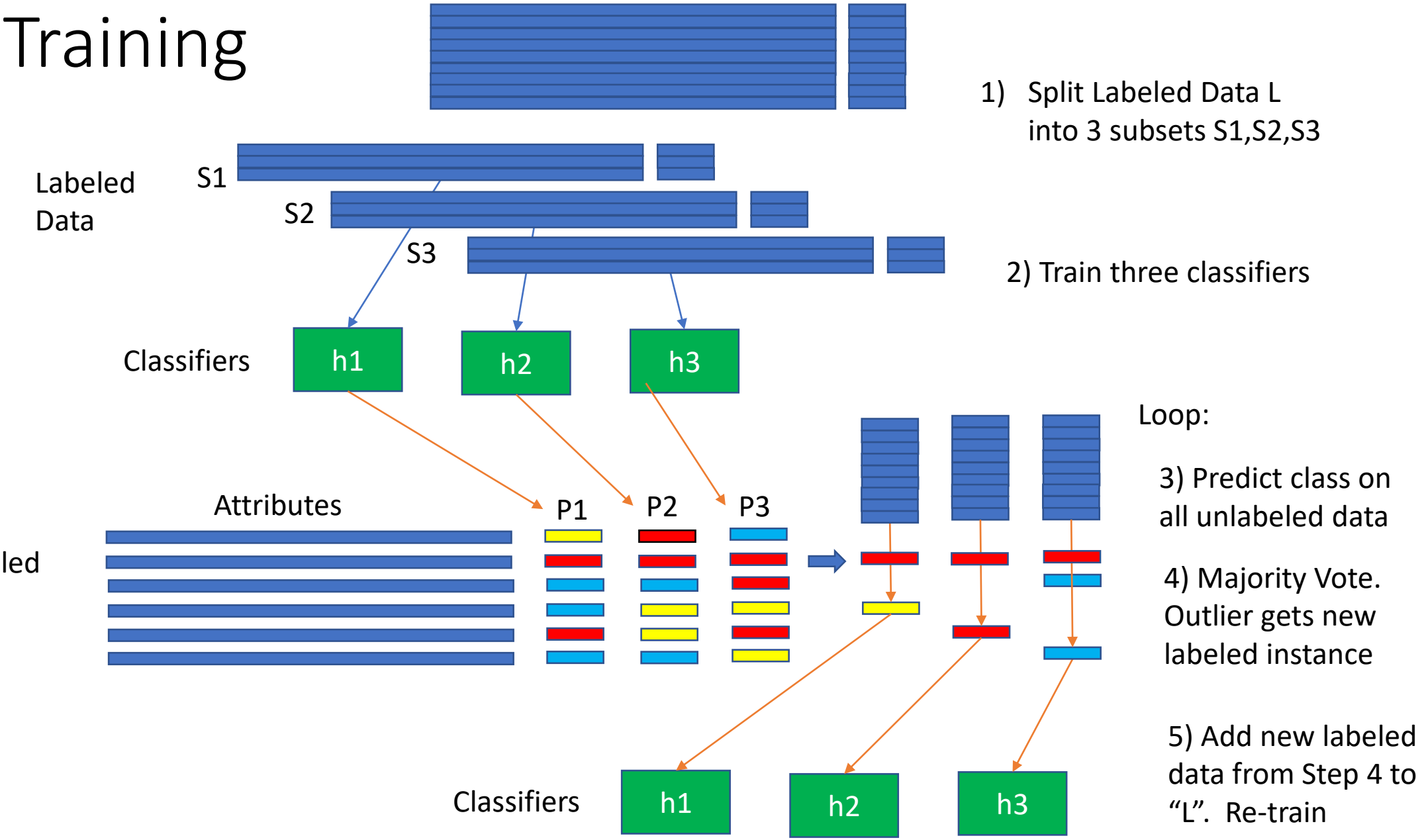
- Requires sufficient/independent attribute set
  - Not necessarily available
  - How to determine?
- Assessing confidence
  - Can be computationally expensive

Tri-Training

- Three classifiers
- Split labeled data into 3 subsets (instance-based)
- Train each using its labeled subset
- Predict on unlabeled data → 3 sets of predictions
- Majority vote. A 2/3 agreement results in new labeled instance for the "wrong" classifier
- Re-train using new labeled data

- Advantage:
  - Does not require attributes with "sufficient and redundant views"
  - No need to measure "confidence" of prediction

Tri-Training



1) Split Labeled Data L into 3 subsets S1,S2,S3

2) Train three classifiers

Loop:

3) Predict class on all unlabeled data

4) Majority Vote. Outlier gets new labeled instance

5) Add new labeled data from Step 4 to "L". Re-train

The bigger picture…

Tri-Training is NOT a machine learning algorithm.

- It is not a peer to Bayes, CNN, Decision Trees

Rather:

- Tri-Training re-purposes ML algorithms to allow use of unlabeled data!

Angluin, Laird (1988)

- How can a learning algorithm be adapted to handle noisy data?
- Findings:
  - If “teacher” (i.e. data) has independent random errors
  - Then a strategy that will work is to select the most consistent rule for the sample
- Notes:
  - It usually requires a feasibly small number of samples if the noise affects < ½ the samples
  - In general it is intractable, but shown to be polynomial for one rule type

Machine Learning 2: 343-370, 1988

Learning From Noisy Examples

DANA ANGLUIN (ANGLUIN@YALE.EDU)
Department of Computer Science, Yale University, P.O. Box 2158, New Haven, CT 06520, U.S.A.

PHILIP LAIRD (LAIRD%PLU@IO.ARC.NASA.GOV)
NASA Ames Research Center, MS 244-17, Moffett Field, CA 94035, U.S.A.

(Received: June 1, 1987)
(Revised: November 20, 1987)

Keywords: Concept learning, learning from examples, probably approximately correct learning, noisy data, theoretical limitations

Abstract. The basic question addressed in this paper is: how can a learning algorithm cope with incorrect training examples? Specifically, how can algorithms that produce an “approximately correct” identification with “high probability” for reliable data be adapted to handle noisy data? We show that when the teacher may make independent random errors in classifying the example data, the strategy of selecting the most consistent rule for the sample is sufficient, and usually requires a feasibly small number of examples, provided noise affects less than half the examples on average. In this setting we are able to estimate the rate of noise using only the knowledge that the rate is less than one half. The basic ideas extend to other types of random noise as well. We also show that the search problem associated with this strategy is intractable in general. However, for particular classes of rules the target rule may be efficiently identified if we use techniques specific to that class. For an important class of formulas – the k-CNF formulas studied by Valiant – we present a polynomial-time algorithm that identifies concepts in this form when the rate of classification errors is less than one half.

The basis for Semi-supervised Learning…

[1] Z. Zhou and M. Li, "Tri-Training: Exploiting Unlabeled Data Using Three Classifiers" in IEEE Transactions on Knowledge & Data Engineering, vol. 17, no. 11, pp. 1529-1541, 2005.

"A new co-training style semi-supervised learning algorithm is named tri-training is proposed."

[1] D. Angluin and P. Laird, "Learning from noisy examples," Machine Learning, vol.2, no.4, pp.343–370, 1988.

Abstract. The basic question addressed in this paper is: how can a learning algorithm cope with incorrect training examples? Specifically, how can algorithms that produce an "approximately correct" identification with "high probability" for reliable data be adapted to handle noisy data?

- PAC Learning -

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Valiant, L. G. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560–566). Los Angeles, CA: Morgan Kaufmann.

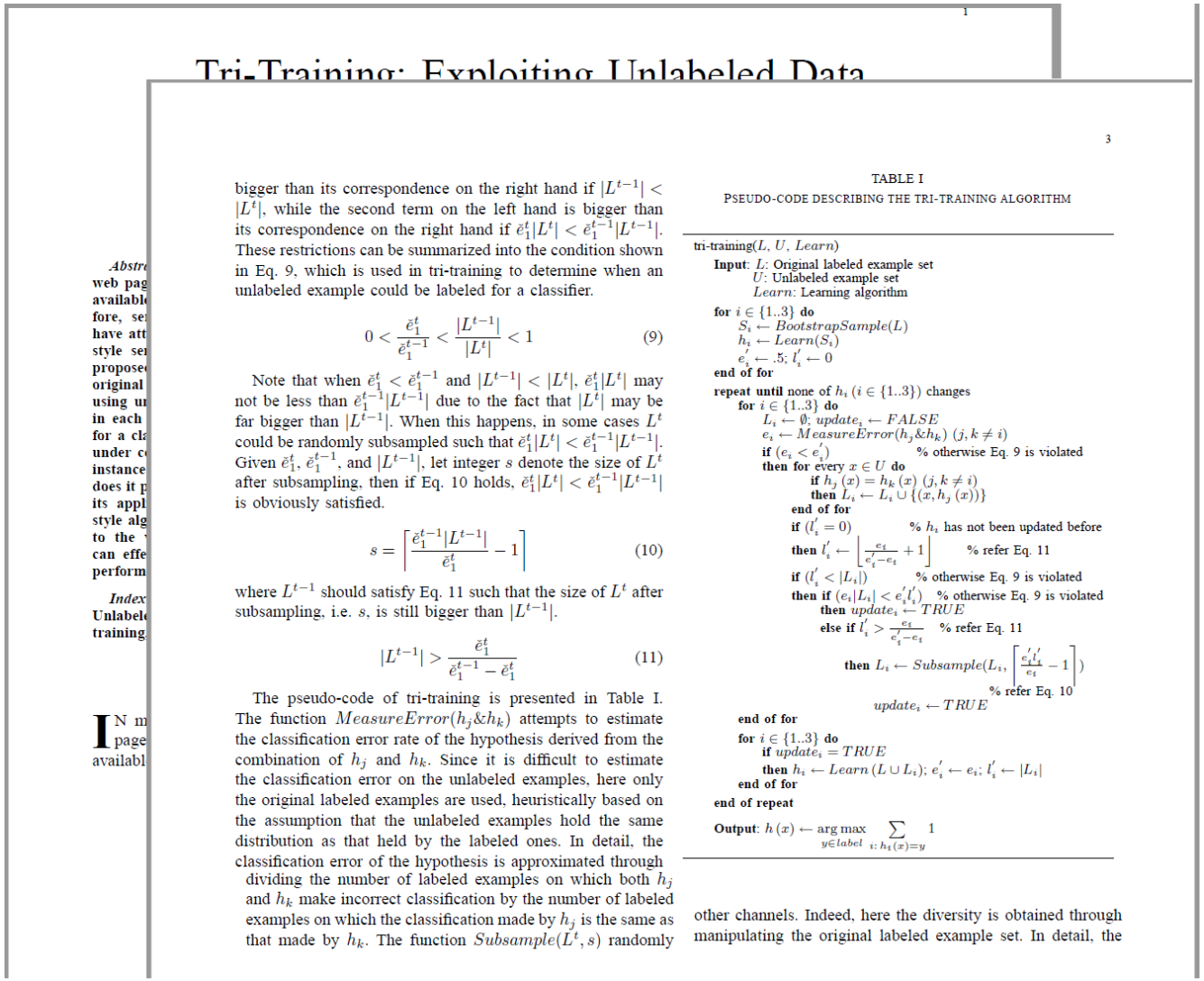
"We give a methodology for studying [learning. It consists of an] information gathering mechanism, the learning protocol, […] and the class of concepts that can be learned"

4. Robustness

How much tolerance to erroneous data should we require of a learning system? **[...]**

In this section we will modify Algorithm E and show that the modified form Algorithm E*, is resistant to a quantifiable, if low, error rate, **[...]**

Equations 1.11 and Code Walk-through



1

Tri-Training: Exploiting Unlabeled Data

3

bigger than its correspondence on the right hand if $|L^{t-1}| < |L^t|$, while the second term on the left hand is bigger than its correspondence on the right hand if $\breve{e}_1^t|L^t| < \breve{e}_1^{t-1}|L^{t-1}|$. These restrictions can be summarized into the condition shown in Eq. 9, which is used in tri-training to determine when an unlabeled example could be labeled for a classifier.

$$0 < \frac{\breve{e}_1^t}{\breve{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|} < 1 \tag{9}$$

Note that when $\breve{e}_1^t < \breve{e}_1^{t-1}$ and $|L^{t-1}| < |L^t|$, $\breve{e}_1^t|L^t|$ may not be less than $\breve{e}_1^{t-1}|L^{t-1}|$ due to the fact that $|L^t|$ may be far bigger than $|L^{t-1}|$. When this happens, in some cases L^t could be randomly subsampled such that $\breve{e}_1^t|L^t| < \breve{e}_1^{t-1}|L^{t-1}|$. Given $\breve{e}_1^t$, $\breve{e}_1^{t-1}$, and $|L^{t-1}|$, let integer s denote the size of L^t after subsampling, then if Eq. 10 holds, $\breve{e}_1^t|L^t| < \breve{e}_1^{t-1}|L^{t-1}|$ is obviously satisfied.

$$s = \left\lceil \frac{\breve{e}_1^{t-1}|L^{t-1}|}{\breve{e}_1^t} - 1 \right\rceil \tag{10}$$

where L^{t-1} should satisfy Eq. 11 such that the size of L^t after subsampling, i.e. s, is still bigger than $|L^{t-1}|$.

$$|L^{t-1}| > \frac{\breve{e}_1^t}{\breve{e}_1^{t-1} - \breve{e}_1^t} \tag{11}$$

The pseudo-code of tri-training is presented in Table I. The function $MeasureError(h_j\&h_k)$ attempts to estimate the classification error rate of the hypothesis derived from the combination of h_j and h_k. Since it is difficult to estimate the classification error on the unlabeled examples, here only the original labeled examples are used, heuristically based on the assumption that the unlabeled examples hold the same distribution as that held by the labeled ones. In detail, the classification error of the hypothesis is approximated through dividing the number of labeled examples on which both h_j and h_k make incorrect classification by the number of labeled examples on which the classification made by h_j is the same as that made by h_k. The function $Subsample(L^t, s)$ randomly

TABLE I
PSEUDO-CODE DESCRIBING THE TRI-TRAINING ALGORITHM

tri-training(L, U, $Learn$)
Input: L: Original labeled example set
U: Unlabeled example set
$Learn$: Learning algorithm
for $i \in \{1..3\}$ **do**
$S_i \leftarrow BootstrapSample(L)$
$h_i \leftarrow Learn(S_i)$
$e_i' \leftarrow .5$; $l_i' \leftarrow 0$
end of for
repeat until none of h_i $(i \in \{1..3\})$ changes
for $i \in \{1..3\}$ **do**
$L_i \leftarrow \emptyset$; $update_i \leftarrow FALSE$
$e_i \leftarrow MeasureError(h_j\&h_k)$ $(j, k \neq i)$
if $(e_i < e_i')$ % otherwise Eq. 9 is violated
then for every $x \in U$ **do**
if $h_j(x) = h_k(x)$ $(j, k \neq i)$
then $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
end of for
if $(l_i' = 0)$ % h_i has not been updated before
then $l_i' \leftarrow \left\lfloor \frac{e_i}{e_i' - e_i} + 1 \right\rfloor$ % refer Eq. 11
if $(l_i' < |L_i|)$ % otherwise Eq. 9 is violated
then if $(e_i|L_i| < e_i' l_i')$ % otherwise Eq. 9 is violated
then $update_i \leftarrow TRUE$
else if $l_i' > \frac{e_i}{e_i' - e_i}$ % refer Eq. 11
then $L_i \leftarrow Subsample(L_i, \left\lceil \frac{e_i' l_i'}{e_i} - 1 \right\rceil)$
% refer Eq. 10
$update_i \leftarrow TRUE$
end of for
for $i \in \{1..3\}$ **do**
if $update_i = TRUE$
then $h_i \leftarrow Learn(L \cup L_i)$; $e_i' \leftarrow e_i$; $l_i' \leftarrow |L_i|$
end of for
end of repeat
Output: $h(x) \leftarrow \arg\max_{y \in label} \sum_{i;\, h_i(x) = y} 1$

other channels. Indeed, here the diversity is obtained through manipulating the original labeled example set. In detail, the

When can newly labeled examples be used?

These restrictions can be summarized into the condition shown in Eq. 9, which is used in tri-training to determine when an unlabeled example could be labeled for a classifier.

$$0 < \frac{\check{e}_1^t}{\check{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|} < 1 \qquad (9)$$

e(t) / e(t-1) < 1 **"The error rate must be decreasing"**

|L(t-1)| / |L(t)| < 1 **"The number of labeled samples must be increasing"**

|L(t)| * e(t) < |L(t-1)| * e(t-1) **"The number of samples time the error rate must be decreasing"**

Tri-Training Pseudo-Code (priming section)

TABLE I

PSEUDO-CODE DESCRIBING THE TRI-TRAINING ALGORITHM

---

tri-training(L, U, $Learn$)

Input: L: Original labeled example set
U: Unlabeled example set
$Learn$: Learning algorithm

for $i \in \{1..3\}$ **do**
$S_i \leftarrow BootstrapSample(L)$
$h_i \leftarrow Learn(S_i)$
$e_i' \leftarrow .5$; $l_i' \leftarrow 0$
end of for

repeat until none of h_i $(i \in \{1..3\})$ changes ...

"last loop" error
"last loop" # of algorithm-labeled samples

Tri-Training Pseudo-Code (updating section)

repeat until none of h_i $(i \in \{1..3\})$ changes
  for $i \in \{1..3\}$ **do**
    $L_i \leftarrow \emptyset$; $update_i \leftarrow FALSE$
    $e_i \leftarrow MeasureError(h_j \& h_k)$ $(j, k \neq i)$
    if $(e_i < e_i^{'})$ % otherwise Eq. 9 is violated
    then **for** every $x \in U$ **do**
      if $h_j(x) = h_k(x)$ $(j, k \neq i)$
      then $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
    end of for
    if $(l_i^{'} = 0)$ % h_i has not been updated before
    then $l_i^{'} \leftarrow \left\lfloor \frac{e_i}{e_i^{'} - e_i} + 1 \right\rfloor$ % refer Eq. 11
    if $(l_i^{'} < |L_i|)$ % otherwise Eq. 9 is violated
    then if $(e_i |L_i| < e_i^{'} l_i^{'})$ % otherwise Eq. 9 is violated
      then $update_i \leftarrow TRUE$
      else if $l_i^{'} > \frac{e_i}{e_i^{'} - e_i}$ % refer Eq. 11
        then $L_i \leftarrow Subsample(L_i, \left\lceil \frac{e_i^{'} l_i^{'}}{e_i} - 1 \right\rceil)$
        % refer Eq. 10
        $update_i \leftarrow TRUE$
  end of for
  for $i \in \{1..3\}$ **do**
    if $update_i = TRUE$
    then $h_i \leftarrow Learn(L \cup L_i)$; $e_i^{'} \leftarrow e_i$; $l_i^{'} \leftarrow |L_i|$
  end of for
end of repeat

For each hypothesis...

Initialization:
Clear the algorithm-labeled set of samples
"Estimate" the classification error for this classifier using labeled data and the other two classifiers.

Equation 9 determines when an unlabeled example can be used
Straightaway we check the first term of Eq. 9. If the overall error is not decreasing then we cannot use any of the samples. $\frac{\check{e}_1^t}{\check{e}_1^{t-1}} < 1$ from (9)

For all the unlabeled data
Perform prediction using other 2 classifiers.
If they agree then add the sample with agreed-to label to data for this classifier.

(??) One-time (first pass only). Create a L' using equation 11

Equation 9 again...
If the count of labeled examples is not increasing then we cannot use any of the samples. $\frac{|L^{t-1}|}{|L^t|} < 1$ from (9)

Equation 9... checking: $\frac{\check{e}_1^t}{\check{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|}$ from (9)

If there are too many samples (L(t) is large) then DECREASE the number of samples so as to make the second term larger than the first term. Do this by sub-sampling.

If there was any unlabeled data that could be used then re-train the classifier using the labeled + the newly labeled data.

Update the e'i (error threshold) and l'i (number of unlabeled samples used)

Testing and Rationale

Datasets <applicability>

- (12) UCI, (1) Web Page Classification

“Unlabel Rate” <scalability>

- 20% to 80%

- ML Algorithm <generality>
  - Decision Tree, NN, Naïve Bayes
- Compare to other Semi-Supervised <compare to peers>
  - Co-training, Self-training1, Self-training2

Results

(avg on UCI datasets)

		tri-training			co-training			self-training1			self-training2		
		initial	final	improv	initial	final	improv	initial	final	improv	initial	final	improv
80% UNLABEL RATE													
J4.8 decision tree	ave.	.180	.159	**15.1%**	.199	.189	6.9%	.180	.169	7.8%	.171	.173	-3.1%
BP neural networks	ave.	.154	.136	**16.4%**	.173	.162	9.1%	.154	.141	15.3%	.145	.142	3.3%
Naive Bayes	ave.	.201	.187	**8.0%**	.188	.184	2.4%	.201	.188	7.6%	.193	.194	-1.3%
60% UNLABEL RATE													
J4.8 decision tree	ave.	.153	.138	**13.1%**	.179	.169	7.9%	.153	.148	8.2%	.155	.156	2.0%
BP neural networks	ave.	.136	.122	**14.0%**	.170	.158	8.2%	.136	.127	8.1%	.133	.139	-9.5%
Naive Bayes	ave.	.200	.189	**5.5%**	.190	.185	2.3%	.200	.190	4.4%	.196	.196	1.0%
40% UNLABEL RATE													
J4.8 decision tree	ave.	.149	.135	**13.3%**	.165	.158	5.8%	.149	.139	11.5%	.148	.152	-2.0%
BP neural networks	ave.	.128	.117	**12.0%**	.164	.152	9.8%	.128	.121	6.7%	.129	.133	-4.9%
Naive Bayes	ave.	.190	.180	**6.2%**	.185	.181	1.5%	.190	.180	5.7%	.194	.196	-0.6%
20% UNLABEL RATE													
J4.8 decision tree	ave.	.147	.128	**15.5%**	.171	.163	7.0%	.147	.139	8.2%	.143	.141	-1.8%
BP neural networks	ave.	.127	.111	**17.1%**	.144	.136	7.5%	.127	.120	7.1%	.128	.126	1.8%
Naive Bayes	ave.	.190	.179	**6.6%**	.182	.179	1.8%	.190	.181	5.4%	.191	.193	-0.6%

Conclusions

Good efficiency and generalizability because:

- “gracefully”(?) choose examples to label
- Use multiple classifiers to compose final hypothesis

Wide applicability

- Does not require sufficient and redundant views
- No constraint on learning algorithm

- Algorithm is simple, but effectively exploited unlabeled examples

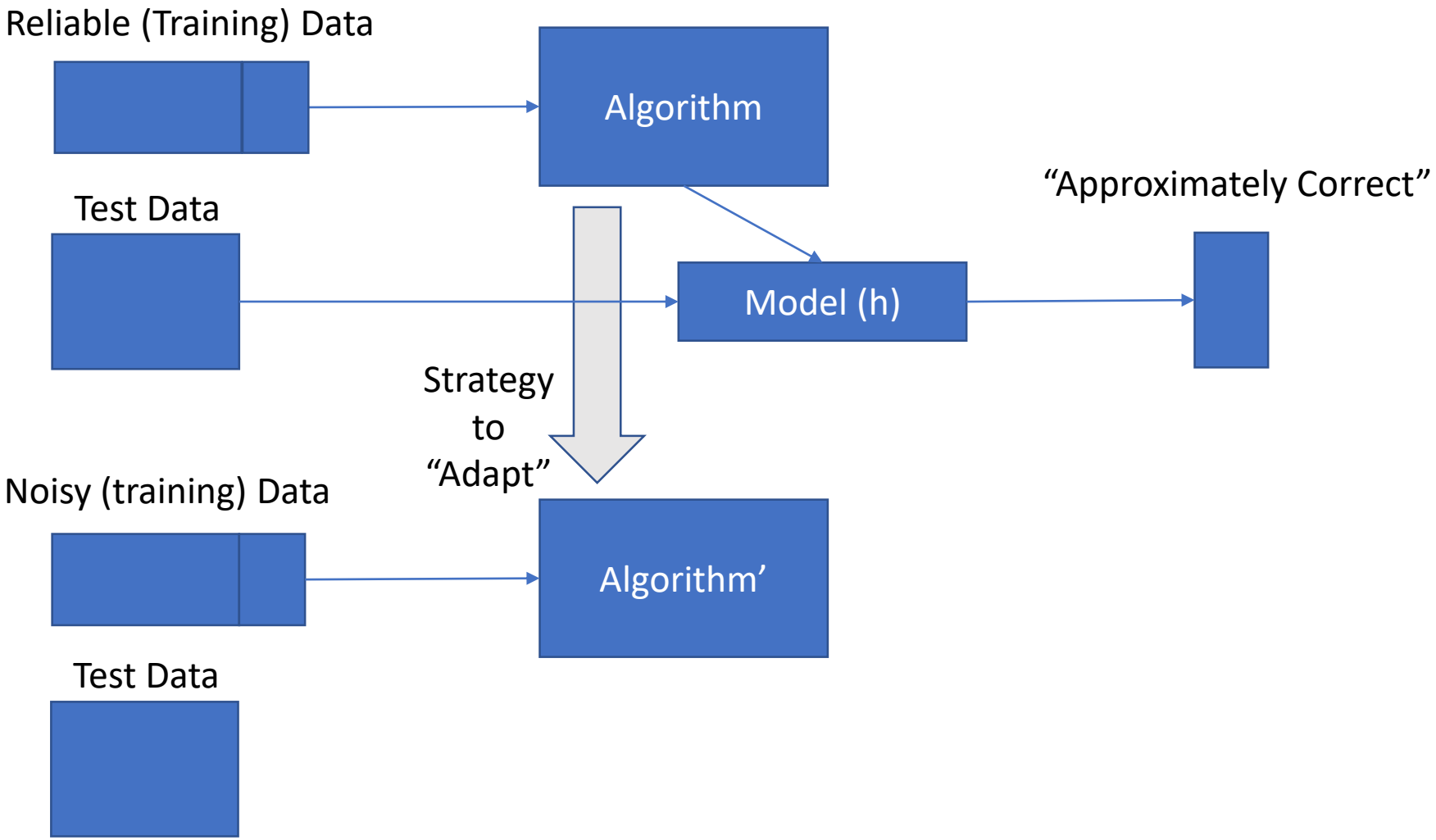
Future Research

- Data Editing [18]
  - Help identify wrongly-labeled examples
  - Incorporating Data Editing into Tri-training
- Active Learning [7]
  - Labels for selected unlabeled samples asked from user
  - Query by Committee [25]
- Current implementation “incremental learning”
  - instead of full re-train each loop

Thank You

Angluin, Laird (1988)

How to cope with incorrect training examples?



PAC (probably approximately correct) Learning
(tutorial references…)

- https://en.wikipedia.org/wiki/Probably_approximately_correct_learning
- https://jeremykun.com/2014/01/02/probably-approximately-correct-a-formal-theory-of-learning/