Understanding the Yarowsky Algorithm

Steven Abney
University of Michigan

Bootstrapping

- Co-training
  - well understood
  - view independence
- Yarowsky algorithm
  - Suggestion: precision independence $p(j|f, \text{unlabeled}) = p(j|f, \text{labeled})$
  - Precision: density of label j
  - But: not well supported in the data

Different Approach

- No independence assumption
- Optimization of objective function
  - H (negative of likelihood)
  - K (upper bound on H)
- Variants of Yarowsky algorithm
  - Y-1/DL-EM (L, LU)
  - Y-1/DL-1 (R, VS)
  - YS (P, R, FS)

Generic Yarowsky Algorithm Y-0

- Given: labeled examples Λ_0, unlabeled examples V_0
  - Y_j: set of examples labeled j
- Train classifier $\rightarrow \pi_x(j)$ prediction distribution
  - Yarowsky: $[\![j = j*]\!]$
- Label examples
  - Set $Y(x) = \hat{y}$ if $\pi_x(\hat{y}) > \zeta$
  - where $\hat{y}$ is most-probable label $\arg\max_j \pi_x(j)$
  - and ζ is labeling threshold
- Stop if no change

Decision List Induction

- Rules $f \rightarrow j$ with weight θ_{fj}
  - We assume $0 \leq \theta_{fj} \leq 1$ and $\sum_j \theta_{fj} = 1$
- Prediction distribution
  - Point distribution $\pi_x(j) = [\![j = j*]\!]$
  - Mixture distribution $\pi_x(j) = \frac{1}{m} \sum_{f \in F_x} \theta_{fj}$
- Update rule
  - Raw precision: $\theta_{fj} = q_f(j)$
  - Fixed smoothing: $\theta_{fj} = \tilde{q}_f(j; \epsilon) \quad \epsilon = 0.1$
  - Other update rules: variable smoothing, peaked, EM, EM + variable smoothing
  - Update threshold: change θ_f only if precision $> \eta$

Differences from Original Yarowsky Algorithm

- Prediction distribution: use mixture distribution, not point distribution
- Labeling
  - Minimal labeling threshold: $\zeta = \frac{1}{L}$
  - No "unlabeling" – once labeled always labeled, though label may change
- No update threshold
- Original algorithm: parallel update (all θ_{fj})
  - We also consider sequential update (single best f)
- Original algorithm: smoothed precision as update
  - We consider a variety of update rules

Objective Function

- Maximize log likelihood

$$\begin{aligned} l &= \log \prod_x \pi_x(Y(x)) \\ &= \sum_x \log \pi_x(Y(x)) \\ &= \sum_x \sum_j [\![j = Y(x)]\!] \log \pi_x(j) \\ &= \sum_x \phi_{xj} \log \pi_x(j) \\ &= -\sum_x H(\phi_x \| \pi_x) \end{aligned}$$

- Minimize cross entropy

$$H = \sum_x H(\phi_x \| \pi_x)$$

Extension to Unlabeled Data

- Labeling distribution ϕ, prediction distribution π

$$\phi_x(j) = \begin{cases} [\![j = Y(x)]\!] & \text{if } x \text{ is labeled} \\ \frac{1}{L} & \text{otherwise} \end{cases}$$

$$\phi_x(j) = [\![x \in Y_j]\!] + [\![x \in \mathrm{V}]\!] \frac{1}{L}$$

To Minimize H

$$H = \sum_x H(\phi_x \| \pi_x) = \sum_x H(\phi_x) + \sum_x D(\phi_x \| \pi_x)$$

- Assign labels to unlabeled examples: $\sum_x H(\phi_x) \to 0$
- Make prediction dist agree with label dist: $\sum_x D(\phi_x \| \pi_x) \to 0$
- Equal to maximum likelihood if all examples are labeled

Modified Generic Yarowsky Algorithm Y-1

- Given: labeled examples Λ_0, unlabeled examples V_0
- Train classifier $\longrightarrow \pi_x(j)$
- Label examples
  - Set $Y(x) = \hat{y}$ if $\boxed{\text{previously labeled or } \pi_x(\hat{y}) > 1/L}$
- Stop if no change
- "Generic" – does not specify *base learning algorithm*

Theorem 1

If the base learner reduces

$$D = \sum_{x} D(\phi_x \| \pi_x)$$

or

$$D_\Lambda = \sum_{x \in \Lambda} D(\phi_x \| \pi_x)$$

then Y-1 converges to a local minimum of H.

Proof Sketch

- Training step
  - Hold ϕ constant, change π
  - Case 1: base learner reduces D, hence H
- Labeling step
  - Hold π constant, change ϕ

$$H(p\|\pi_x) = \sum_j p_j \log \frac{1}{\pi_x(j)}$$

  - Reduce H by placing all mass in j that minimizes the log

$$\begin{aligned}\arg\min_j \log \frac{1}{\pi_x(j)} &= \arg\max_j \pi_x(j) \\ &= \hat{y}\end{aligned}$$

Case 2

- If base learner reduces D_Λ

$$H = \sum_x H(\phi_x) + \sum_{x \in \Lambda} D(\phi_x \| \pi_x) + \sum_{x \in \mathrm{V}} D(\phi_x \| \pi_x)$$

- Third term may increase
  - but only if new $\pi_x \neq u$
  - hence x was unlabeled, becomes labeled

$$\begin{aligned} H_0 &= \textstyle\sum_j \phi_{xj}^{\mathrm{old}} \log \frac{1}{\pi^{\mathrm{old}}} = \sum_j u(j) \log \frac{1}{u(j)} = H(u) \\ H_1 &= \textstyle\sum_j \phi_{xj}^{\mathrm{old}} \log \frac{1}{\pi^{\mathrm{new}}} \\ H_2 &= \textstyle\sum_j \phi_{xj}^{\mathrm{new}} \log \frac{1}{\pi^{\mathrm{new}}} = \log \frac{1}{\pi^{\mathrm{new}}(\hat{y})} < H(u) \end{aligned}$$

$$\Delta H = H_2 - H_1 + H_1 - H_0 < 0$$

Base Learner

- Yarowsky decision list learner does not maximize likelihood
- A learner that does: DL-EM

$$\pi(f|x) = 1/m$$

$$\pi(j|f) = \theta_{fj}$$

$$\pi(f,j|x) = \frac{1}{m}\theta_{fj}$$

$$\pi(j|x) = \sum_{g \in F_x} \frac{1}{m}\theta_{gj}$$

$$\pi(f|x,j) = \frac{1}{\pi(j|x)}\left(\frac{1}{m}\theta_{fj}\right)$$

$$\theta_{fj}^{\text{new}} = \frac{1}{Z}\sum_{x \in Y_j} \pi(f|x,j)$$

Theorem 2

DL-EM decreases D_Λ

- Corollary

  Algorithm Y-1 with DL-EM as base learner converges to a local minimum of H (a local maximum of likelihood)

Proof Sketch

- Reduction in D_Λ can be expressed as:

$$\text{gain} = -\Delta D_\Lambda = \log \pi^{\text{new}}(j|x) - \log \pi^{\text{old}}(j|x)$$

- EM algorithm is based on nonnegativity of divergence:

$$0 \leq D(\pi^{\text{old}}_{xj} \| \pi^{\text{new}}_{xj}) = \text{gain} - \mathsf{E}_f \left[\log \theta^{\text{new}}_{fj} - \log \theta^{\text{old}}_{fj}\right]$$

- That is:

$$\text{gain} \geq \mathsf{E}_f \left[\log \theta^{\text{new}}_{fj} - \log \theta^{\text{old}}_{fj}\right]$$

- Take expectation over j and x, and maximize $\mathsf{E}_f \log \theta^{\text{new}}_{fj}$ under the constraint that θ_f sums to unity. Result is the DL-EM update:

$$\theta^{\text{new}}_{fj} = \frac{1}{Z} \sum_{x \in Y_j} \pi^{\text{old}}(f|x,j)$$

Detail

$$
\begin{aligned}
& D(\pi_{xj}^{\mathrm{old}} \| \pi_{xj}^{\mathrm{new}}) \\
&= \sum_f \pi_{xj}^{\mathrm{old}}(f) \log \frac{\pi_{xj}^{\mathrm{old}}(f)}{\pi_{xj}^{\mathrm{new}}(f)} \\
&= \sum_f \pi_{xj}^{\mathrm{old}}(f) \log \left(\frac{\frac{1}{m}\theta_{fj}^{\mathrm{old}}}{\pi_x^{\mathrm{old}}(j)} \cdot \frac{\pi_x^{\mathrm{new}}(j)}{\frac{1}{m}\theta_{fj}^{\mathrm{new}}} \right) \\
&= \log \pi_x^{\mathrm{new}}(j) - \log \pi_x^{\mathrm{old}}(j) - \mathsf{E}_f \left[\log \theta_{fj}^{\mathrm{new}} - \log \theta_{fj}^{\mathrm{old}} \right] \\
&= \mathrm{gain} - \mathsf{E}_f \left[\log \theta_{fj}^{\mathrm{new}} - \log \theta_{fj}^{\mathrm{old}} \right]
\end{aligned}
$$

Maximizing D Instead of D_Λ

- Structure is the same. Resulting update:

$$\theta_{fj}^{\text{new}} = \frac{1}{Z}\left[\sum_{x \in Y_j} \pi_{xj}^{\text{old}}(f) + \frac{1}{L}\sum_{x \in \mathrm{V}} \pi_{xj}^{\text{old}}(f)\right]$$

- Yarowsky variants
  - Y-1/DL-EM (L, LU)
  - Y-1/DL-1 (R, VS)
  - YS (P, R, FS)

Objective Function K

- Upper bounding H

$$
\begin{aligned}
H &= -\sum_x \sum_j \phi_{xj} \log \sum_{g \in F_x} \frac{1}{m} \theta_{gj} \\
&\leq -\sum_x \sum_j \phi_{xj} \sum_{g \in F_x} \frac{1}{m} \log \theta_{gj} \\
&= \frac{1}{m} \sum_x \sum_{g \in F_x} H(\phi_x \| \theta_g)
\end{aligned}
$$

- Minimize K to minimize upper bound on H:

$$
K = \sum_x \sum_{g \in F_x} H(\phi_x \| \theta_g)
$$

Rationale

- Squeeze H between K and 0
- K is in principle reducible to 0

$$K = \sum_x \sum_{g \in F_x} [H(\phi_x) + D(\phi_x \| \theta_g)]$$

  - Label all examples: $H(\phi_x) \to 0$
  - Each feature perfectly predicts label: $D(\phi_x \| \theta_g) \to 0$
  - Initial labeling must cooperate to permit perfect prediction

Decision List Induction DL-0, DL-1

- DL-0: base learner used by Yarowsky
  - If $\tilde{q}_f(j;\epsilon) > 0.95$ for some j
  - Set $\theta_{fj} = \tilde{q}_f(j;\epsilon)$
  - Where $\epsilon = 0.1$
  - Define $\pi_x(j) = [\![j = j*]\!]$
- DL-1-VS. (DL-1-R uses raw precision instead of variable smoothing.)
  - $\boxed{\text{No threshold}}$
  - Set $\theta_{fj} = \tilde{q}_f(j;\epsilon)$
  - Where $\epsilon = \boxed{\frac{|X_f \Lambda|}{L} \cdot \frac{p(\mathrm{V}|f)}{p(\Lambda|f)}}$
  - Define $\pi_x(j) = \boxed{\frac{1}{m} \sum_{g \in F_x} \theta_{gj}}$

Theorem 3

Algorithm Y-1 using DL-1-VS or DL-1-R as base learning algorithm converges to local minimum of K.

Proof Sketch

- Like DL-EM proof
  - Training step: hold ϕ constant, adjust θ
  - Labeling step: hold θ constant, adjust ϕ
- Labeling step

$$\begin{aligned} K(x) &= \sum_{g \in F_x} H(\phi_x \| \theta_g) \\ &= \sum_j \phi_{xj} \sum_{g \in F_x} \log \frac{1}{\theta_{gj}} \end{aligned}$$

  - Minimize $K(x)$ by concentrating all mass in $\arg\min_j \sum_{g \in F_x} \log \frac{1}{\theta_{gj}}$
  - If training step minimizes over just Λ, any increase in K on unlabeled examples is compensated for in labeling step

Training Step

- Minimize K as function of θ, under constraint that $\sum_j \theta_{fj} = 1$
- Solution:

$$\theta_{fj} = \frac{1}{|X_f|} \sum_{x \in X_f} \phi_{xj}$$

- If ranging over Λ only (DL-1-R), reduces to raw precision:

$$\theta_{fj} = \frac{|X_f Y_j|}{|X_f \Lambda|} = q_f(j)$$

- If ranging over all examples (DL-1-VS), reduces to variably smoothed precision:

$$\begin{aligned} \theta_{fj} &= p(\Lambda|f) q_f(j) + p(\mathrm{V}|f) u(j) \\ &= \tilde{q}_f(j;\epsilon) \quad \text{where} \quad \epsilon = \frac{|X_f \Lambda|}{L} \cdot \frac{p(\mathrm{V}|f)}{p(\Lambda|f)} \end{aligned}$$

Detail

- Smoothed precision is mixture of raw precision and uniform distribution

$$\begin{aligned}\tilde{q}_f(j) &= \frac{|X_f Y_j| + \epsilon}{|X_f \Lambda| + L\epsilon} \\ &= \frac{q_f(j) + \delta}{1 + L\delta} \qquad \delta = \epsilon / |X_f \Lambda| \\ &= \frac{1}{1 + L\delta} q_f(j) + \frac{L\delta}{1 + L\delta} u(j)\end{aligned}$$

- Mixing coefficient is $p(\Lambda|f)$

$$\begin{aligned}\epsilon &= \frac{|X_f \Lambda|}{L} \cdot \frac{p(\mathrm{V}|f)}{p(\Lambda|f)} \\ L\delta &= \frac{p(\mathrm{V}|f)}{p(\Lambda|f)} \quad = \quad \frac{1}{p(\Lambda|f)} - 1 \\ \frac{1}{1 + L\delta} &= p(\Lambda|f)\end{aligned}$$

Sequential Variants

- Yarowsky variants
  - Y-1/DL-EM (L, LU)
  - Y-1/DL-1 (R, VS)
  - YS (P, R, FS)
- Somewhat like Collins & Singer "Yarowsky-Cautious"
- Algorithm YS
  - Add one feature f at a time
  - Label new examples that have f
  - Feature weights and labels are indelible

Three Variants

- Differ in update rule

$$\begin{aligned}
&\text{YS-P (``peaked'')} && \theta_{fj} = p(\Lambda|f)q_f(j) + p(\mathrm{V}|f)[\![j = j\dagger]\!] \\
& && j\dagger = \arg\max_j q(j|f) \\
&\text{YS-R (``raw'')} && \theta_{fj} = q_f(j) \\
&\text{YS-FS (``smoothed'')} && \theta_{fj} = \tilde{q}_f(j;\epsilon) \quad = \tfrac{1}{1+L\delta} q_f(j) + \tfrac{L\delta}{1+L\delta} u(j)
\end{aligned}$$

- Theorem 4: All three reduce K

Proof Sketch

$$\text{gain} = \sum_x \sum_{g \in F_x} \left[H(\phi_x^{\text{old}} \| \theta_g^{\text{old}}) - H(\phi_x^{\text{new}} \| \theta_g^{\text{new}}) \right]$$

- "Training": hold ϕ constant *except for unlabeled examples*. Choose f, modify θ_f, set labels for unlabeled examples that have feature f.
  - Unlabeled examples have $\phi_{xj} = 1/L$, $\theta_{gj} = 1/L$
  - Labeling them decreases K, include that in "training" gain
- "Labeling": change labels for old labeled examples
  - Does not increase K – same proof as for DL-EM and DL-1

"Training" Gain

- Special properties
  - K changes only for examples that possess feature f
  - Old θ_f is uniform distribution
  - All θ_g are uniform distribution for features g of unlabeled examples
  - Labeling dist ϕ_x is either $[\![x \in Y_j]\!]$ or uniform
  - New ϕ_x is $[\![j = j*]\!]$ for previously unlabeled examples with f
- Gain:

$$|X_f\Lambda| \left[\log L - H(q_f \| \theta_f)\right] + |X_f\mathrm{V}| \left[\log L - \log \frac{1}{\theta_{fj*}}\right]$$

- Maximize it, result is update for YS-P:

$$\theta_{fj} = p(\Lambda|f) q_f(j) + p(\mathrm{V}|f) [\![j = j\dagger]\!]$$

Using Smoothed or Raw Precision

- Since $\log L = H(u)$:

$$\text{gain} = |X_f\Lambda|\left[H(u) - H(q_f\|\theta_f)\right] + |X_f\mathrm{V}|\left[H(u) - \log\frac{1}{\theta_{fj*}}\right]$$

- Since $H(u) \geq \log\frac{1}{\theta_{fj*}}$, gain is nonnegative if:

$$H(u) \geq H(q_f\|\theta_f)$$

  – We can show this is true if $\theta_f = \tilde{q}_f$, hence YS-FS increases gain

- Since $H(u) = H(q_f\|u)$, the previous condition is equivalent to:

$$D(q_f\|u) \geq D(q_f\|\theta_f)$$

  – This is true if $\theta_f = q_f$, so YS-R increases gain

Summary

Y-1/DL-EM (L, LU)	Y-1 close to original DL-EM not	optimize H parallel update
Y-1/DL-1 (R, VS)	Y-1 and DL-1 close to original	optimize K parallel update
YS (P, R, FS)	FS from original	improve K sequential update

- Differences from original
  - No thresholding in training or labeling
  - No "unlabeling"
  - Mixture prediction rather than "max" prediction

Connection to Co-Training

$$H = \sum_x \left[H(\phi_x) + D(\phi_x \| \pi_x) \right]$$

- If $D(\phi_x \| \pi_x)$ is small and $H(\pi_x)$ is small, then $H(\phi_x)$ must be small

$$H(\pi_x) \leq \frac{1}{m} \sum_{f \in F_x} H(\theta_f) + \frac{1}{m^2} \sum_{f \in F_x} \sum_{g \in F_x} D(\theta_f \| \theta_g)$$

- Hence:

if features are confident	$H(\theta_f)$ is small
and **they agree with each other**	$D(\theta_f \| \theta_g)$ is small
then $H(\pi_x)$ is small	

- Find confident features that agree on unlabeled data, label them consistently with labeled data. Minimizes H.