WFST enabled Minimum Bayes Risk (MBR) Training

Tina Raissi

June 26, 2023

1 Training Criterion

For an acoustic feature sequence $X = x_1^T$ and a word sequence $W = w_1^N$, of length T and N, respectively, define the output sequence a_1^S corresponding to W. This can be phonemes, wordpieces, or characters.

For a set of parameters θ, the statistical formulation of automatic speech recognition task based on maximum a-posteriori probability (MAP) calculates the Bayes risk via a loss function $L(.,.)$ as follows

$$\mathcal{R}_{\mathrm{B}}(\theta) = \sum_{\bar{W}\in\mathcal{W}} \sum_{W\in\mathcal{W}} \int p_\theta(W|X) L(\bar{W}, W) pr(\bar{W}, X) dX \tag{1}$$

where $\mathcal{W}$ is the hypothesis space, and $\bar{W}$ is the correct hypothesis for X. Note the following points:

- Not every loss function is suitable in terms of a tractable computation. We need a loss that considers local dependency, e.g., Hamming distance.
- if L is Levenshtein distance, then Eq. (1) gives the expected word error rate (WER)
- The true probability distribution $pr(\bar{W}, X)$ is generally not known, in practice we approximate this with a given training data

For a finite training set of utterances $\{X\}_{r=1}^M$, we define an accuracy function $A(.,.)$. The training in Eq. (2) is known as *minimum Bayes risk*[1]

$$\mathcal{F}_{\mathrm{BR}}(\theta) = \frac{1}{M} \sum_{r=1}^{M} \sum_{W\in\mathcal{W}} p_\theta(W|X^r) A(\bar{W}^r, W) \tag{2}$$

It is possible to have different accuracy functions on phone or state level for minimum phone error (MPE)[2] and state-level minimum Bayes risk (sMBR) trainings, respectively. By using log probabilities and accuracy function equal to Kronecker delta we end up in maximum mutual information (MMI) training. It is possible to reformulate Eq. (2) via the Bayes identity as:

$$\mathcal{F}_{\mathrm{BR}}(\theta) = \sum_{r=1}^{M} \sum_{W\in\mathcal{W}} \frac{p_{\theta_{\text{AM}}}(X^r|W) p_{\theta_{\text{LM}}}(W) A(\bar{W}^r, W)}{\sum_{W'} p_{\theta_{\text{AM}}}(X^r|W') p_{\theta_{\text{LM}}}(W')} \tag{3}$$

where $\theta = \{\theta_{\text{AM}}, \theta_{\text{LM}}\}$, consisting of acoustic and language model parameters. Assume we are already given an LM.

2 WFST/Graph Representation

For decoding or sequence discriminative training we generally need to consider the set of all possible word sequences called also as the hypothesis space. This can be done by taking an N-best list of the most likely hypotheses or to use lattices, called sometimes also as graphs or Lattices.

A transducer $\mathcal{L}$ is an a-cyclic graph where nodes represent the time steps and edges represent the words. Each arc can be augmented with weights. Each path π through $\mathcal{L}$ consists of a sequence of edges e_1^N corresponding to a word sequence of length N. The edges carry the product of the acoustic and language model.

$$p(e_1^N, X) = \prod_{n=1}^{N} w[e_n] \tag{4}$$

The probability of a certain word within the sequence is equal to the sum over probability of all transducer paths $\pi(\mathcal{L})$ going through that specific edge:

$$p(e|X) = \sum_{\substack{e_1^N \in \pi(\mathcal{L}) \\ \exists n: e_n = e}} p(e_1^N|X) \tag{5}$$

The sum over all paths is generally carried out via forward-backward algorithm.[1]. Denote forward and backward scores accumulated at a certain transducer node s by $\phi(s)$ and $\psi(s)$, respectively. For partial edge sequences e_1^k and e_h^N reaching the edge e in forward and backward paths we have

$$\mathbf{fb}(e, X) = \sum_{\substack{e_1^k: \\ n[e_k]=p[e]}} \prod_{m=1}^{k} w[e_m] \cdot w[e] \cdot \sum_{\substack{e_h^N: \\ p[e_h]=n[e]}} \prod_{n=h}^{N} w[e_n] \tag{6}$$

Following Eq. (6), we can calculate the prior as $\phi(F) = p(X)$. Consequently, it is possible to calculate the arc posterior as $\mathbf{fb}(e|X) = \frac{\mathbf{fb}(e,X)}{p(X)}$.

3 Expectation Semiring

It is possible to define a specific semiring for the operations on the transducer. Typically, the MBR training relies on the *expectation semiring*[3] defined with the following operations:

- Each edge is augmented with the weight $(p, v) \in \mathbb{R}^+ \times \mathbb{R}$ of probability and value
- $(p_1, v_1) \oplus (p_2, v_2) = (p_1 + p_2, v_1 + v_2)$
- $(p_1, v_1) \otimes (p_2, v_2) = (p_1 p_2, p_1 v_2 + v_1 p_2)$
- $\bar{1} = (1, 0)$
- $\bar{0} = (0, 0)$
- $inv(p, v) = (p^{-1}, -p^{-2} v)$

Definition of different semirings relevant for training and decoding in ASR.(Source: [4])

Semiring	$\mathbb{K}$	$x \oplus y$	$x \otimes y$	$\bar{0}$	$\bar{1}$
probability	$\mathbb{R}^+$	$x + y$	$x \cdot y$	0	1
log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$-\log(\exp(-x) + \exp(-y))$	$x + y$	$+\infty$	0
tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\min(x, y)$	$x + y$	$+\infty$	0
expectation	$\mathbb{R}^+ \times \mathbb{R}$	$(x_1 + y_1, x_2 + y_2)$	$(x_1 x_2, x_1 y_2 + y_1 x_2)$	$(0, 0)$	$(1, 0)$

We augment each arc of the transducer with the pair (p, v), where p is similar to the arc weights when we have a probability semiring, i.e. $w[e][p] = p$. For a certain cost c, e.g. the value of the accuracy function, the value is initialized as $w[e][v] = w[e][p] \cdot c[e]$. The cost shall be additive along each path in the transducer:

$$c(e_1^N) = \sum_{n=1}^{N} c[e_n] \tag{7}$$

The forward-backward computation on the transducer using the expectation semiring makes use of the accumulated forward and backward scores at a certain node s, as shown in Eq. (8) and Eq. (9), respectively.

$$\phi(s) = \bigoplus_{\substack{e_1^n: \\ n[e_n]=s}} \bigotimes_{m=1}^{n} w[e_m] \tag{8}$$

$$\psi(s) = \bigoplus_{\substack{e_1^n: \\ p[e_n]=s}} \bigotimes_{m=1}^{n} w[e_m] \tag{9}$$

The multiplication of the weights of the arcs through the transducer in the above equations can be roken down into the multiplication of the probabilities over a transducer using the probability semiring, and the cross multiplication between cost and probability for the value as follows:

$$\bigotimes_{n=1}^{N} w[e_n] = \bigotimes_{n=1}^{N} (p_n, v_n) = \left(\prod_{n=1}^{N} p_n, \sum_{n=1}^{N} v_n \prod_{\substack{m=1 \\ m \neq n}} p_m \right) \tag{10}$$

You can see the first element of the resulting tuple in Eq. (10) as a probability distribution over the paths in the transducer, since the sum over all paths of this would sum up to one. The second element would represent the expected cost. We show this in relation to the sMBR gradient in Section 5. For the calculation of the expected cost we use the probability and the value components of the forward-backward values with the interpretations shown in Eq. (11) and Eq. (12), respectively.

$$\mathbf{fb}(e, X)[p] = p(e, X) \tag{11}$$

$$\mathbf{fb}(e, X)[v] = p(e, X) \cdot c[e] \tag{12}$$

[1]For notation completeness refer also to Lucas' document

By using Eqs. (11) and (12) this would be equivalent to we can calculate the cost of each arc as:

$$c[e] = \frac{\mathbf{fb}(e, X)[v]}{\mathbf{fb}(e, X)[p]} \tag{13}$$

4 Gradient of MBR Training Criterion

We calculate the derivative $\frac{\partial \mathcal{F}_{\text{BR}}}{\partial \theta_{\text{AM}}}$ with respect only to the acoustic model parameters. For simplicity, we define the expected cost for r-th utterance under the choice of $\theta_{\text{AM}}$and the loss function as shown in Eq. (14).

$$\mathbb{E}_{p_{\theta_{\text{AM}}}}[A_r] = \sum_W \frac{p_{\theta_{\text{AM}}}(X^r|W)p_{\theta_{\text{LM}}}(W)A(\bar{W}^r, W)}{\sum_{W'} p_{\theta_{\text{AM}}}(X^r|W')p_{\theta_{\text{LM}}}(W')} \tag{14}$$

In the following steps, we denote $s_1^{T_r}$ as the aligned hidden state sequence to the utterance X^r. The relation between the state sequence and the word sequence is defined by an intermediate mapping step, where we assign one output label to each state. In standard Hybrid NN/HMM this can be a generalized triphone (CART label). One can define any type of state-tying. We assume there is a unique mapping of the output label to a certain word sequence. Given this two deterministic and unique mapping steps, we can drop the dependency to the word sequence and operate on the state level.

Similarly to maximum likelihood criterion (ML) criterion, the derivative can be seen as the sum over the training utterances and over time of weighted log derivatives. The weights are normalized state occupancies and state accuracies in ML and MBR, respectively. By using the definition of state accuracy $a(s|X^r)$ in Eq. (15), we have the derivative of Eq. (16), as shown in [4].

$$a(s|X^r) = \sum_W \left[\left(A(\bar{W}^r, W) - \mathbb{E}_{p_{\theta_{\text{AM}}}}[A_r]\right) \cdot \sum_{s_1^{T_r}|W:s_t=s} \frac{p_{\theta_{\text{AM}}}(X^r, s_1^{T_r})p(W)}{\sum_{W'} p_{\theta_{\text{AM}}}(X^r, W')} \right] \tag{15}$$

$$\frac{\partial \mathcal{F}_{\text{BR}}}{\partial \theta_{\text{AM}}} = \sum_{r=1}^{M} \mathbb{E}_{p_{\theta_{\text{AM}}}}[A_r] \sum_{t=1}^{T_r} a(s|X^r) \frac{\partial \log p_{\theta_{\text{AM}}}(x_{rt}|s_t)}{\partial \theta_{\text{AM}}} \tag{16}$$

5 SMBR Training with Semiring Operations

It is possible to express the derivative in terms of semiring operations. We need the following points:

- $\mathbf{P}$: a probability WFST on state level, where each arc has the information about the time step and the state, with $w[e] = p_{\theta_{\text{AM}}}(x_t, s_t|.)$.
- $\mathbf{A}$: an accuracy WFST with the same structure as $\mathbf{P}$ but with arc weights having the value from the $A(.,.)$
- $\mathbf{E}$: an auxiliary WFST with the same structure as $\mathbf{P}$ and $\mathbf{A}$ with arc weights $w_{\mathbf{E}}[e] := (w_{\mathbf{P}}[e], w_{\mathbf{P}}[e]w_{\mathbf{A}}[e])$
- $\mathbf{Q}$: an expectation WFST that has posterior probabilities from $\mathbf{P}$ as the first element and expected accuracy of the arc
- $\nabla \log \mathbf{P}$: gradient WFST that has the derivative of the log probabilities from $\mathbf{P}$
- Formulation of the expected cost in terms of forward-backward scores in expectation semiring

The expected cost is defined as the sum over expected cost of the arcs and can be shown as follows:

$$\begin{aligned}\mathbb{E}_{p_{\theta_{\text{AM}}}}[\mathbf{A}_r] &= \frac{\sum_{e_1^N} w_{\mathbf{E}}[e_1^N][p] \cdot w_{\mathbf{E}}[e_1^N][c]}{p(X)} \\ &= \sum_{e_1^N} p(e_1^N|X)c(e_1^N) \\ &= \sum_{e} p(e|X)c(e)\end{aligned}$$

In practice this is done by taking the probability and value components of a complete forward pass.

$$\mathbb{E}_{p_{\theta_{\text{AM}}}}[\mathbf{A}_r] = \frac{\phi(F)[v]}{\phi(F)[p]} \tag{17}$$

It has been shown that the gradient in Eq. (16) can be written via semiring computation[5]. Namely, by using the covariance of the gradient WFST $\nabla \log \mathbf{P}$ and $\mathbf{A}$ with respect to $\mathbf{P}$. This will lead to Eq. (18) with the definition in Eq. (19).

$$\nabla \mathcal{F}_{\text{BR}} = \sum_{r=1}^{M} \mathbb{E}_{p_{\theta_{\text{AM}}}}[A_r]\mathbf{Cov}(\mathbf{A}_r, \nabla \log \mathbf{P}_r) \tag{18}$$

$$\mathbf{Cov}(\mathbf{A}, \nabla \log \mathbf{P}) = \sum_{e \in \mathbf{P}} w_{\mathbf{Q}}[e][v].w_{\nabla \log \mathbf{P}}[e] \tag{19}$$

References

[1] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition." in *Proc. of Interspeech*, vol. 6, 2006, pp. 2406–2409.

[2] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. of IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2002, pp. I–105.

[3] J. Eisner, "Expectation semirings: Flexible em for learning finite-state transducers," in *Proceedings of the ESSLLI workshop on finite-state methods in NLP*, 2001, pp. 1–5.

[4] B. Hoffmeister, G. Heigold, D. Rybach, R. Schluter, and H. Ney, "Wfst enabled solutions to asr problems: Beyond hmm decoding," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 551–564, 2011.

[5] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "Modified mmi/mpe: A direct evaluation of the margin in speech recognition," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 384–391.