Modeling Phonetic Context with Non-random Forests for Speech Recognition

Hainan Xu*, Guoguo Chen*, Daniel Povey*† and Sanjeev Khudanpur*†

* Center for Language and Speech Processing
† Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA

Overview

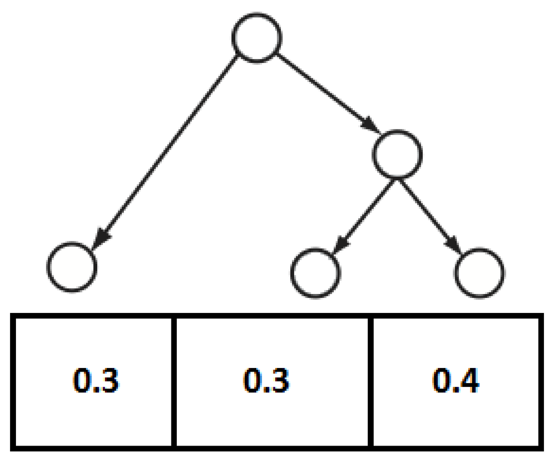
- Modern ASR systems use decision trees to map triphones into equivalence classes as units for parameter sharing
- We introduce a deterministic method for building multiple complementary decision trees by introducing an entropy term in the objective function for tree-building
- Acoustic emission scores are combined during decoding and we see consistent gains from the use of multiple trees

Decision Trees

- Phonetic decision trees are used to map context dependent phones into equivalence classes as units for parameter sharing
- One tree might be biased and we want to use multiple decision trees and combine systems built on different trees
- **Problem**: the standard procedure for building the tree is deterministic; we want a deterministic method to build decision trees and preferably be able to control how "different" they are
- Our solution: including an entropy term in the objective function for tree building

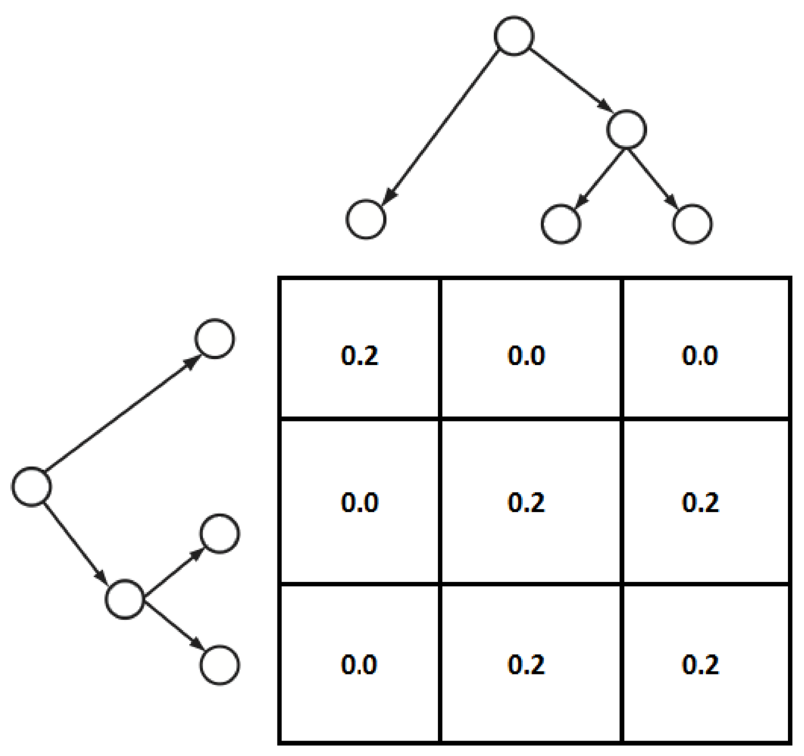
Entropy of a Decision Tree and Decision Trees

- A decision tree defines a "distribution" on the data, of which we could compute entropy



$$\mathcal{H}(d) = -0.3\log 0.3 - 0.3\log 0.3 - 0.4\log 0.4$$

- n decisions divides the data into partitions on a n-dimension grid, of which we could compute "joint-entropy".



$$\mathcal{H}(D) = -0.2\log 0.2 - 0.2\log 0.2 - 0.2\log 0.2 - 0.2\log 0.2 - 0.2\log 0.2$$

Note: Not all combinations of leaves are possible; also not all possible combinations occur in data, i.e. having non-zero "probability" in the partition.

Objective Function for Building Multiple Trees

- In the standard single tree case, the objective function is (normalized as per frame) Gaussian likelihoods on the data, denoted by $L(d)$, where d is a tree;
- We have defined entropy of a single tree $H(d_i)$ and entropy of trees $H(D)$.
- For building multiple trees, we define the new objective function as

$$\sum_{i=1}^{n} L(d_i) + \lambda\left(\mathcal{H}(D) - \frac{\sum_{i=1}^{n}\mathcal{H}(d_i)}{n}\right)$$

Key Observation

The 2nd and 3rd terms ensures that the joint-entropy grows larger, while entropies of single trees remain small. Thus the large joint-entropy has to be a result of trees being different.

System Setup

- We build the trees following the *Young and Woodland* paper, with splitting and merging stages; we made a minor modification to make the algorithm work with multiple trees.
- After trees are built, different acoustic models are built on top of each tree and trained independently
- During decoding, we combine acoustic log-likelihood estimates from different models to get a combined score
- For observation o and triphone state s, if the log-likelihoods given by each model are $\log p_1(o|s), \log p_2(o|s), ..., \log p_n(o|s)$, then the combined log-likelihood is (we try to favor the larger log-probabilities)

$$\log \bar{p}(o|s) = \frac{\sum_i \log p_i(o|s)\exp(C\cdot\log p_i(o|s))}{\sum_i \exp(C\cdot\log p_i(o|s))}, C = 0.1$$

- For transition probabilities in HMMs, we simply take the algebraic means.
- To generate the decoding graph, we build a "virtual tree" such that each of its leaf corresponds to a unique and valid combination of leaves in each individual trees.

Experiments

- We evaluate our system on 4 datasets: *WSJ*, *SWBD*, *TED-LIUM* and *Librispeech*.
- Impact of the entropy term

# trees	λ	avg-entropy	joint-entropy
1	-	7.63	7.63
2	0.1	7.67	7.85
2	0.25	7.72	8.11
2	0.5	7.76	8.41
2	1	7.78	8.78
3	1	7.74	9.00
4	1	7.72	9.07

Table 1: Entropy of multi-trees (*TED-LIUM*)

# trees	λ	avg # leaves	# virtual-leaves
1	-	3973	3973
2	0.1	4030	8173
2	0.25	4115	12969
2	0.5	4204	21138
2	1	4237.5	36828
3	1	4123	97999
4	1	4078.5	164811

Table 2: Number of leaves in multi-trees (*TED-LIUM*)

- Comparison between the single tree and the multi-tree method

	dev		test	
# trees	clean	other	clean	other
baseline	5.93	20.42	6.59	22.47
tree 1	6.20	20.67	6.75	22.68
tree 2	6.27	21.07	6.87	22.84
multi	**5.82**	**19.86**	**6.46**	**21.62**

Table 3: WER of individual and combined DNN models on *Librispeech* ($\lambda = 1$)

- More results on the recognition accuracy of multi-tree systems

	WSJ		SWBD		TED-LIUM	
# trees	eval92	dev93	swbd	eval2000	dev	test
1	7.07	4.06	13.4	19.2	21.7	19.4
2	6.55	4.08	13.0	18.8	**21.2**	18.6
3	**6.46**	**3.72**	**12.8**	**18.7**	**21.2**	**18.5**

Table 4: WER of DNN models on *WSJ*, *SWBD* and *TED-LIUM* ($\lambda = 1$)

	dev		test	
# trees	clean	other	clean	other
1	5.93	20.42	6.59	22.47
2	5.82	19.86	6.46	**21.62**
3	**5.80**	**19.77**	**6.27**	21.68

Table 5: WER of DNN models on *Librispeech* ($\lambda = 1$)

Conclusions

- Combination of models trained on different trees could consistently give better results than single tree systems; the gains are larger for noisy speech.
- More trees generally help; though the relative gain becomes smaller for larger numbers.

Acknowledgements

This work was partially supported by NSF Grant No IIS 0963898, DARPA BOLT Contract No HR0011-12-C-0015, and an unrestricted gift from Google Inc. (No 2012-R2-106).