A Basis Method for Robust Estimation of Constrained MLLR

Dan Povey and Kaisheng Yao

Microsoft

CMLLR intro (1)

- Constrained Maximum Likelihood Linear Regression (CMLLR)
- Also known as feature-space MLLR (fMLLR)
- Is an affine transformation of the features (affine = linear plus an offset):

$$\mathbf{x} \rightarrow \mathbf{A}^{(s)}\mathbf{x} + \mathbf{b}^{(s)}$$

(s is the speaker)

CMLLR intro

- Common to write it as a single matrix:

$$\mathbf{x} \rightarrow \mathbf{W}^{(s)} \mathbf{x}^{+}$$

where

$$\mathbf{x}^{+} = \left[\mathbf{x}^{T}\ ,\ 1\right]^{T}$$

and

$$\mathbf{W}^{(s)} = \left[\mathbf{A}^{(s)}\ ;\ \mathbf{b}^{(s)}\right]$$

CMLLR estimation

- Maximize log-likelihood of transformed data, plus #frames times log-det of $\mathbf{A}^{(s)}$
- Need for log determinant is obvious if we view this as a model-space transform ("constrained MLLR" perspective)
- Compact suff. stats and efficient update.
- CMLLR is probably the most widely used adaptation method:
  - It's simple, effective and robust
  - It works with relatively little adaptation data

CMLLR with very little adaptation data

- For typical setups, CMLLR starts giving improvements at around 5 seconds of adaptation data
- The improvement saturates at, say, 30 seconds of adaptation data
- Many tasks of interest have less than 30 seconds (often less than 10 seconds) of adaptation data.
- We would like to improve CMLLR performance in this regime.

Prior work

- Simple approaches to reducing #parameters:
  - Diagonal CMLLR (make $\mathbf{A}^{(s)}$ diagonal)
  - Block-diagonal $\mathbf{A}^{(s)}$
- Bayesian approaches
  - fMAPLR (MAP solution given an appropriate prior)
  - Need to choose a prior that's tractable
  - This generally means ignoring correlations between rows
- Parameter reduction using a basis
  - See next slide

Parameter reduction using a basis

- Choose a basis size N (less than the total #parameters in the transform), estimate a set of basis matric$\mathbf{W}_n$ somehow, and use:

$$\mathbf{W}^{(s)} = \sum_{n=1}^{N} d_n^{(s)} \mathbf{W}_n$$

- In test time, there are N parameters to estimate.
- Prior work (Visweswariah et. al) used this approach

Relationship to prior work

- There are a couple of differences between our approach and that prior work:
  - We allow the basis size to vary with amount of adaptation data (so don't have to "tune for" a specific amount of adaptation data)
  - We have a neat way of estimating the basis and a fast way of estimating the coefficients
- Our most important contribution is to (try to) popularize this method
  - Previous work not widely known
  - We wrote this method as a journal paper (CSL; accepted; not yet published but submitted version available online), in which we explain in great detail how to implement it

Basis methods vs. alternatives

- Approaches like making $\mathbf{A}^{(s)}$ diagonal are very ad hoc and do not work very well.
- Bayesian methods seem like the "right way" to do it in principle.
  - Don't have to make a "hard decision" about how many parameters to estimate
  - Just smooth each parameter the "right amount"
- But they are not very easy to take advantage of:
  - For best performance, would have to use priors that correlate transform rows. Very inefficient.
  - Prior hard to estimate (cannot observe $\mathbf{W}^{(s)}$)
  - Forest of alternatives.

Estimating the basis

- Want to estimate the basis matrices $\mathbf{W}_n$ for $1 \leq n \leq D(D+1)$
- Need an "ordered" basis from most to least important (will choose a variable number of coefficients, depending on amount of data).
- PCA would give us this…
  - But it is very ad hoc
  - Previous work showed it's important to estimate basis using ML (but their method not applicable when basis size will vary in test time)

Preconditioned PCA

- There is a situation in which PCA would be the "right thing to do", i.e. would approximate ML.
- Think of $\mathbf{W}$ as a vector (concatenate the rows)
- If the objective function were quadratic, and if the Hessian (i.e. the quadratic term) for each speaker s was a multiple of the unit matrix… then weighted PCA would be the same as ML.
- We use a change of variables, so that in an appropriately averaged sense, this Hessian is proportional to the unit matrix.
- Not quite Maximum Likelihood, but pretty close.

Preconditioned PCA (cont'd)

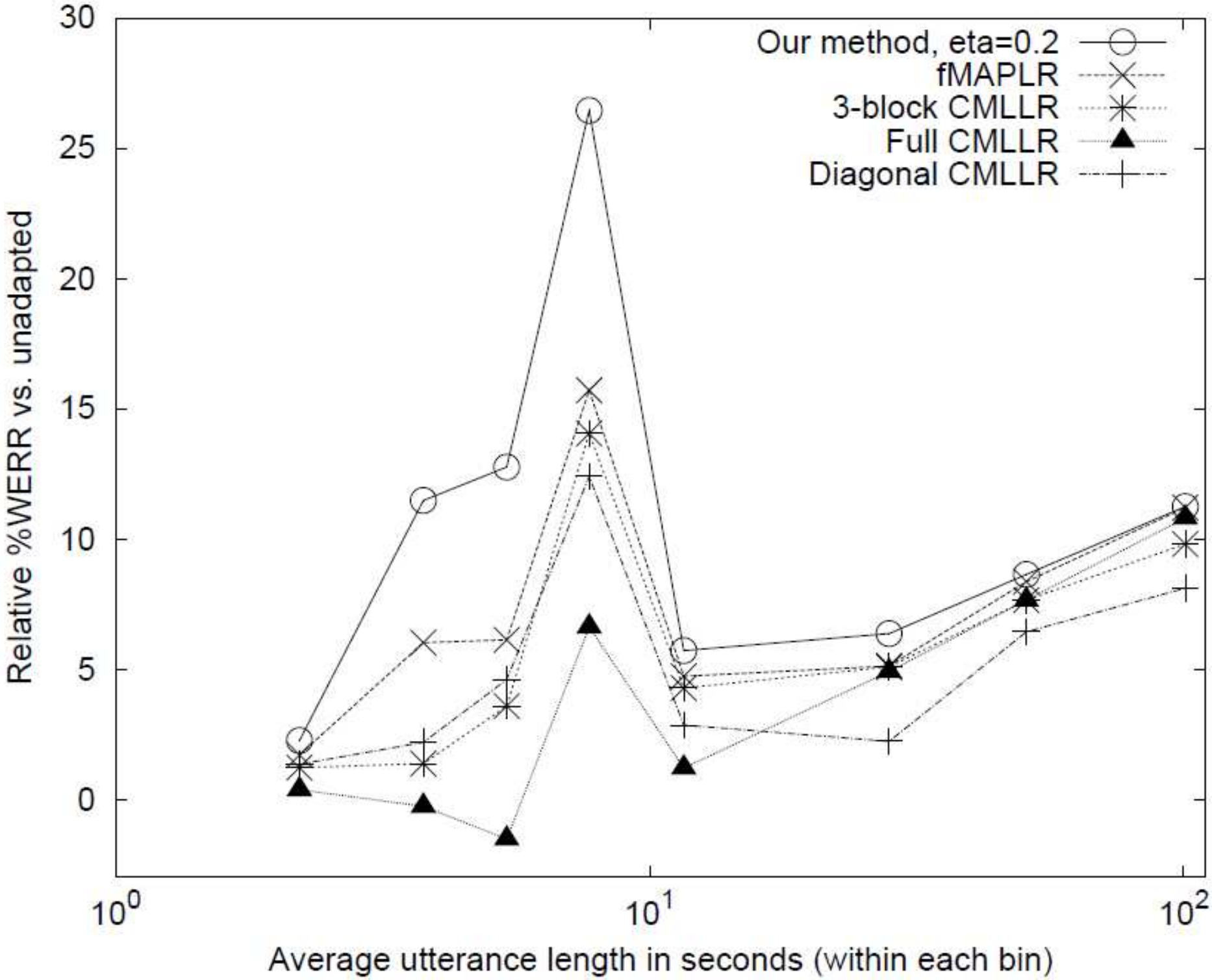
- We compute a matrix M that approximates the average Hessian… (note: M is D(D+1) x D(D+1))
- We use the Cholesky factor of M to perform a change of variables that makes the average Hessian proportional to the unit matrix.
- We compute the appropriately weighted scatter of gradients of the objective function, in this space, and do PCA on it that.
- We can show that this is a reasonable approximation to Maximum Likelihood

Test-time computation

- Choose a number of coefficients proportional to the amount of adaptation data
  - In our experiments we added one new parameter for every 5 new frames
  - Remaining coefficients will be zero.
- For ~10 iterations, we:
  - Choose a search direction (either steepest descent, or conjugate gradient; both work fine)
  - Do line search using ~3 iterations of Newton's method.
- The preconditioning ensures that this converges fast (Hessian close to multiple of unit matrix).

Experimental setup

- Have two setups:
  - Interactive Voice Response (IVR): short utterances, small vocabulary, low WER.
  - Enhanced Voice Mail (EVM): longer utterances (voice mails), large vocabulary, higher WER
  - Both have gender dependent models, online cepstral mean normalization.
- We divided each dataset into four subsets (“bins”) based on utterance duration
- Mean bin duration ranges from 2.2 s to 101 s.
- Note: first four bins are IVR, next four are EVM.


Our method, eta=0.2
fMAPLR
3-block CMLLR
Full CMLLR
Diagonal CMLLR
Relative %WERR vs. unadapted
30
25
20
15
10
5
0
10^0
10^1
10^2
Average utterance length in seconds (within each bin)

Results: points to note.

- X-axis is log (bin duration). Each point is average WER reduction (WERR) for one bin.
- WERRs on left side are larger because of lower absolute WER in IVR.
- Leftmost point (2 seconds): very little improvement with any technique.
- Rightmost point (100 seconds): no difference between our method, fMAPLR and fMLLR.
- In between, we get improvement
- Our technique consistently the best.

Results: further observations

- Normal CMLLR doesn't improve WER at all until about 5 seconds of speech.
- However, fMAPLR, diagonal and block-diagonal CMLLR improve WER even at ~3 seconds
- Our method gives about twice as much improvement as these methods, for less than ~10 seconds of speech.
- Our method becomes equivalent to CMLLR in the limit of large amount of data
- When asking how much this would help in any given situation, take into account the distribution of utterance lengths.

Conclusions

- Have described a basis method to make CMLLR estimation robust to small amounts of adaptation data.
- Considerably better than commonly used baselines (for short utterance lengths)
- In journal paper, we describe in detail how to implement it.