HMM-based acoustic model adaptation and discriminative training

Steven Wegmann
ICSI

11 April 2012

HMM-based adaptation and discriminative training are important techniques for improving accuracy

Both procedures start with HMM's ML parameters

- Estimated using a large training corpus drawn from many speakers

Both procedures adjust the model parameters

- Adaptation: model estimation using limited, novel data
- Discriminative training: uses "discriminative" estimation criteria

However the goals of the two procedures differ:

- Adaptation: specialization
- Discriminative training: compensation for model failures

What is acoustic model adaptation?

A procedure to adapt or target a speech recognizer to

- A specific acoustic environment
- A particular speaker

To understand how this works, we need to understand

- The adaptation problem
- Two adaptation procedures

HMM parameters

We use (mixtures of) multivariate normal distributions for our output distributions

For simplicity we will discuss 1-dimensional, unimodal models, so the distribution for state l (there are $L \equiv L(M)$ states)

$$x \mid q_l \overset{i.i.d}{\sim} N(\mu_l, \sigma_l^2)$$

Thus the parameters of our acoustic models consist of

- means and variances for the output distributions (important)
- the transition matrices for the states (not so important for speech recognition)

We use HMMs to model triphones

A triphone is just a phone in context

- Phone b preceded by a, followed by c: a-b+c

We typically use three state HMMs for each triphone

There is tremendous variability in the amount of training data for each triphone

- We cluster triphones (at the state level)
- Top-down clustering using decision trees

The acoustic model adaptation problem

We have generic models trained/estimated from a large amount of data recorded from many speakers

- Usually we train from thousands of hours of recordings from thousands of speakers

We are given a relatively small amount of novel data

- From a new/unseen acoustic environment (say 20 hours)
- From a new speaker (maybe as little as a minute)

Our task is to obtain new model parameters that are a better fit for this new task or speaker

- We will sacrifice some of the generic model's generality

The acoustic model adaptation problem (cont'd)

We preserve the structure of the generic HMM

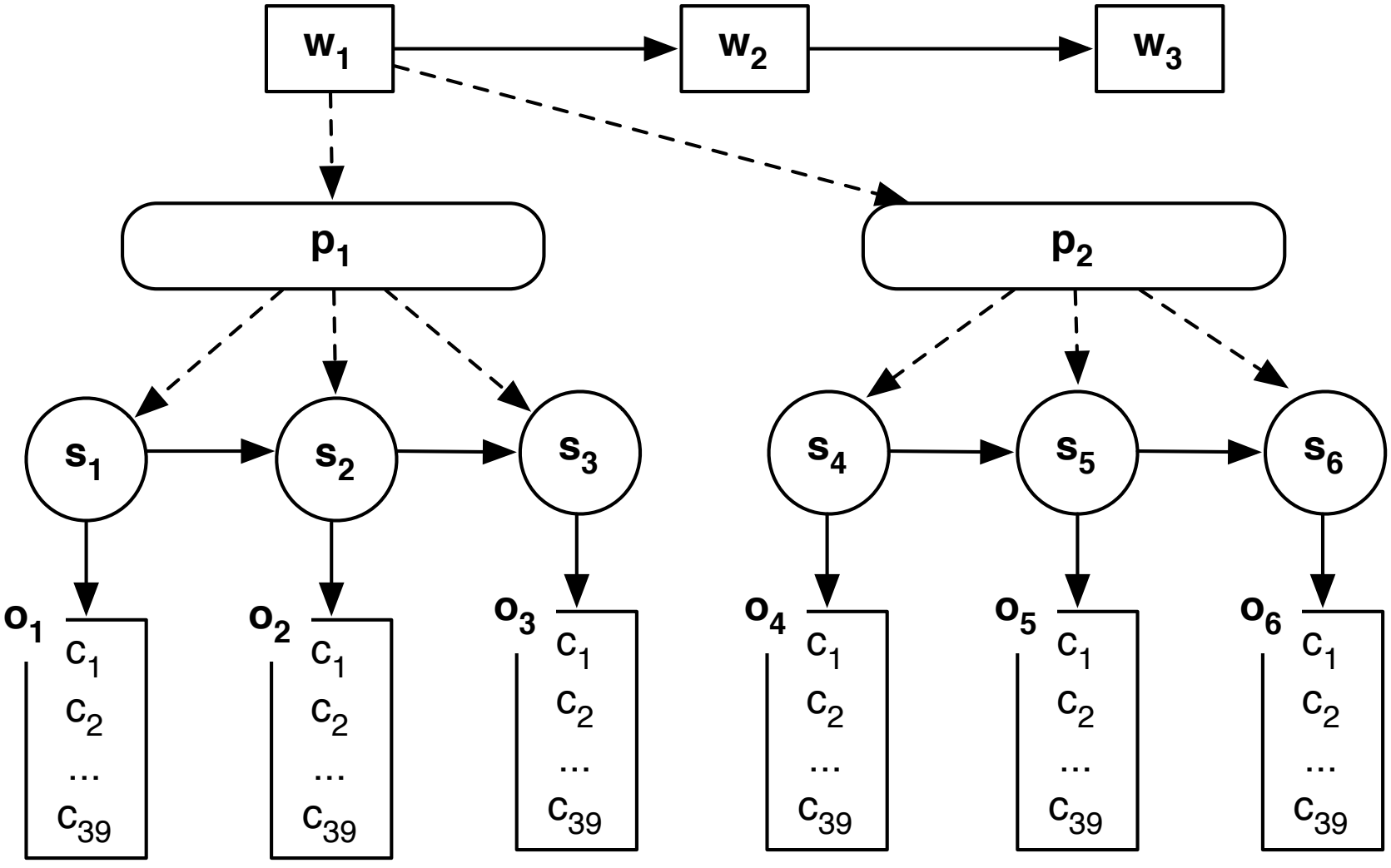
- We only adjust the output distribution means and variances

In particular, we do not retrain starting from scratch with the new data

- We do not have enough data to train full blown models

Hence the terminology *adaptation*

We need transcripts for training



Notation: $s = q$ states, $o = x$ observations

Two modes of adaptation

Adaptation data is just like training data in that it consists of transcribed audio data

- How do we get the transcripts?

Supervised adaptation

- We are given (accurate) transcripts
- Closest to training, most accurate, but may not be realistic

Unsupervised adaptation

- We need to produce the (errorful) transcripts via recognition
- Errors in transcripts degrade adaptation performance

The acoustic model adaptation problem (cont'd)

For clarity without effecting generality

- We will focus on the speaker adaptation problem
- We will work in one feature dimension

The original models θ^{SI} are *speaker independent*

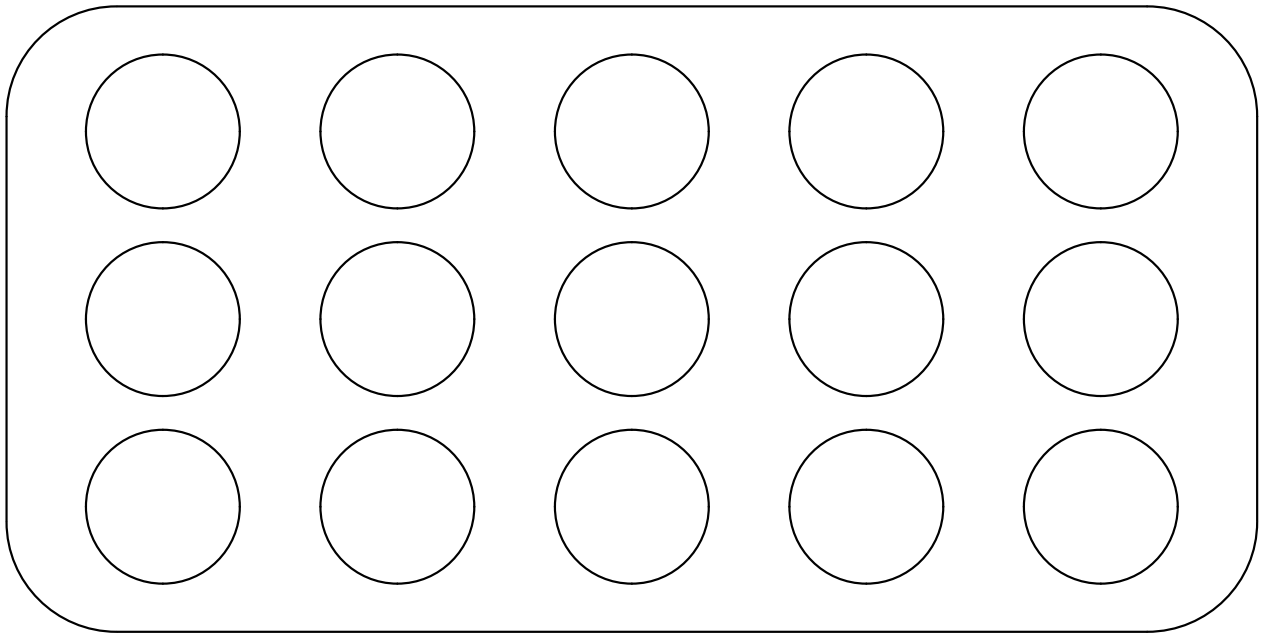
- Model parameters $\{\mu_l^{SI}, \sigma_l^{SI}\}_{l=1}^{L}$
- Training frames $\{y_t\}_{t=1}^{M}$

The adapted models θ^{SD} are *speaker dependent*

- Model parameters $\{\mu_l^{SD}, \sigma_l^{SD}\}_{l=1}^{L}$
- Training frames $\{x_t\}_{t=1}^{N}$

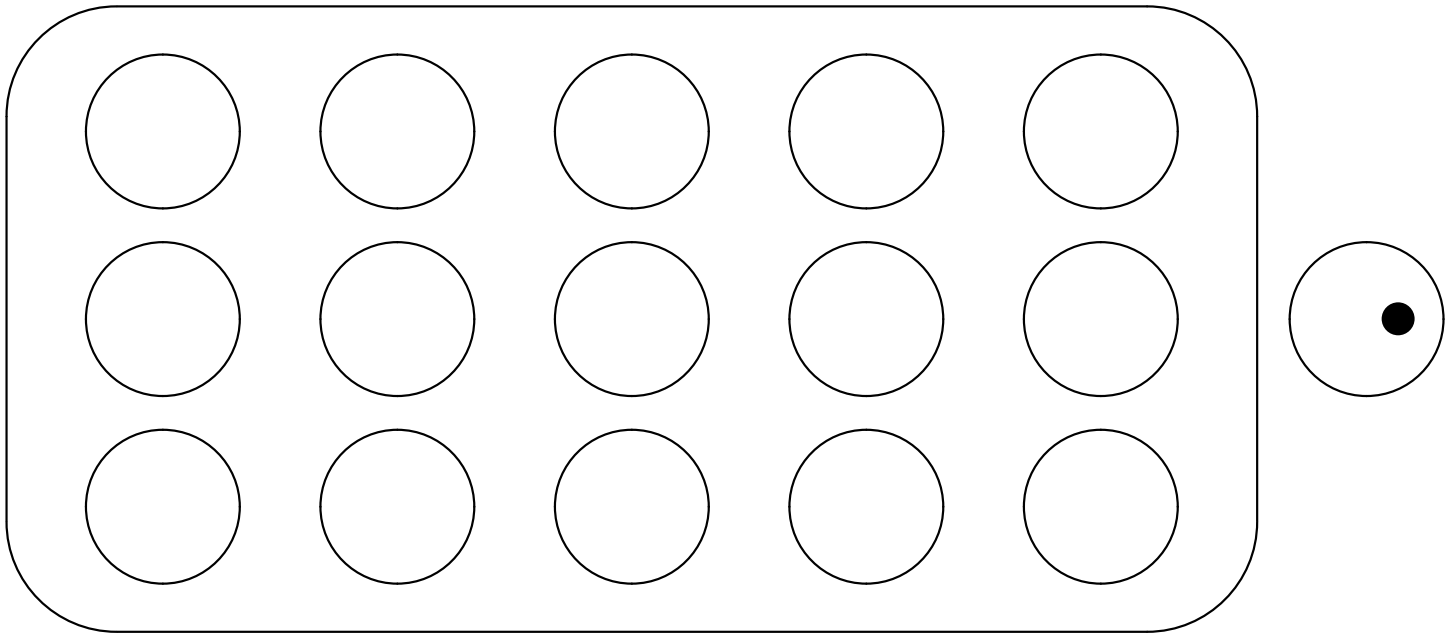
An idealized view of the training data

The oval represents the SI training data with the circles representing the observed training data from the individual training speakers



An idealized view of the adaptation problem

The circle outside of the oval represents all of the data ever produced by the new target speaker, while the black disk is the data we observe ($\{x_t\}_{t=1}^{N}$)



The adaptation problem restated

To adjust the generic speaker independent model so it becomes specialized to the target speaker

Given the small sample from the target speaker ($\{x_t\}_{t=1}^{N}$) we estimate speaker dependent means for all of the states that

- Fit/explain the small sample that we've been given
- Fit/explain all future data generated by this speaker

We will use statistical inference

- We also want to leverage the prior knowledge that the generic models summarize

The speaker independent means

A key part of the Baum-Welch algorithm for HMM parameter estimation is determining the probability distribution of the hidden states across a given frame y_t:

- $p(q_l^t \mid y, \theta^{SI})$
- $\sum_{l=1}^{L} p(q_l^t \mid y, \theta^{SI}) = 1$
- $p(q_l^t \mid y, \theta^{SI})$ is the fraction of frame y_t that is assigned to state q_l (at time t)

Then the ML estimate of the speaker independent mean for state l is the average of the fractional frames assigned to l:

$$\hat{\mu}_l^{SI} = \frac{\sum_{t=1}^{M} p(q_l^t \mid y, \theta^{SI}) y_t}{\sum_{t=1}^{M} p(q_l^t \mid y, \theta^{SI})}$$

A naive approach to adaptation

We use θ^{SI} to compute the fractional counts and set

$$\hat{\mu}_l^{SD} = \frac{\sum_{t=1}^{N} p(q_l^t \mid x, \theta^{SI}) x_t}{\sum_{t=1}^{N} p(q_l^t \mid x, \theta^{SI})}$$

It's useful to introduce the total of the estimated fractional count of frames assigned to state l:

$$\hat{n}_l^{SD} \equiv \sum_{t=1}^{N} p(q_l^t \mid x, \theta^{SI})$$

Where

$$\sum_{l=1}^{L} \hat{n}_l^{SD} = N$$

Problems with the naive approach: uneven counts

The distribution of the adaptation data across the states ($\hat{n}_l^{SD}$) will be far from uniform

- Some states, notably silence, will have a large fraction of the data ($\hat{n}_l^{SD}/N$)
- Other states will not have any adaptation data, i.e. $\hat{n}_l^{SD} = 0$
- This will be exacerbated when N is small

The resulting estimates, $\hat{\mu}_l^{SD}$, will vary in reliability

- If $\hat{n}_l^{SD} > 50$, then $\hat{\mu}_l^{SD}$ is probably a pretty good estimate
- If $\hat{n}_l^{SD} < 4$, then $\hat{\mu}_l^{SD}$ is probably not a very good estimate
- If $\hat{n}_l^{SD} = 0$, then $\hat{\mu}_l^{SD}$ doesn't even make sense

Problems with the naive approach: unreliable counts

Suppose the speaker dependent data is very different from the speaker independent models (or training data)

- Heavy accent
- Novel channel

This can result in unreliable fractional counts which are inputs to the estimates $\hat{\mu}_l^{SD}$

- $p(q_l^t \mid x, \theta^{SI})$

Unsupervised adaptation also leads to unreliable counts

Another naive approach: add $\{x_t\}_{t=1}^N$ to the training data

If we simply add the speakers data $\{x_t\}_{t=1}^N$ to the training data $\{x_t\}_{t=1}^N$ and re-estimate, then the resulting means are

$$\hat{\mu}_l^{ML} = \frac{\hat{n}_l^{SI}\hat{\mu}_l^{SI} + \hat{n}_l^{SD}\hat{\mu}_l^{SD}}{\hat{n}_l^{SI} + \hat{n}_l^{SD}}$$

Since we are assuming $\hat{n}_l^{SI} >> \hat{n}_l^{SD}$ we will have

$$\hat{\mu}_l^{ML} \approx \hat{\mu}_l^{SI}$$

Related question: when do we have enough data to directly estimate SD models?

Two linear adaptation methods

Two linear methods have been developed to address the problem of uneven counts

- MAP (maximum *a posteriori*)
- MLLR (maximum likelihood linear regression)

Multiple adaptation passes address the problem of unreliable counts

MAP and MLLR are examples of *empirical Bayes estimation*

Empirical Bayes (Robbins 1951, Efron and Morris 1973)

In traditional Bayesian analysis prior distributions are chosen before any data are observed

- In empirical Bayes prior distributions are estimated from the data

A example from baseball (Efron-Morris)

- We know the batting averages of 18 players after their first 45 at bats ($\{x_i\}_{i=1}^{18}$)
- We want to predict their batting averages at the end of the season (after 450 at bats)

The obvious solution is to use the early season averages individually

- We predict that player i will have average x_i

Empirical Bayes (cont'd)

There is a better solution that takes into account all of the available information: $y_i = \bar{x} + c(x_i - \bar{x})$

- $\bar{x}$ is the average of the x_i
- c is a "shrinkage factor" compute from the x_i (related to the variance)
- $0 < c < 1$
- $\bar{x}$ and c are empirical estimates of the prior distribution of the observed x_i.

Empirical Bayes applies to adaptation problem

Our adaptation problem is very similar to the baseball problem

- However, we are going to leverage more prior information
- Analogous to prior seasons information with other players

MAP and MLLR use the same empirical prior:

- The estimates from the training data $\{\hat{\mu}_l^{SI}, \hat{\sigma}_l^{SI}\}_{l=1}^{L}$

This empirical prior is used to adjust the speaker dependent means, $\{\hat{\mu}_l^{SD}\}_{l=1}^{L}$, to obtain new estimates:

- MAP uses interpolation
- MLLR uses weighted least squares

The MAP (maximum *a posteriori*) estimates

The MAP estimates for the means are interpolations

$$\hat{\mu}_I^{MAP} = \frac{\tau \hat{\mu}_I^{SI} + \hat{n}_I^{SD} \hat{\mu}_I^{SD}}{\tau + \hat{n}_I^{SD}}$$

- There is an analogous formula for the variances

The parameter τ, the prior count or relevance, determines the interpolation weight

- If $\tau = 0$, then $\hat{\mu}_I^{MAP} = \hat{\mu}_I^{SD}$
- If $\tau = \infty$, then $\hat{\mu}_I^{MAP} = \hat{\mu}_I^{SI}$
- If $\hat{n}_I^{SD} \gg \tau$, then $\hat{\mu}_I^{MAP} \approx \hat{\mu}_I^{SD}$

The parameter τ is a traditional Bayesian prior

The choice of τ is related to your belief about how many frames are necessary to reliably estimate means and variances

For example, I believe that

- A minimum of 5 to 10 frames are necessary for a mean
- 50 frames is reasonable number for a 39 dimensional, diagonal covariance

The value of τ determines when $\hat{\mu}_l^{MAP}$ starts to look more like $\hat{\mu}_l^{SD}$ as opposed to $\hat{\mu}_l^{SI}$. I would be comfortable with

- $\tau = 5$ for mean adaptation
- $\tau = 25$ for variance adaptation

MAP adaptation

MAP adaptation can only effect states with adaptation data

- If $\hat{n}_l^{SD} = 0$, then $\hat{\mu}_l^{MAP} = \hat{\mu}_l^{SI}$

When does MAP adaptation under-perform?

- Small amounts of adaptation data
- Unsupervised adaptation

When does MAP adaptation excel?

- Large amounts of adaptation data
- Supervised adaptation

MAP wrap-up

In practice we empirically "validate" our beliefs about τ

One can "derive" the MAP estimate using conjugate priors (Gauvain and Lee 1993)

MAP adaptation is a somewhat misleading name for this procedure

MLLR (maximum likelihood linear regression)

We use a weighted linear regression model to predict $\{\hat{\mu}_l^{SD}\}_{l=1}^{L}$ from the empirical priors $\{\hat{\mu}_l^{SI}\}_{l=1}^{L}$

$$\hat{\mu}_l^{SD} = a_0 + a_1 \hat{\mu}_l^{SI} + \epsilon_l$$

Where the errors are distributed

$$\epsilon_l \overset{i.i.d}{\sim} N(0, \frac{(\hat{\sigma}_l^{SI})^2}{\hat{n}_l^{SD}} \sigma^2)$$

Thus, we assume the variance in the error, ϵ_l, has two factors

- A uniform (unknown) variance: σ^2
- A (known) state specific weight: $(\hat{\sigma}_l^{SI})^2 / \hat{n}_l^{SD}$

MLLR (cont'd)

I am ignoring a minor technicality about states with $\hat{n}_l^{SD} = 0$

The form of the state specific weight, $(\hat{\sigma}_l^{SI})^2/\hat{n}_l^{SD}$, means the model is influenced more by states with

- A small speaker independent variance, $(\hat{\sigma}_l^{SI})^2$
- A large speaker dependent count $\hat{n}_l^{SD}$

To estimate $a = (a_0, a_1)^t$ we use weighted least squares, i.e., we minimize the weighted residual sum of squares error

$$\text{WRSS}(a) = \sum_{l=1}^{L} \frac{(\hat{\mu}_l^{SD} - a_0 - a_1\hat{\mu}_l^{SI})^2}{(\hat{\sigma}_l^{SI})^2/\hat{n}_l^{SD}} = \sum_{l=1}^{L} \hat{n}_l^{SD} \left(\frac{\hat{\mu}_l^{SD} - a_0 - a_1\hat{\mu}_l^{SI}}{\hat{\sigma}_l^{SI}} \right)^2$$

Relationship to the original formulation (Leggetter and Woodland 1994)

In the original formulation, a is chosen to maximize the log-likelihood of the speaker dependent data (here and below the C_i do not depend on a):

$$\mathrm{LL}(a) = -\frac{1}{2}\sum_{l=1}^{L}\sum_{t=1}^{N} p(q_l^t \mid x, \theta^{SI}) \left(\frac{x_t - a_0 - a_1\hat{\mu}_l^{SI}}{\hat{\sigma}_l^{SI}}\right)^2 + C_1$$

It is easy to show that these two formulations are the same:

$$-\frac{1}{2}\,\mathrm{WRSS}(a) = \mathrm{LL}(a) - C_2$$

The weighted least squares solution

We introduce three matrices

$$Z = \begin{pmatrix} \hat{\mu}_1^{SD} \\ \hat{\mu}_2^{SD} \\ \vdots \\ \hat{\mu}_I^{SD} \end{pmatrix}, Y = \begin{pmatrix} 1 & \hat{\mu}_1^{SI} \\ 1 & \hat{\mu}_2^{SI} \\ \vdots & \\ 1 & \hat{\mu}_I^{SI} \end{pmatrix}, E = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_I \end{pmatrix}$$

Then the model can be written in this form

$$Z = Ya + E$$

We use least squares because this is an inconsistent system ($L > 2$)

The weighted least squares solution (cont'd)

To un-weight the problem we introduce $\delta_l \overset{i.i.d}{\sim} N(0, \sigma^2)$,

$$D = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_l \end{pmatrix}, \text{ and } W = \begin{pmatrix} \hat{n}_1^{SD}/(\hat{\sigma}_1^{SI})^2 & 0 & \dots & 0 \\ 0 & \hat{n}_2^{SD}/(\hat{\sigma}_2^{SI})^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{n}_l^{SD}/(\hat{\sigma}_L^{SI})^2 \end{pmatrix}$$

The equivalent, un-weighted model is

$$W^{\frac{1}{2}} Z = W^{\frac{1}{2}} Ya + D$$

The weighted least squares solution (cont'd)

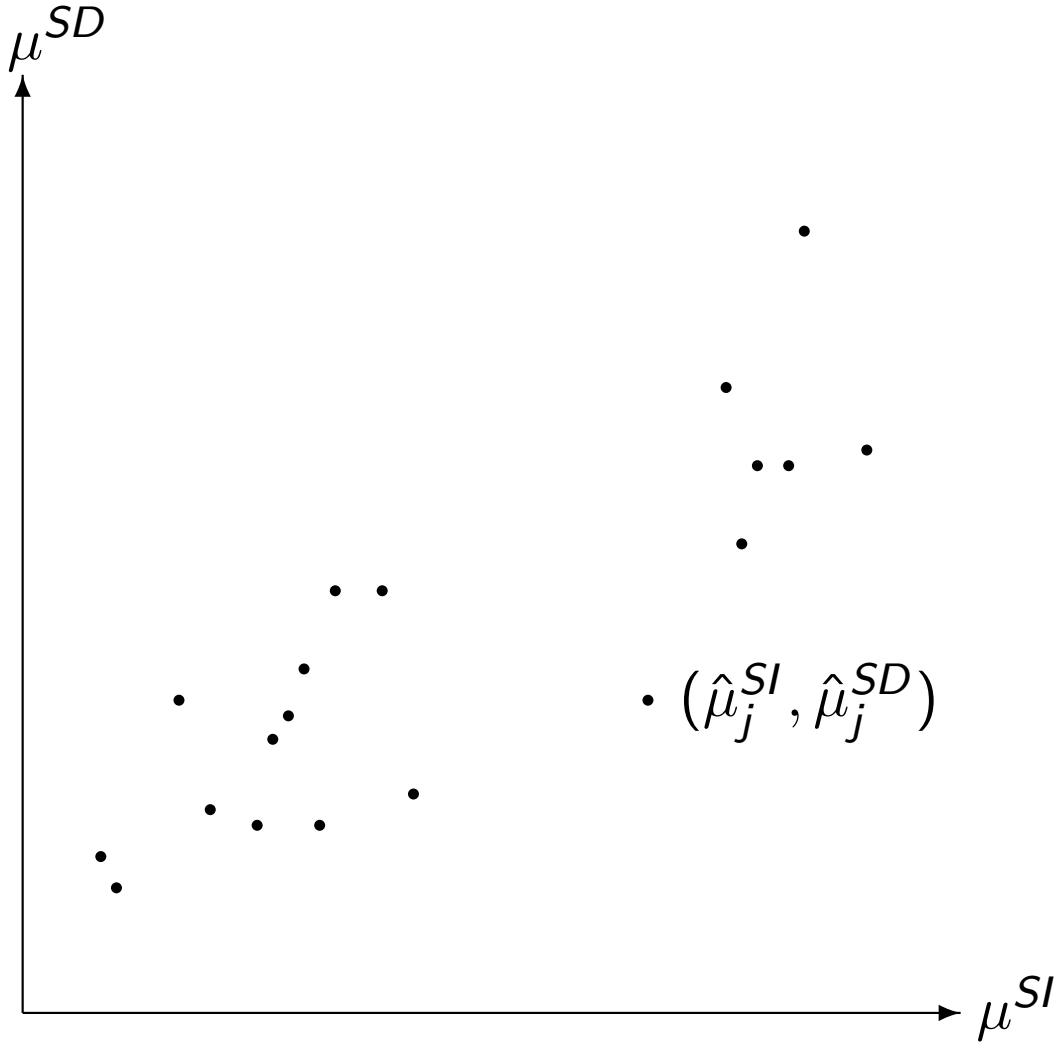
The least squares estimate for a is

$$\hat{a} = \left(Y^t W Y\right)^{-1} Y^t W Z$$

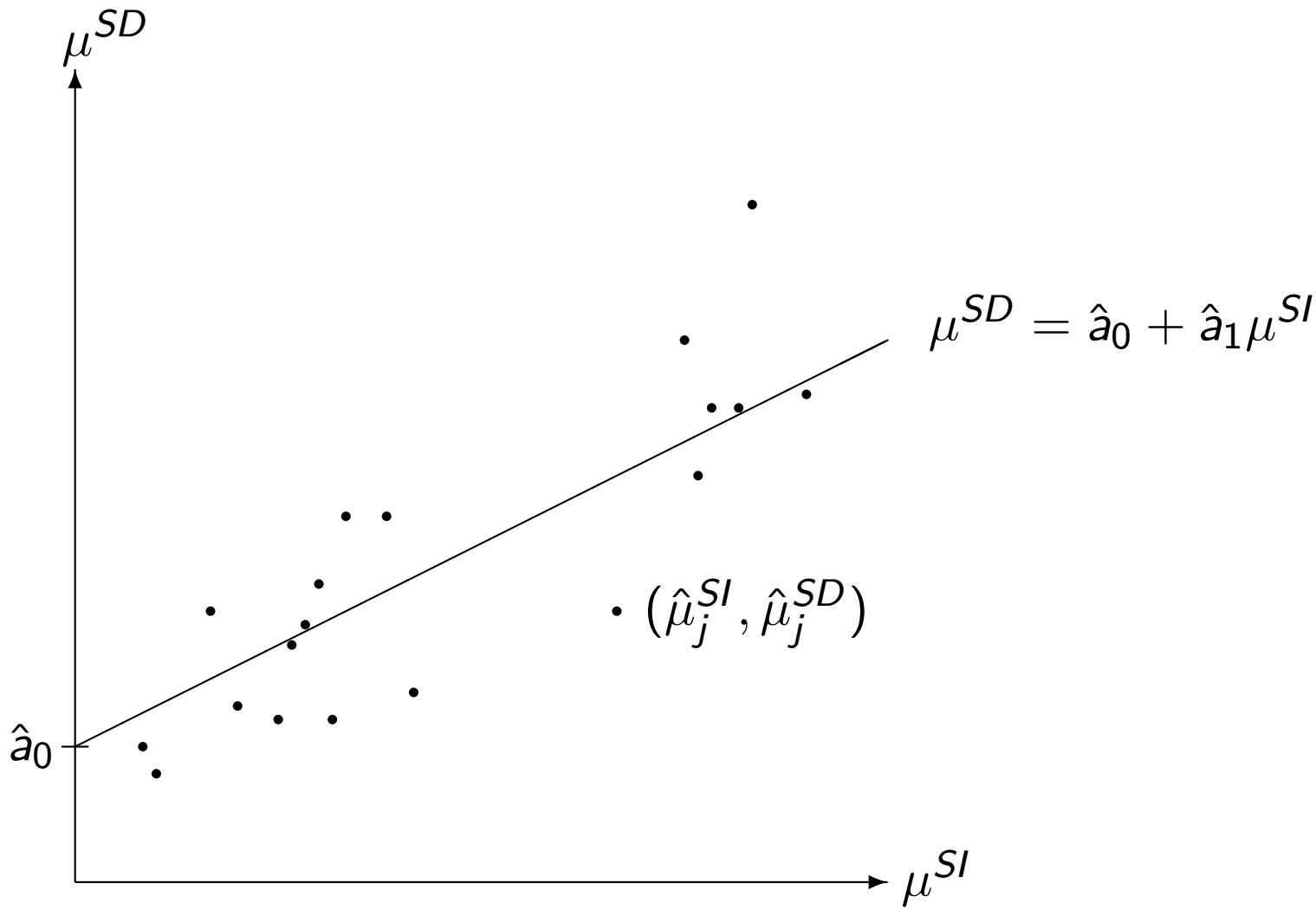
Finally, the MLLR estimates for the means are given by

$$\hat{\mu}_l^{MLLR} = \hat{a}_0 + \hat{a}_1 \hat{\mu}_l^{SI}$$

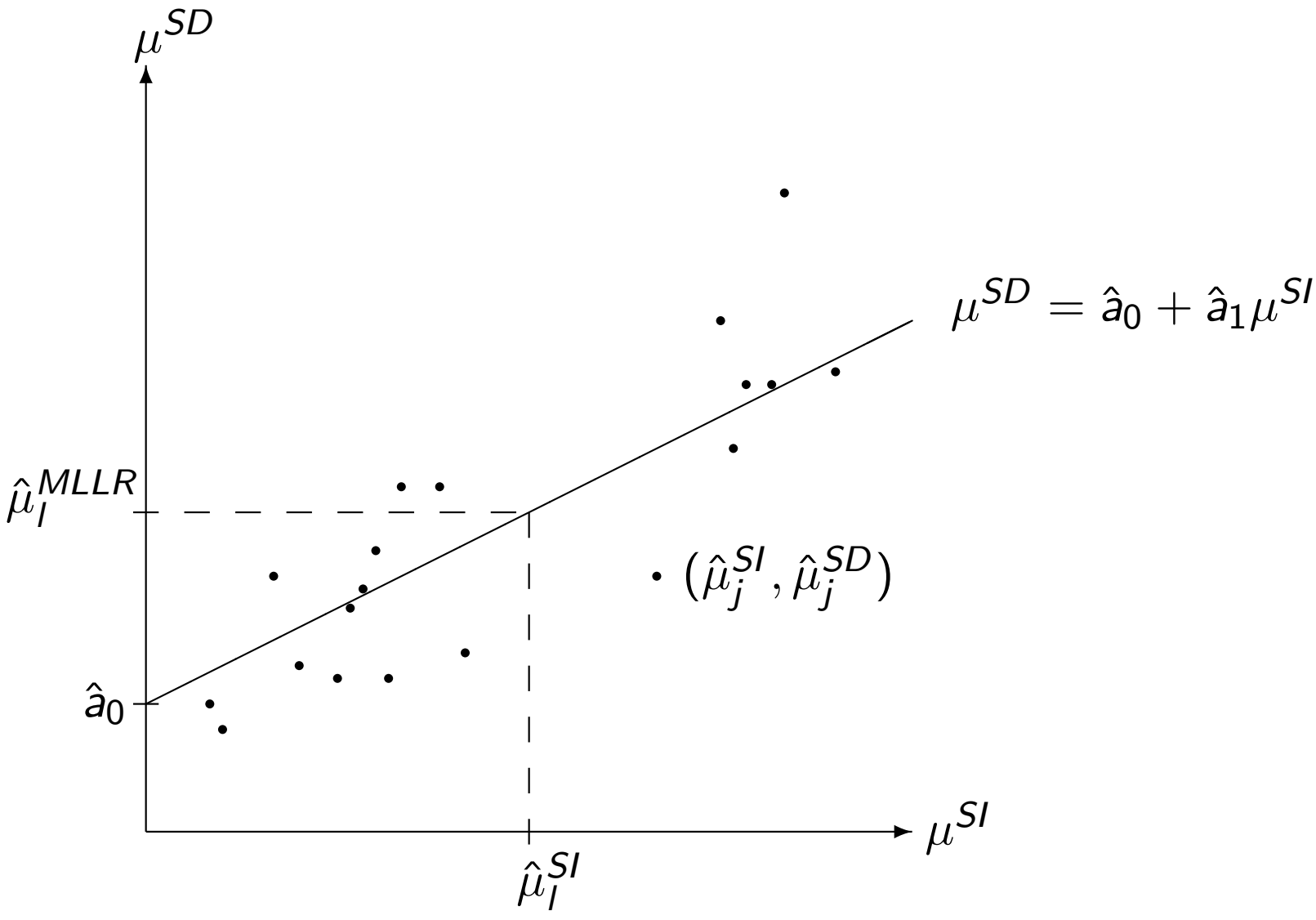
MLLR step 1: gather the SI and SD data



MLLR step 2: do the least squares fit



MLLR step 3: use the regression to compute $\hat{\mu}_l^{MLLR}$



MLLR vs MAP

All of the means are adjusted by the MLLR transform

- Even states where $\hat{n}_l^{SD} = 0$

The estimates $\{\hat{\mu}_l^{MLLR}\}_{l=1}^{L}$ are influenced most by data from states l where $\hat{n}_l^{SD}/(\hat{\sigma}_l^{SI})^2$ is large

- This follows from the weighted least squares formulation

MLLR outperforms MAP

- Small amounts of data
- Unsupervised adaptation

MAP outperforms MLLR

- Large amounts of data

MLLR wrap-up

The MLLR framework allows for multiple transformations

- Groups of states (components) are given separate transforms
- This grouping can be done by hand (e.g. by phoneme groups) or by automatic clustering
- Number of transforms is a function of N

MLLR in the d-dimensional case is a straightforward generalization

- There are d weighted regressions

"Maximum likelihood linear regression" is a peculiar name

- Least squares is the maximum likelihood solution to the linear regression problem!

Intro to discriminative training

Earlier we showed how to estimate a HMM's parameters using maximum likelihood

- Via the Baum-Welch algorithm

Maximum likelihood estimation is asymptotically optimal in most situations

- Baum-Welch also has good asymptotic properties

Why consider other estimation methods?

- What if the model is wrong!

Motivation (cont'd)

When the model doesn't fit the data, you can do better than the MLE

In the case of speech recognition there are (at least) two successful alternatives to the MLE

- Maximum mutual information (MMI)
- Minimum phone error (MPE)

Both of these estimation methods use model selection criteria

- That are more close related to the recognition problem than maximum likelihood
- That are "discriminative" in nature

Recognition reminder

Given a utterance X, we select M^{recog} via:

$$M^{recog} = \arg\max_{M} P(M \mid X)$$

We do not model $P(M \mid X)$, instead we use Bayes' Rule

$$P(M \mid X) = \frac{P(X \mid M)P(M)}{P(X)}$$

This decomposes the problem into two probability models

- The *acoustic model* gives the likelihood $P(X \mid M)$
- The *language model* gives the prior $P(M)$

Generative vs Discriminative classifiers

What we've just described is an example of a *generative* classifier

- Model $P(X \mid M)$ separately for each class M
- X is random
- Stronger model assumptions
- Uses maximum likelihood estimation
- Estimation is "easy"

A *discriminative* classifier models $P(M \mid X)$

- Model the class probabilities $P(M \mid X)$ directly
- M is random
- Weaker model assumptions
- Uses conditional likelihood estimation
- Estimation is "hard"

Generative vs Discriminative classifier (cont'd)

	Generative	Discriminative
Model	$P(X \mid M)$	$P(M \mid X)$
Estimation	MLE, "easy"	CMLE, "hard"
Model assumptions	Stronger	Weaker
Advantages	More efficient when model is correct (uses $P(X)$)	More robust, fewer assumptions
Disadvantages	IRL model is rarely correct	Ignores $P(X)$

Discriminative classifiers

Model the class boundaries or membership probabilities directly

- Logistic regression
- Neural networks
- Support vector machines

Requires simultaneous consideration of all classes—including correct

- In contrast to generative: just the correct class
- Makes the training task much harder

Brief technical interlude about recognition

We scale the acoustic model by a factor $1/\kappa$

- Mostly because of between/within frame correlation
- Choice of κ is made via 'tuning' to minimize errors

So recognition actually uses

$$M^{recog} = \arg\max_{M} P(X \mid M, \Theta)^{\frac{1}{\kappa}} P(M)$$

Weighted version of $P(M \mid X, \Theta)$:

$$P^{\kappa}(M \mid X, \Theta) \equiv \frac{P(X \mid M, \Theta)^{\frac{1}{\kappa}} P(M)}{\sum_{j=1}^{J} P(X \mid M_j, \Theta)^{\frac{1}{\kappa}} P(M_j)}$$

Brief technical interlude (cont'd)

Recognition problem becomes

$$M^{recog} = \arg\max_{M} P^{\kappa}(M \mid X, \Theta)$$

$A(M, M_{ref})$ is the phone accuracy of M relative to M_{ref}

- Convert both M and M_{ref} to a phone string using a dictionary
- Technicalities involving time boundaries

Three model selection criteria

ML: likelihood of the training data

$$\mathcal{F}_{ML}(\Theta) = P(X \mid M_{ref}, \Theta)$$

MMI: conditional likelihood of the training data

$$\mathcal{F}_{MMI}(\Theta) = P^{\kappa}(M_{ref} \mid X, \Theta)$$

MPE: expected phone accuracy on the training data

$$\mathcal{F}_{MPE}(\Theta) = \sum_{j=1}^{J} P^{\kappa}(M_j \mid X, \Theta) A(M_j, M_{ref})$$

Model estimation (training) using these criteria

These are simply different model selection/estimation criteria

- We don't change the structure of the HMM

Each criterion has its own estimation algorithm

- ML uses the Baum-Welch algorithm
- MMI/MPE use a variant called *extended Baum-Welch*

Maximum likelihood

Model selection criterion:

$$\mathcal{F}_{ML}(\Theta) = P(X \mid M_{ref}, \Theta)$$

Model estimation: maximizes training data likelihood

$$\hat{\Theta}_{ML} = \arg\max_{\Theta} \mathcal{F}_{ML}(\Theta)$$

Maximum mutual information

Model selection criterion:

$$\mathcal{F}_{MMI}(\Theta) = P^{\kappa}(M_{ref} \mid X, \Theta)$$

$\mathcal{F}_{MMI}$ is intuitively related to recognition accuracy

Model estimation: maximizes training data conditional likelihood

$$\hat{\Theta}_{MMI} = \arg\max_{\Theta} \mathcal{F}_{MMI}(\Theta)$$

This is conditional likelihood estimation

- Equivalent (original) formulation: mutual information

Minimum phone error

Model selection criterion:

$$\mathcal{F}_{MPE}(\Theta) = \sum_{j=1}^{J} P^{\kappa}(M_j \mid X, \Theta) A(M_j, M_{ref})$$

$\mathcal{F}_{MPE}$ is intuitively related to recognition accuracy

MPE: maximizes expected phone accuracy on the training data

$$\hat{\Theta}_{MPE} = \arg\max_{\Theta} \mathcal{F}_{MPE}(\Theta)$$

Perhaps a better name: maximum phone accuracy!

Parameter estimation using MMI: introduction

We choose Θ to maximize

$$\mathcal{F}_{MMI}(\Theta) = \frac{P(X \mid M_{ref}, \Theta)^{\frac{1}{\kappa}} P(M_{ref})}{\sum_{j=1}^{J} P(X \mid M_j, \Theta)^{\frac{1}{\kappa}} P(M_j)}$$

The denominator term is key to estimation with MMI

- Maximum likelihood ignored it

Parameter estimation using MMI: introduction (cont'd)

We expand the denominator

$$\mathcal{F}_{MMI}(\Theta) = \frac{P(X \mid M_{ref}, \Theta)^{\frac{1}{\kappa}} P(M_{ref})}{P(X \mid M_{ref}, \Theta)^{\frac{1}{\kappa}} P(M_{ref}) + \sum_{M \neq M_{ref}} P(X \mid M, \Theta)^{\frac{1}{\kappa}} P(M)}$$

Roughly speaking, large $\mathcal{F}_{MMI}(\Theta)$ (say $= 1$) means that for every imposter $M \neq M_{ref}$

$$P(X \mid M_{ref}, \Theta)^{\frac{1}{\kappa}} P(M_{ref}) > P(X \mid M, \Theta)^{\frac{1}{\kappa}} P(M)$$

This would give perfect recognition on the training data!

Parameter estimation using MMI: extended BW

Extended BW training combines two separate BW estimations

- The numerator: $P(X \mid M_{ref}, \Theta)^{\frac{1}{\kappa}} P(M_{ref})$
- The denominator: $\sum_{j=1}^{J} P(X \mid M_j, \Theta)^{\frac{1}{\kappa}} P(M_j)$

The numerator BW is (essentially) the usual algorithm

For the denominator we would like to run J BWs

- One BW for each term $P(X \mid M_j, \Theta) P(M_j)$
- Then combine somehow

Parameter estimation using MMI: extended BW (cont'd)

The problem is that J can be extremely large (∞!)

We make an approximation by summing over a subset $\{M_k\}_{k=1}^{K}$

- $K \ll J$
- Obtained by K-best recognition on the training data
- This recognition uses $\hat{\Theta}_{ML}$
- Choosing the recognition language model is tricky

Parameter estimation using MMI: extended BW (cont'd)

The actual procedure uses the framework of *lattices*

- An efficient way to store the K-best information
- Word and phone level start and end times

The forward-backward algorithm has been extended to this lattice-based framework

- Including the numerator

We will omit the details, see

- Gold-Morgan-Ellis, Chapter 28
- Dan Povey's Ph.D. thesis

Parameter estimation using MMI: update formula inputs

Each BW produces a set of accumulators

- Numerator (correct): $\{\mu_l^{num}, n_l^{num}\}_{l=1}^{L}$
- Denominator (impostors) :$\{\mu_l^{den}, n_l^{den}\}_{l=1}^{L}$

The previous value of the mean, μ_l

- At the start $\mu_l = \hat{\mu}_l^{MLE}$

A state specific smoothing constant, D_l

- $D_l = E \times n_l^{den}$
- E is tunable, usually $1 \leq E \leq 2$
- So $D_l \geq n_l^{den}$

Parameter estimation using MMI: mean update formula

MMI estimate

$$\hat{\mu}_l = \frac{n_l^{num}\mu_l^{num} - n_l^{den}\mu_l^{den} + D_l\mu_l}{n_l^{num} - n_l^{den} + D_l}.$$

To get to $\hat{\mu}_l$ from μ_l we move

- Towards to centroid of the correct data (numerator)
- Away from the centroid of the imposter data (denominator)

MPE uses a slight variation on this formula

- An additional smoothing term with $\hat{\mu}_l^{MLE}$
- However, the counts are now related to phone accuracy

Discriminative training wrap-up

MMI and MPE have resulted in impressive gains in recognition accuracy

- It took many years of research to work out the current, successful formalism

MMI/MPE only work because the HMM model doesn't fit the data

- What model assumptions are at fault?
- Maybe we should look for a better model!

Promising, recent research using hybrid HMM/neural networks

- Builds on earlier work (e.g. by Morgan)
- Uses deep belief networks