NIST 2006
Speaker Recognition Workshop

A collaborative journey

Jason Pelecanos, Jiri Navratil
and Ganesh N Ramaswamy

Conversational Biometrics and Multimedia Mining Group
IBM T. J. Watson Research Center
Yorktown Heights, New York
http://www.research.ibm.com/CBG

Overview

- Goals for NIST 2006
- Overview of Contributions
- Binary Trees
- Support Vector Machines
- Results Overview
- Conclusions

Goals for NIST 2006

Goals for NIST 2006

- Supply collaborative partners with speaker recognition statistics that provide complementary information
- These statistics may be in the form of:
  - Speaker recognition utterance pair scores
  - Utterance side information
- Demonstrate improvements that are attributed to the inclusion of such statistics

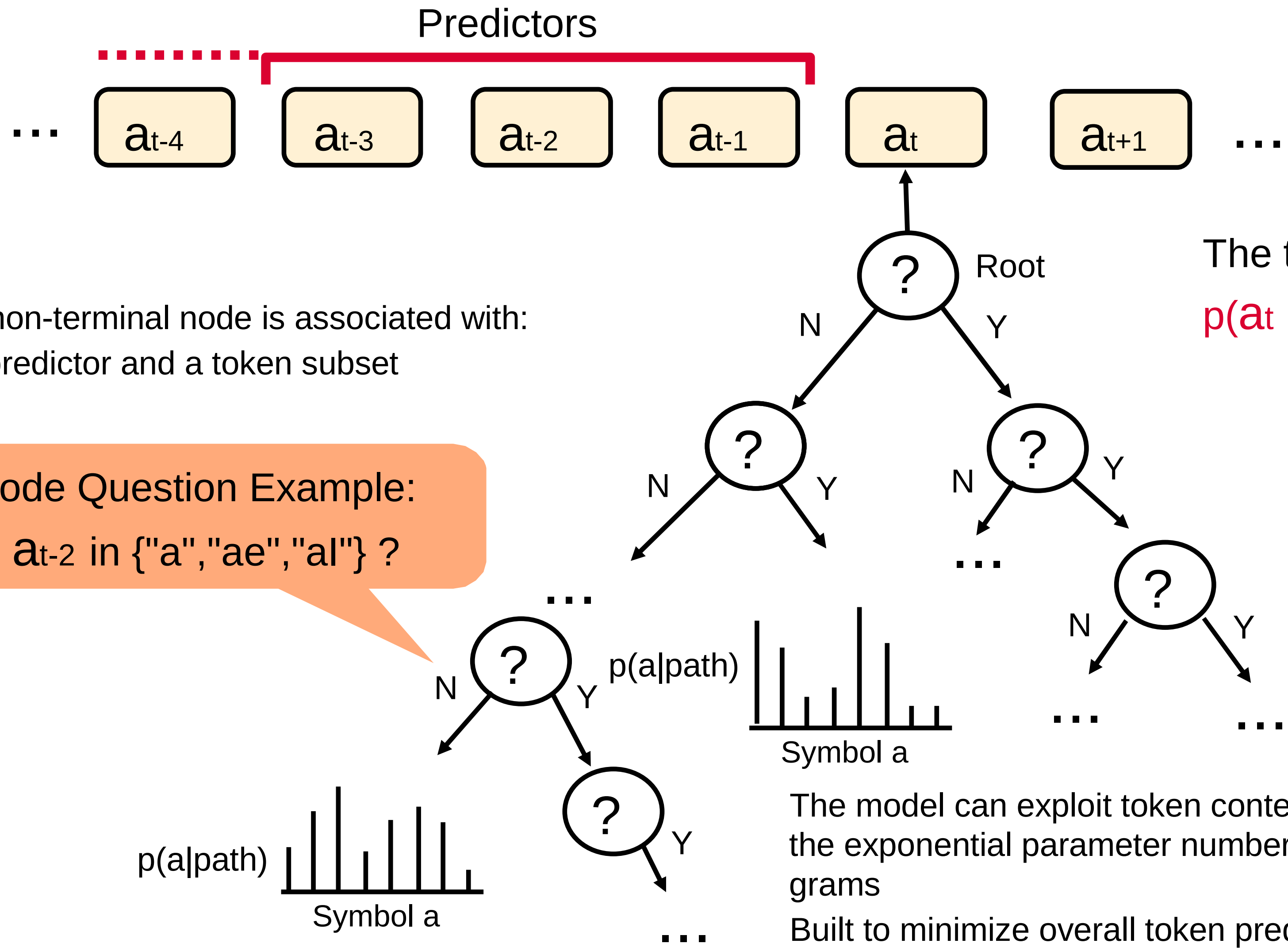
Overview of Contributions

Contributions to MIT and QUT

- QUT
  - Binary tree phonetic N-gram statistics
  - GIX sequences
  - SVM results
  - Handset labels for the fusion component of the QUT system
- MIT
  - Word level N-gram statistics using Binary trees

Binary Trees

Binary Tree Modeling



The tree returns

$p(a_t \mid f(\text{predictors}))$

A non-terminal node is associated with:
a predictor and a token subset

Node Question Example:

Is a_{t-2} in {"a","ae","al"} ?

The model can exploit token context of arbitrary length, without the exponential parameter number growth connected with N-grams

Built to minimize overall token prediction entropy

Effective adaptation and smoothing

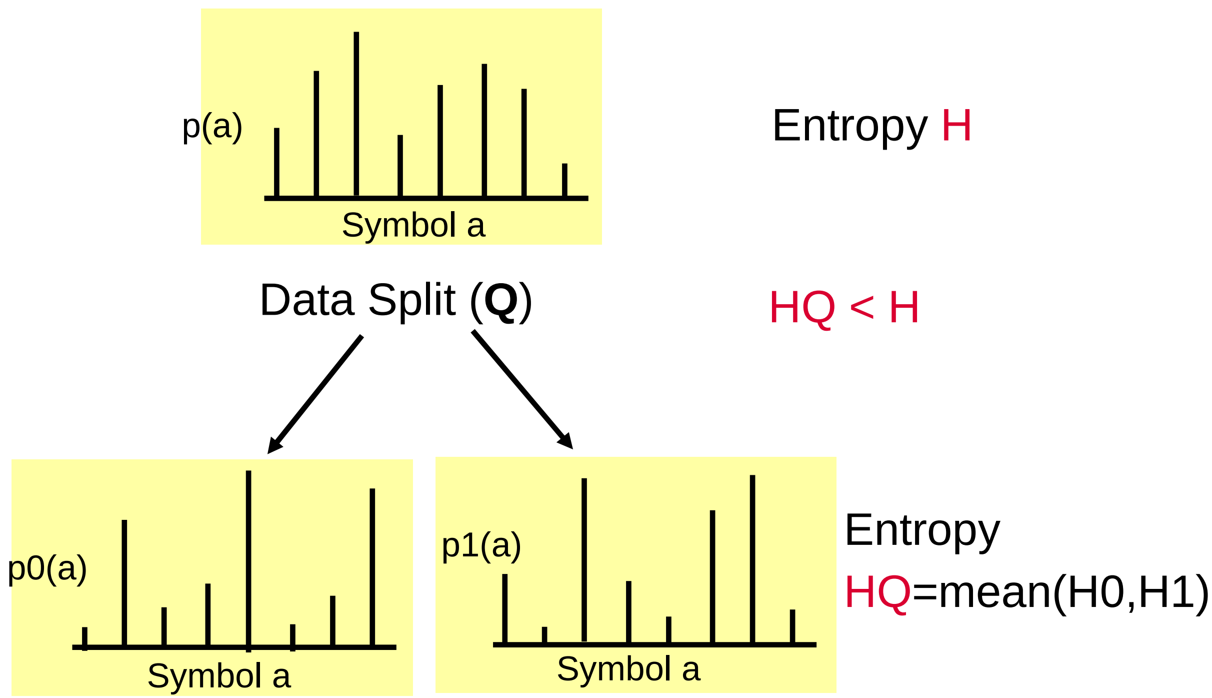
Growing Good Trees - I

Find a structure that minimizes overall prediction entropy (minimizing “node impurity” [Breiman – CART])

<u>Recursive tree growing algorithm</u>

At each node: Find question **Q**,

s.t. H - HQ > r (r significance threshold)



<u>Practice:</u>

<u>Occupancy constraints:</u> Minimum N data count in each candidate split

<u>Cross evaluation:</u> Entropy reduction R=H-HQ computed on a held-out set

Growing Good Trees - II

Minimum prediction entropy on training data = maximum likelihood of the training data

$$\overline{H} = \sum_{l} P_l \cdot H_l$$

$$H_l = -\sum_{s_i \in \mathcal{A}} P_l(s_i) \log_2 P_l(s_i)$$

$$\hat{P}_l(s_i) = \frac{\#(s_i|\alpha_l)}{|\alpha_l|}$$

$$\hat{P}_l = \frac{|\alpha_l|}{\sum_{l=1}^{L} |\alpha_l|}$$

$$\boxed{\mathcal{L} = -\widehat{\overline{H}_l}}$$

$$\begin{aligned} \mathcal{L} &= \frac{1}{T}\sum_{t=1}^{T} \log_2 P(a_t|BT) \\ &= \sum_{l=1}^{L} \hat{P}_l \sum_{s_i \in \mathcal{A}} \hat{P}_l(s_i) \log_2 P_l(s_i) \end{aligned}$$

Growing Good Trees - III

Minimizing prediction entropy per iteration = maximizing mutual information between symbol distribution X and node question Q

$$R = H - H_Q =$$
$$= -p(c_1)H(S\,|\,c_1) - p(c_2)H(S\,|\,c_2) + H(S)$$
$$= \sum_{c\in\{1,2\}} \sum_{s\in A} p(c)\,p(s\,|\,c)\log\, p(s\,|\,c)\,/\log\, p(s)$$
$$= \sum_{c,s} p(s,c)\log\frac{p(s,c)}{p(s)\,p(c)}$$
$$= I(S,Q); \quad Q : A \mapsto \{0,1\}$$

Growing Good Trees - IV

1) Greedy Algorithm [Bahl et al. 1989]

Find **Q**: Is the value of predictor X_k in {symbol subset}

For all Predictors k=1,2,... :

1) Start with empty subset
2) Insert a symbol if H reduced (loop over all symbols)
3) Delete a symbol if H reduced (loop over all symbols)
4) Rep. 2-3 until convergence

(apply occupancy constraints)

Create two children if

H-HQ>r

Repeat recursively

2) Flip-Flop Approximation Algorithm [Nadas1991]

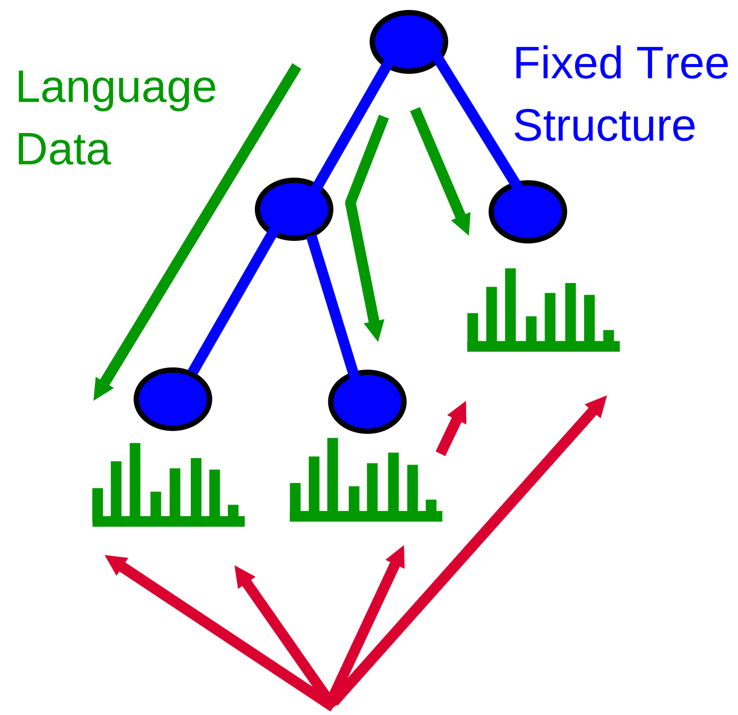
5-10x faster training – used for large symbol vocabularies

BTs (i.e. lexical and Gaussian index)

“Tree Helpers”

Tree-Model Adaptation

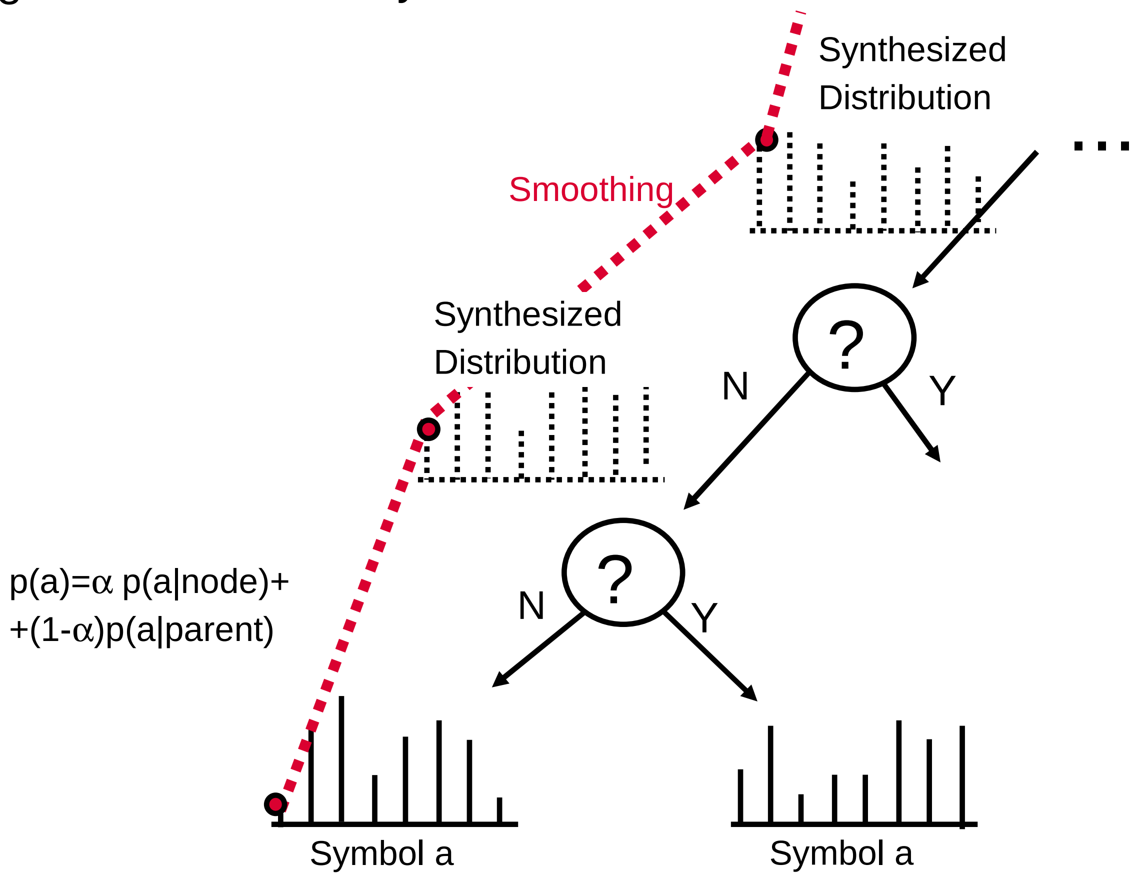
Use language training data to adapt leaves of an existing robust tree model (background, lang.-indep.)



Adapt leaves by interpolation

Recursive Bottom-Up Smoothing

Interpolate with parental node distributions recursively to increase observation mass



BT Components

- IBM/QUT collaboration

  phonetic BTs (12 decoders)

  Gaussian Index (GIX) BTs (size: 512)

  CT-normed BT score output
- IBM/MIT-LL collaboration

  ASR transcripts (size: top-512 most frequent words)

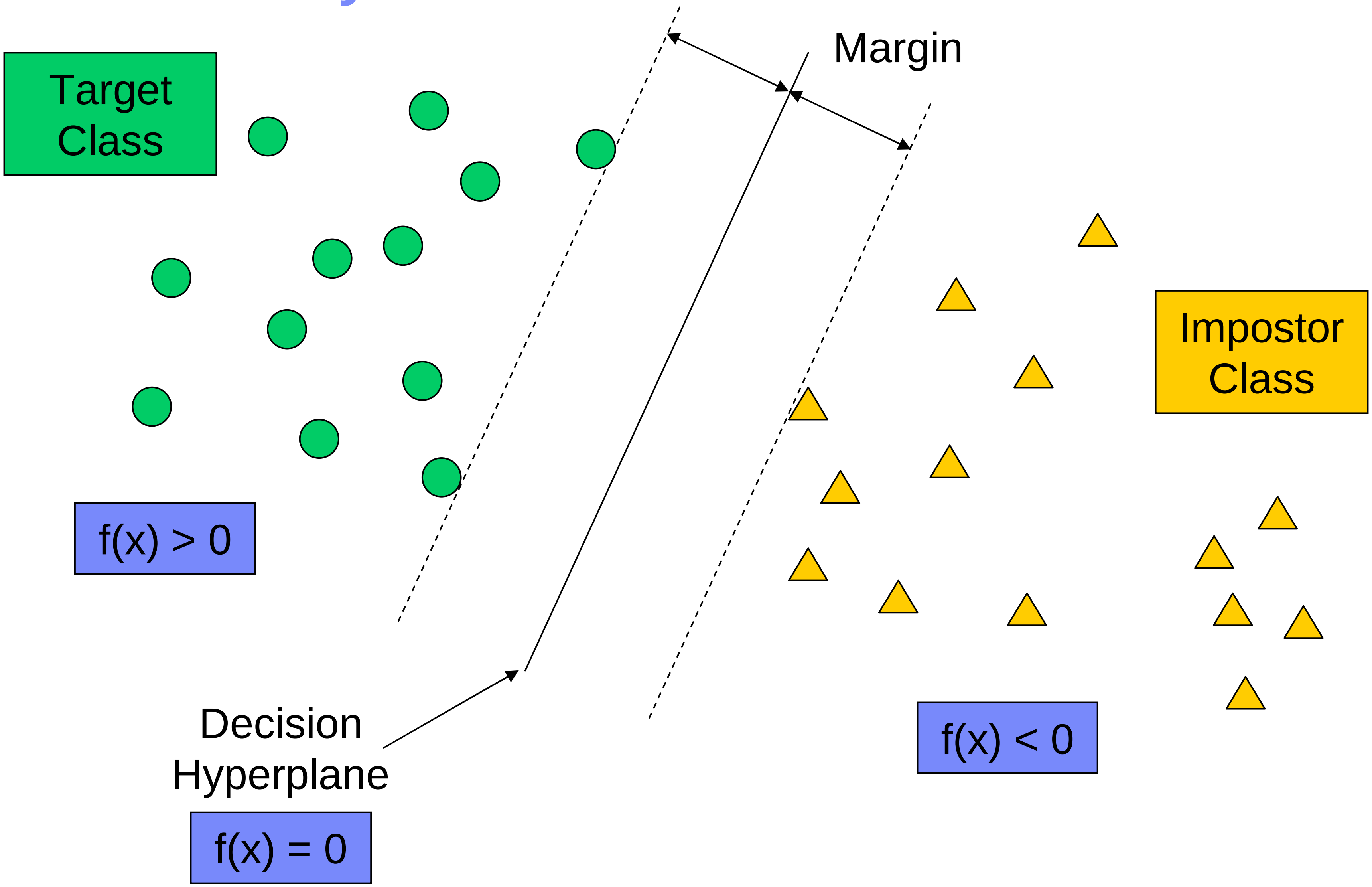
  CT-normed BT score output

Configuration

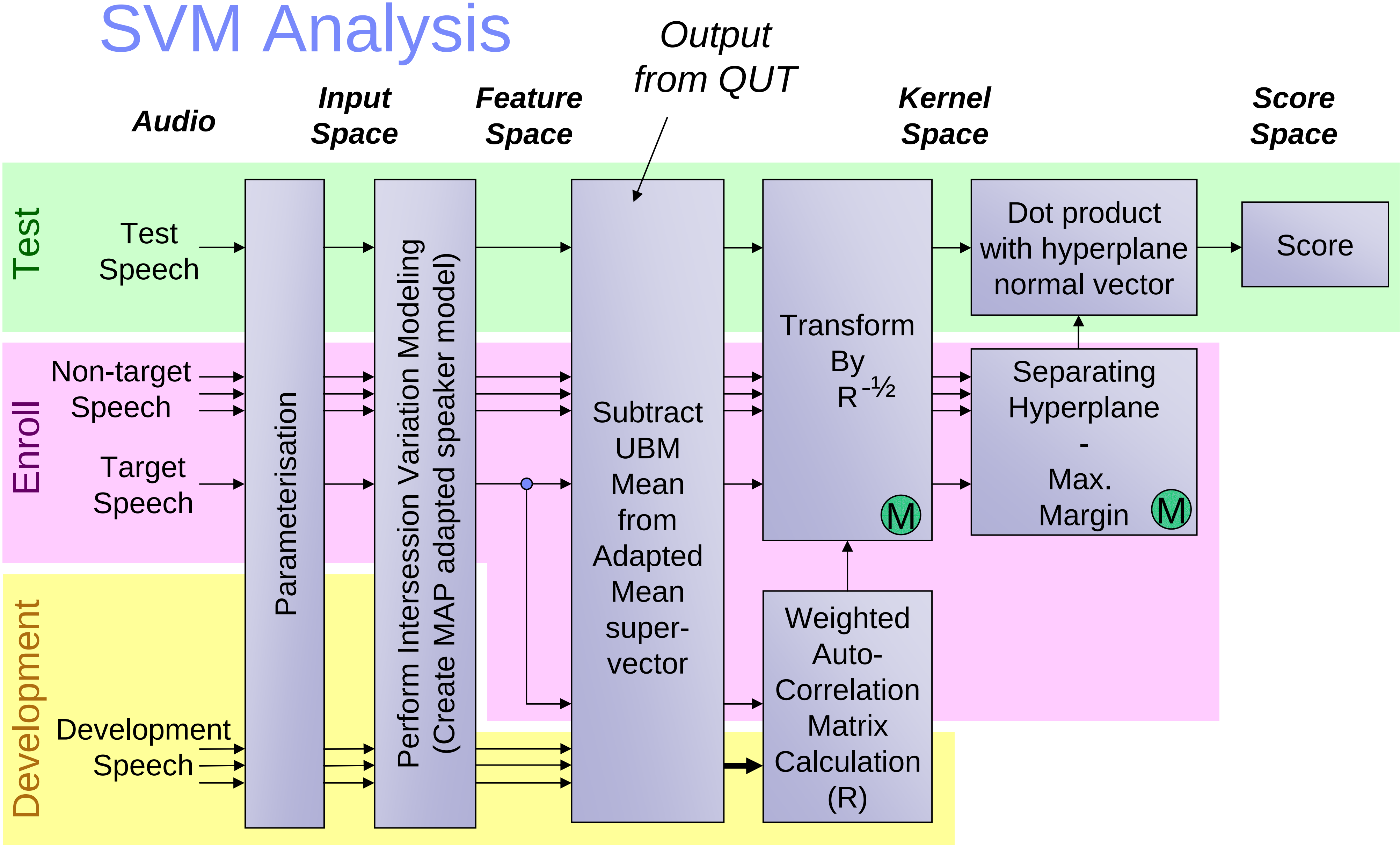
- Adaptation and smoothing used in all components
- The Greedy BT training algorithm used with phonetic sequences; the Flip-Flop algorithm with GIX and lexical features
- T-Norms (1,3,and 8 conv.) and C-Norms (1-conv. only) taken from the 2004 eval

Support Vector Machines

SVM Analysis



SVM Analysis



SVM Analysis

- SVM Kernel Evaluation (GLDS, Campbell)

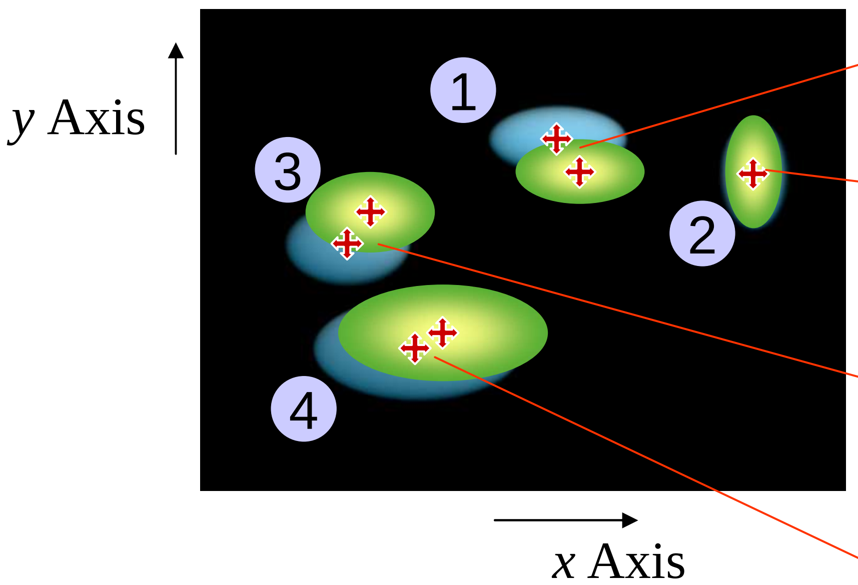
$$f(X) = \sum_{i=1}^{N} w_i c_i K(X, X_i) + d$$

$$K(X, X_i) = \bar{b}(X)' \boldsymbol{R}^{-1} \bar{b}(X_i)$$

- Where $\bar{b}(X)$ is the supervector created from the GMM component means…

Supervector Construction

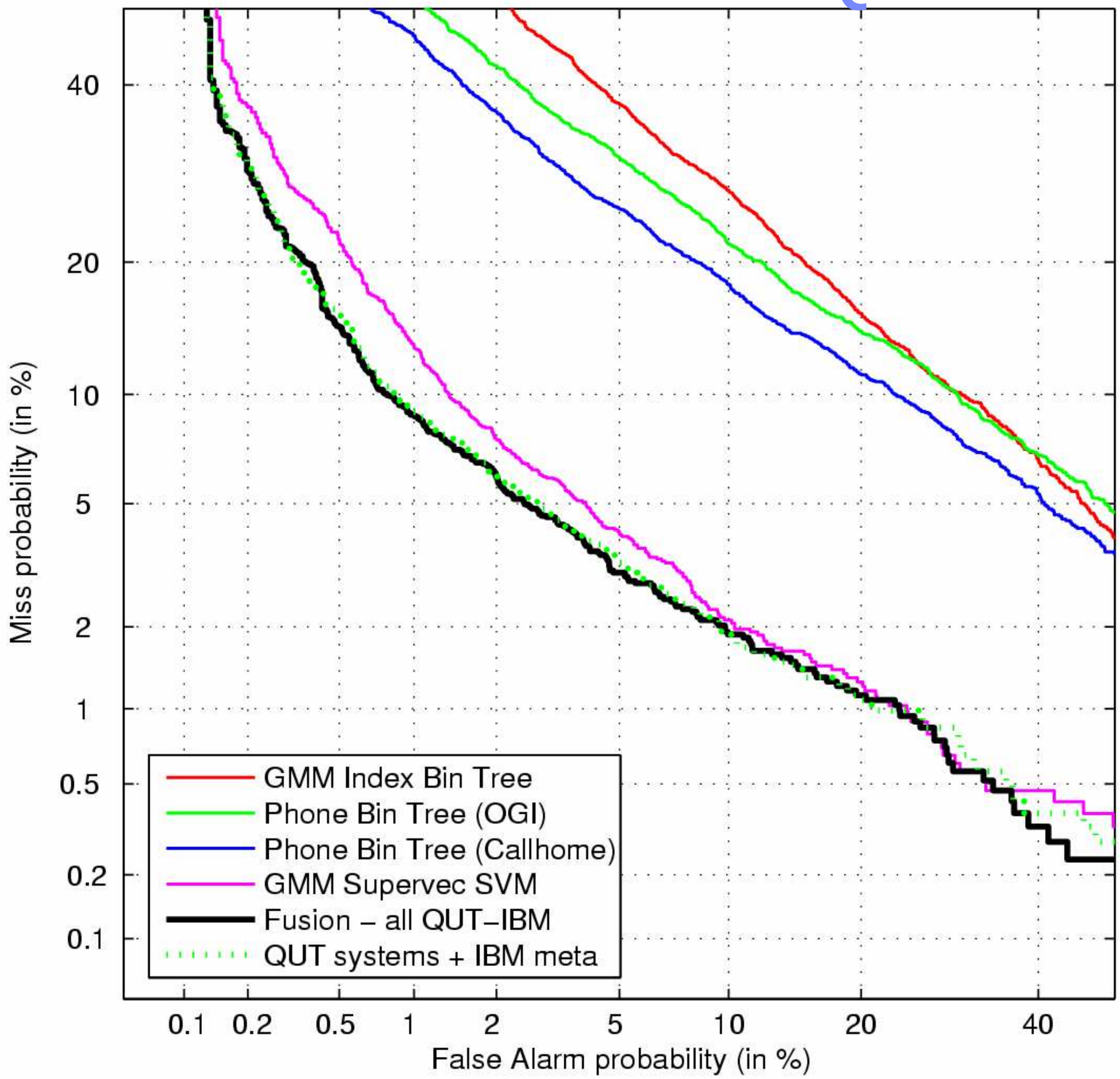
- The SVM feature space supervector is constructed from the concatenation of the ISV adapted and background Gaussian mean differences.



$$\bar{b}(X) = \begin{bmatrix} \Delta\mu_{x_1} \\ \Delta\mu_{y_1} \\ \Delta\mu_{x_2} \\ \Delta\mu_{y_2} \\ \Delta\mu_{x_3} \\ \Delta\mu_{y_3} \\ \Delta\mu_{x_4} \\ \Delta\mu_{y_4} \end{bmatrix}$$

Results Overview

IBM's contribution to the QUT/IBM System



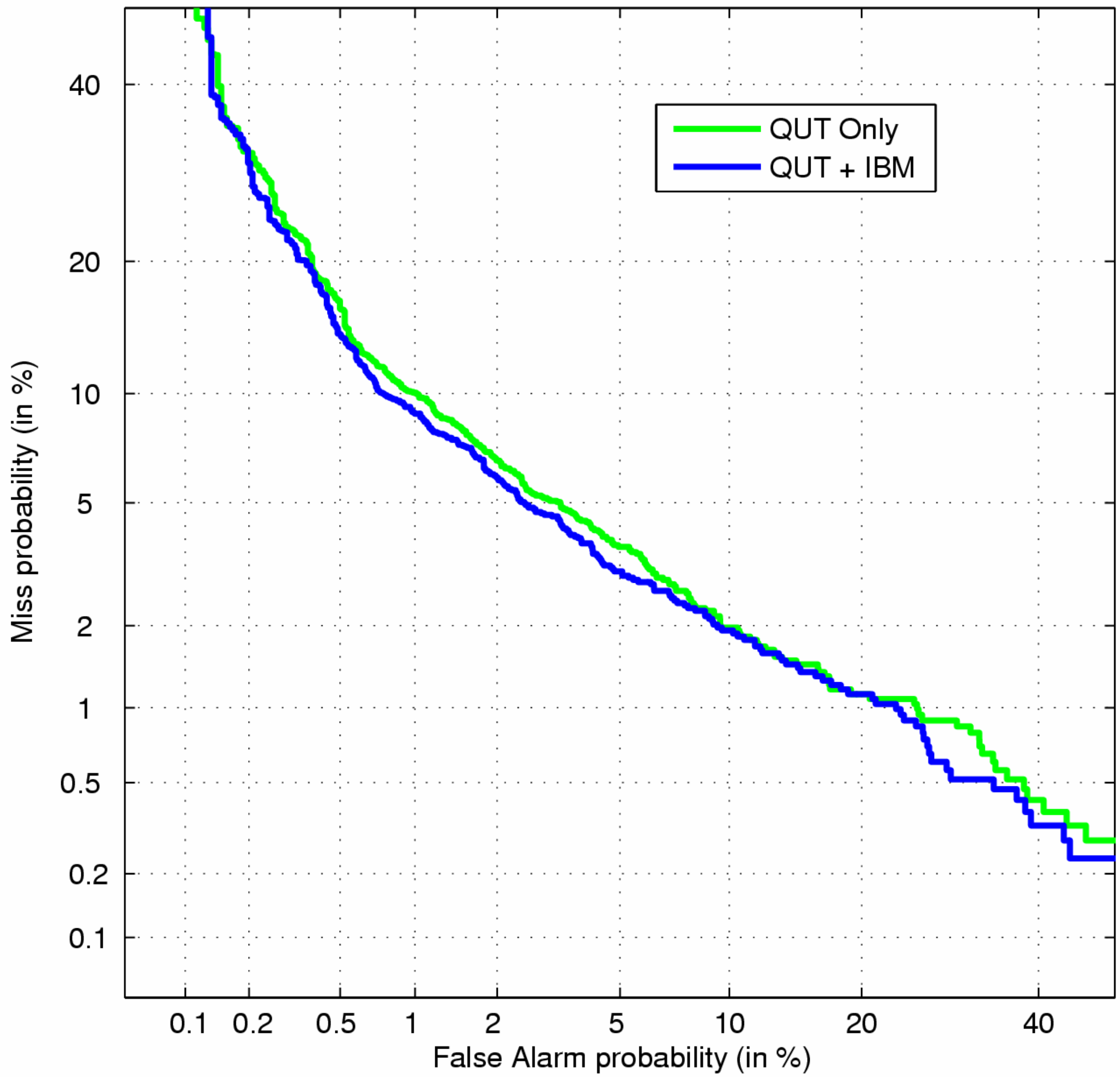
NIST 2006

--

1 session Training "English" only trials

Plot kindly supplied by QUT

IBM's contribution to the QUT/IBM System



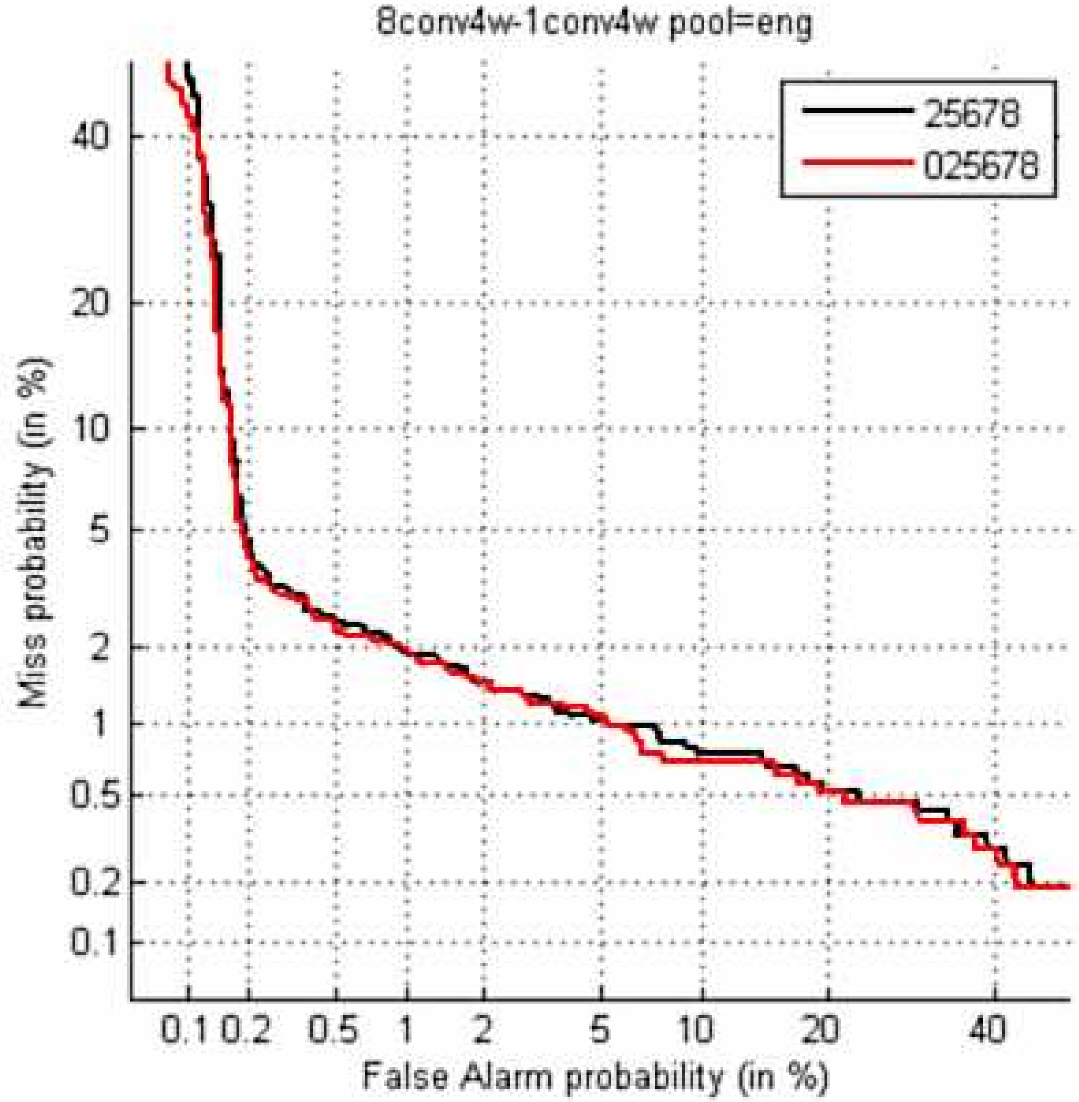
NIST 2006

--

1 session Training "English" only trials

Plot kindly supplied by QUT

IBM's contribution to the MIT/IBM System



NIST 2006

--

8 session Training "English" only trials

Plot kindly supplied by MIT

Conclusions

Conclusions

- Successfully added value to the systems of collaborating teams.
- Binary trees contributed to improving the overall system result on the NIST 2005 data
- SVMs using the ISV Gaussian Means are promising
- Handset type side information provided a useful addition

Acknowledgements

- IBM wishes to gratefully acknowledge the collaborative efforts of the MIT Lincoln Laboratory and the Queensland University of Technology

Questions

Additional Resources

BT References

- Bahl, L.R et al., “A tree-based statistical language model for natural language speech recognition,” IEEE Trans. On Acoustics, Speech, and Signal Processing, Vol. 37, No. 7, July 1989 (BTs in ASR)
- Nadas, A. et al., “An iterative flip-flop approximation of the most informative split in the construction of decision trees,” ICASSP 1991 (FF tree growing)
- Navratil, J. et al., “Phonetic speaker recognition using maximum-likelihood binary-decision tree models," ICASSP-03 (BTs in Speaker Recognition; smoothing and adaptation)
- Navratil, J., “Spoken language recognition - A step towards multilinguality in speech processing," IEEE Trans. on Speech and Audio Processing, Vol. 9, No. 6, September, 2001, pp. 678-85 (BTs in Language ID)
- Navratil, J., “Recent advances in phontoactic language recognition using binary-decision trees,” Interspeech 2006, to appear. (Flip-Flop algorithm evaluation)
- Buhrstein, D., et al. “Minimum impurity partitions,” The Annals of Statistics, Vol. 20, No. 3, 1992, pp. 1637-1646 (general results for BT optimization)
- Pelecanos, J. et al. “IBM SRE05 system,” presentation at the NIST SRE05 Workshop, May, 2005, Montreal, Canada.