Support Vector Machines for Speech Recognition

January 25th, 2002

Aravind Ganapathiraju

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University

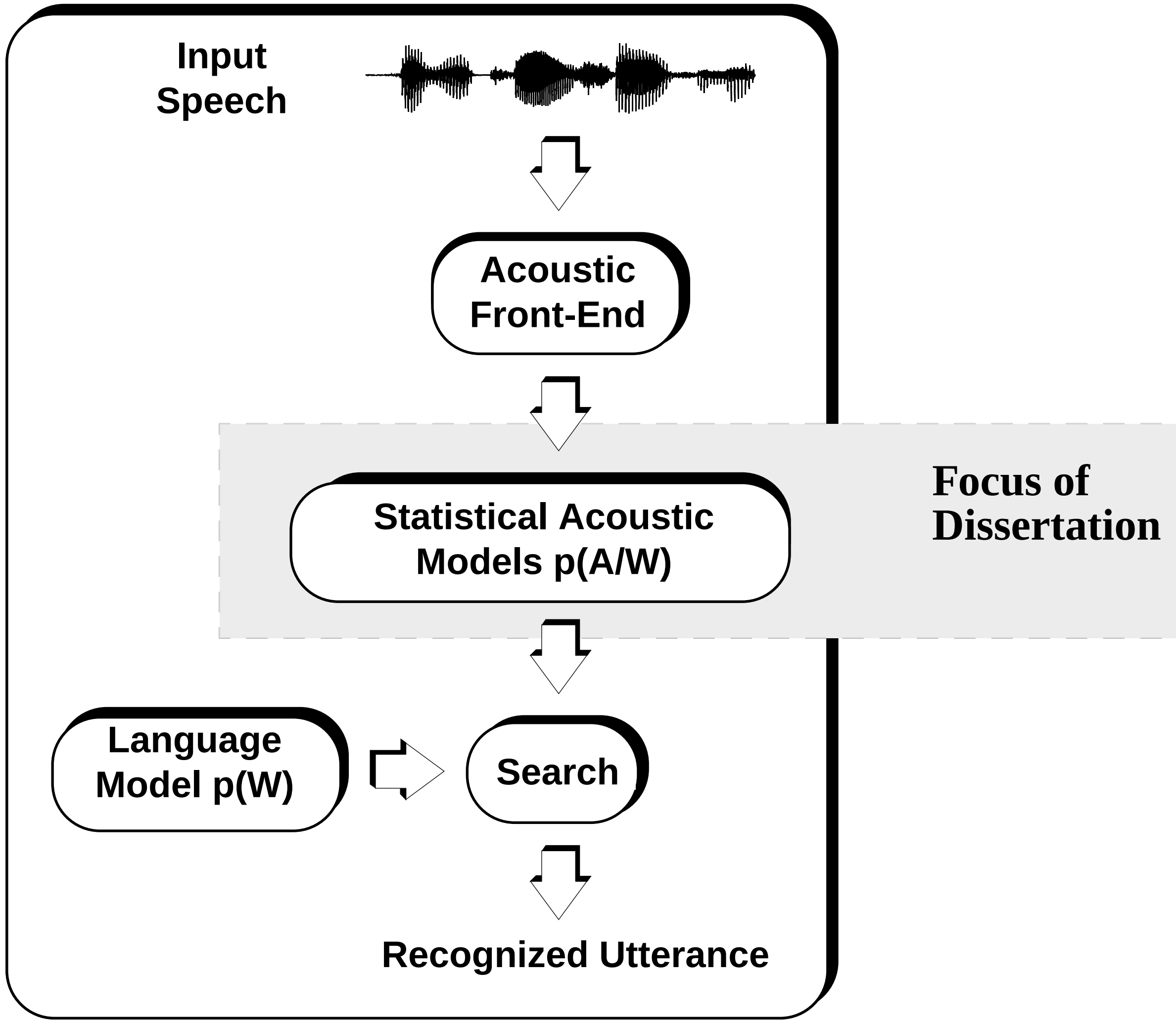
Organization of Presentation

* Motivation for using support vector machines (SVM)
* SVM theory and implementation
* Issues in using SVMs for speech recognition — hybrid recognition framework
* Experiments — data description and experimental results
* Error analysis and oracle experiments
* Summary and conclusions including dissertation contributions

Motivation

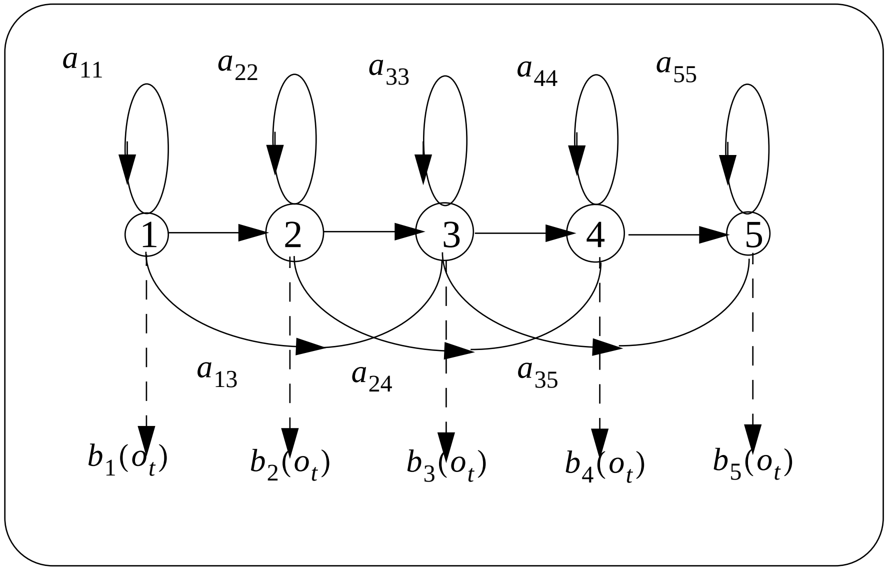
* Need discriminative techniques to enhance acoustic modeling
* Maximum Likelihood-based systems can be improved upon by discriminative machine learning techniques
* Support Vector Machines (SVM) have had significant success on several classification tasks
* Efficient estimation techniques now available for SVMs
* Study the feasibility of using SVMs as part of a full-fledged conversational speech recognition system

ASR Components



* Dissertation addresses acoustic modeling

Acoustic Modeling



- HMMs used is most state-of-the-art systems
- Maximum likelihood (ML) estimation dominant approach
- Expectation-maximization algorithm
- Hybrid Connectionist Systems — artificial neural networks (ANNs) used as probability estimators

SVM Success Stories

* SVMs have been used in several static classification tasks since the 1990's

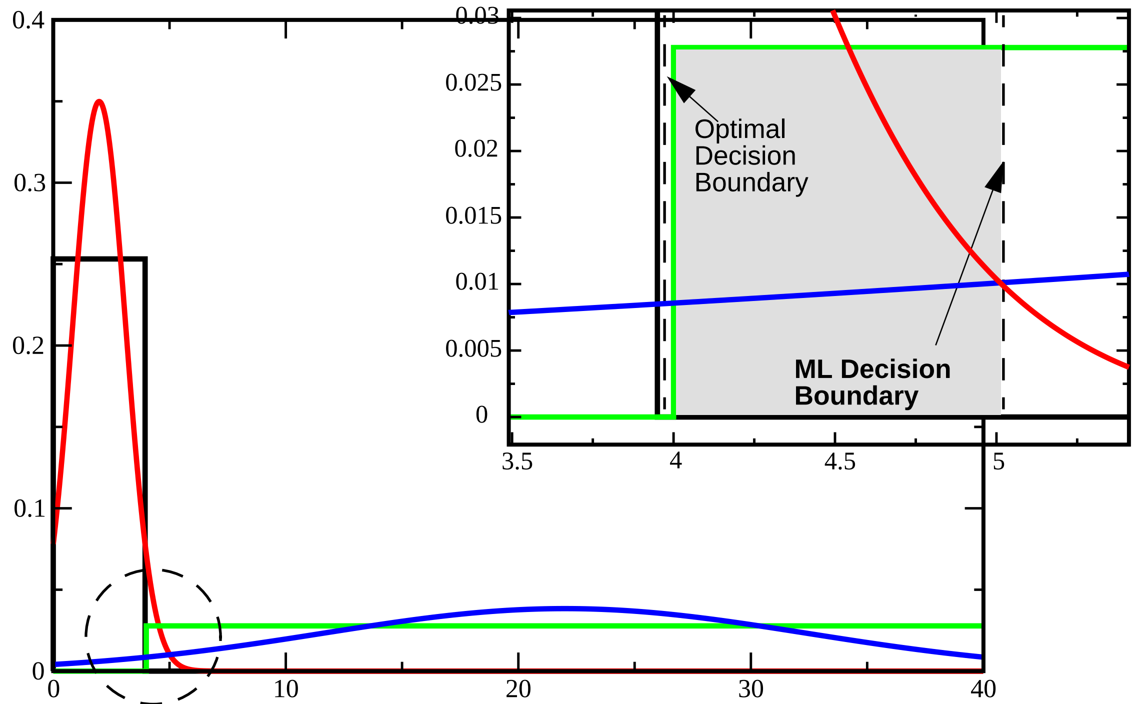
* State-of-the-art performance on the NIST handwritten digit recognition task (Vapnik et al.) — 0.8% error

* State-of-the-art performance on Reuters text categorization (Joachims et al.) — 13.6% error

* Faster training/estimation procedures allow for use of SVMs on complex tasks (Osuna et al.)

* Significant SVM research advances beyond classification — transduction, regression and function estimation

Representation Vs. Discrimination



* Efficient estimation procedures for classifiers based on ML — expectation-maximization makes ML feasible for complex tasks

* Convergence in ML does not necessarily translate to optimal classification

Risk Minimization

* Risk minimization often used in machine learning

$$R(\alpha) = \int Q(z, \alpha)\, dP(z), \qquad \alpha \in \Lambda$$

α: defines the parametrization

Q: is the loss function

z: belongs to the union of the input and output spaces

P: describes the distribution of z.

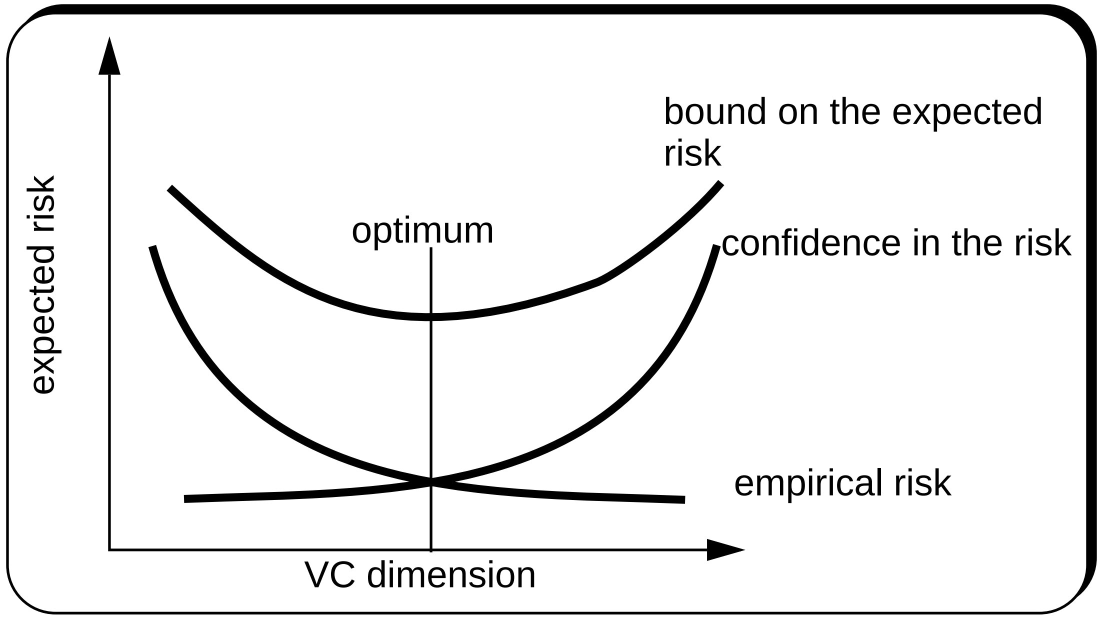
* Loss functions can take several forms (squared error)

* Avoid estimation of P by using empirical risk

$$R_{emp}(\alpha) = \frac{1}{l}\sum Q(z_i, \alpha), \qquad \alpha \in \Lambda$$

* Minimum empirical risk can be obtained by several configurations of the system

Structural Risk Minimization

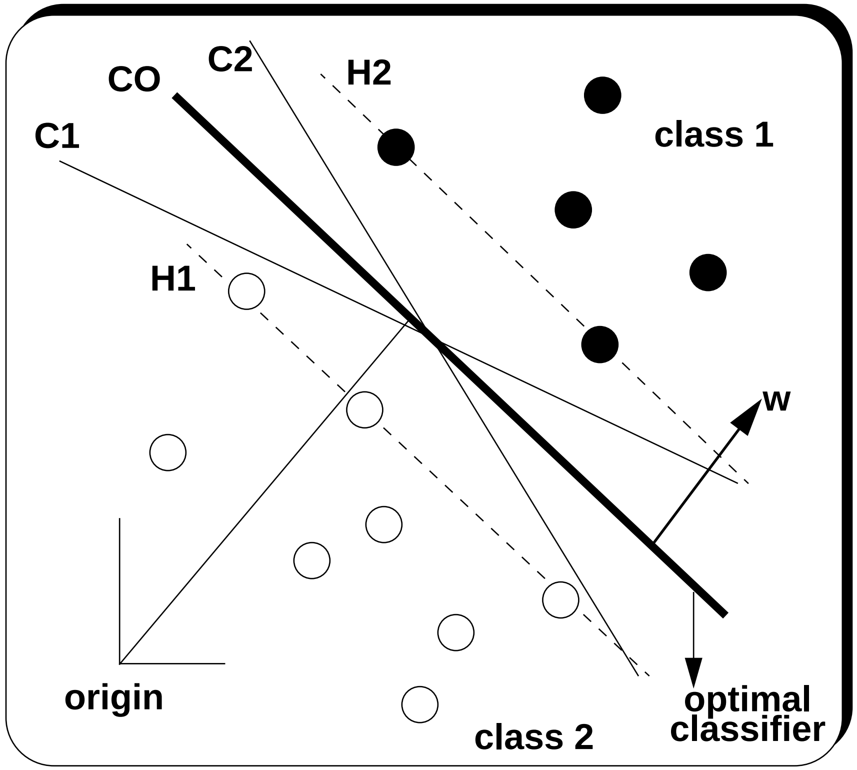


* Control over generalization

$$R(\alpha) \leq R_{emp}(\alpha) + f(h)$$

* h, the VC Dimension is a measure of the capacity of the learning machine

Optimal Hyperplane Classifiers



* Hyperplanes C0, C1 and C2 achieve perfect classification — zero empirical risk

* However, C0 is optimal in terms of generalization

Optimization

* Hyperplane: $\boldsymbol{x} \cdot \boldsymbol{w} + b$

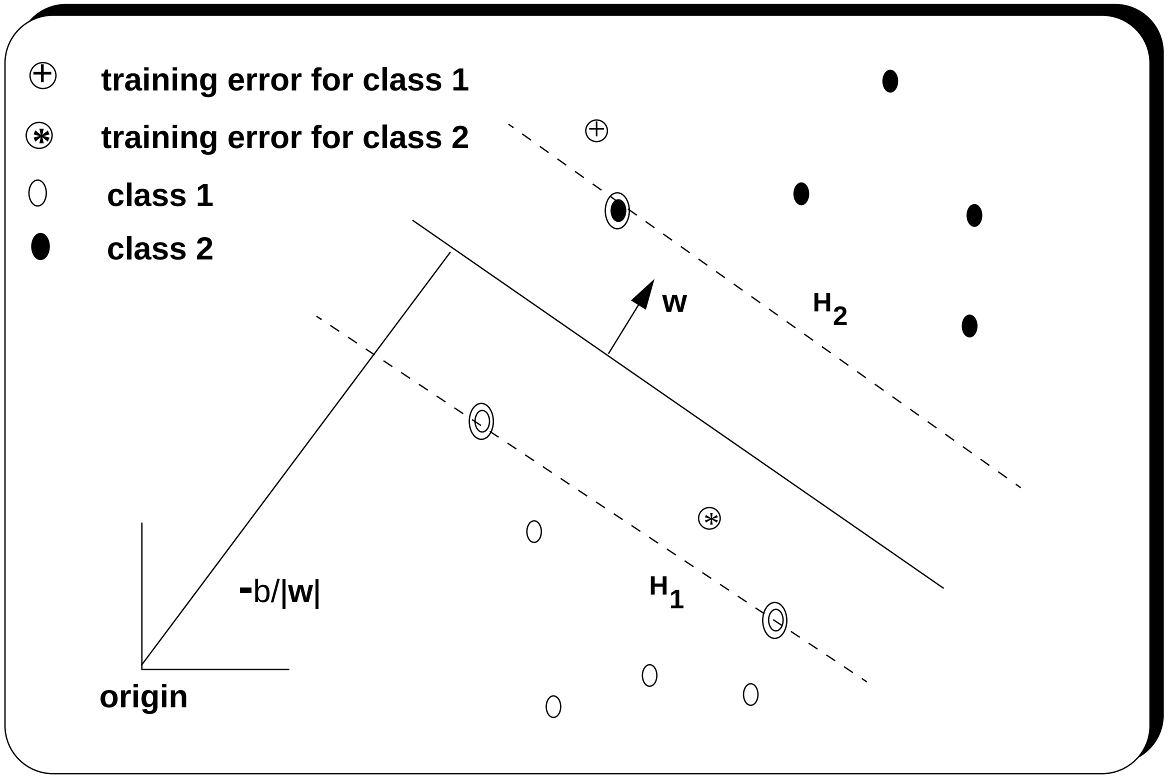
* Constraints: $y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \ge 0 \qquad \forall i$

* Optimize: $L_P = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{N} \alpha_i$

* Lagrange functional setup to maximize margin while satisfying minimum risk criterion

* Final classifier: $f(\boldsymbol{x}) = \sum_{i=1}^{numSVs} \alpha_i y_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b$

Soft Margin Classifiers



* Constraints modified to allow for training errors

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) \geq 1 - \xi_i \qquad \forall i$$

* Error control parameter, C used to penalize training errors

Non-linear Hyperplane Classifiers

* Data for practical applications typically not separable using a hyperplane in the original input feature space

* Transform data to higher dimension where hyperplane classifier is sufficient to model decision surface

$$\Phi : \Re^n \to \Re^N$$

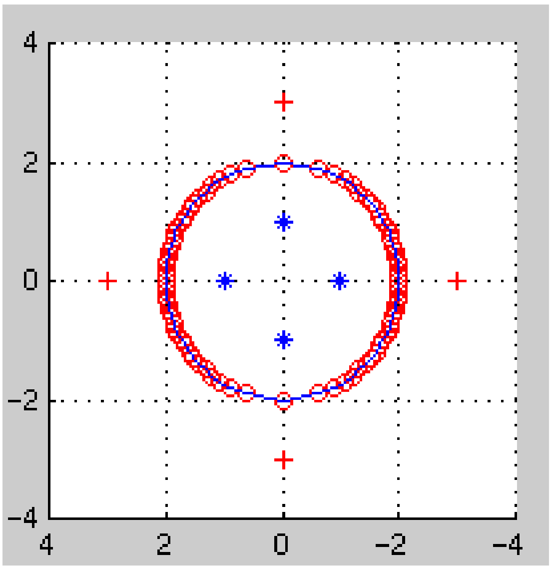
* Kernels used for this transformation

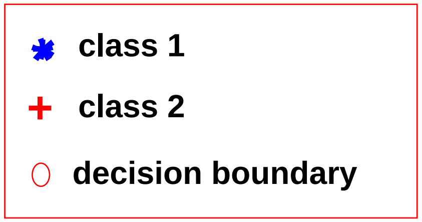
$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

* Final classifier: $f(\boldsymbol{x}) = \sum_{i=1}^{numSVs} \alpha_i y_i K(\boldsymbol{x}, x_i) + b$

Example Non-Linear Classifier

2-dimensional input space





class 1 data points:

(-1,0) (0,1) (0,-1) (1,0)

class 2 data points:

(-3,0) (0,3) (0,-3) (3,0)

class 1 data points:

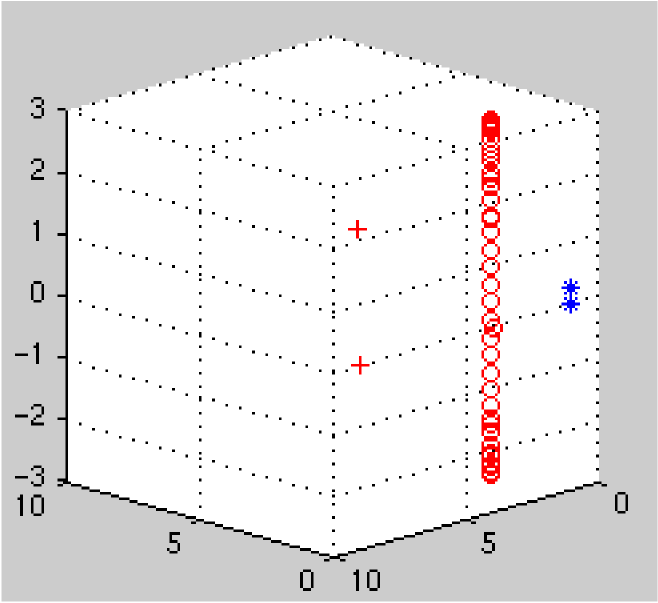
(1,0,0) (0,1,0) (0,1,0) (1,0,0)

class 2 data points:

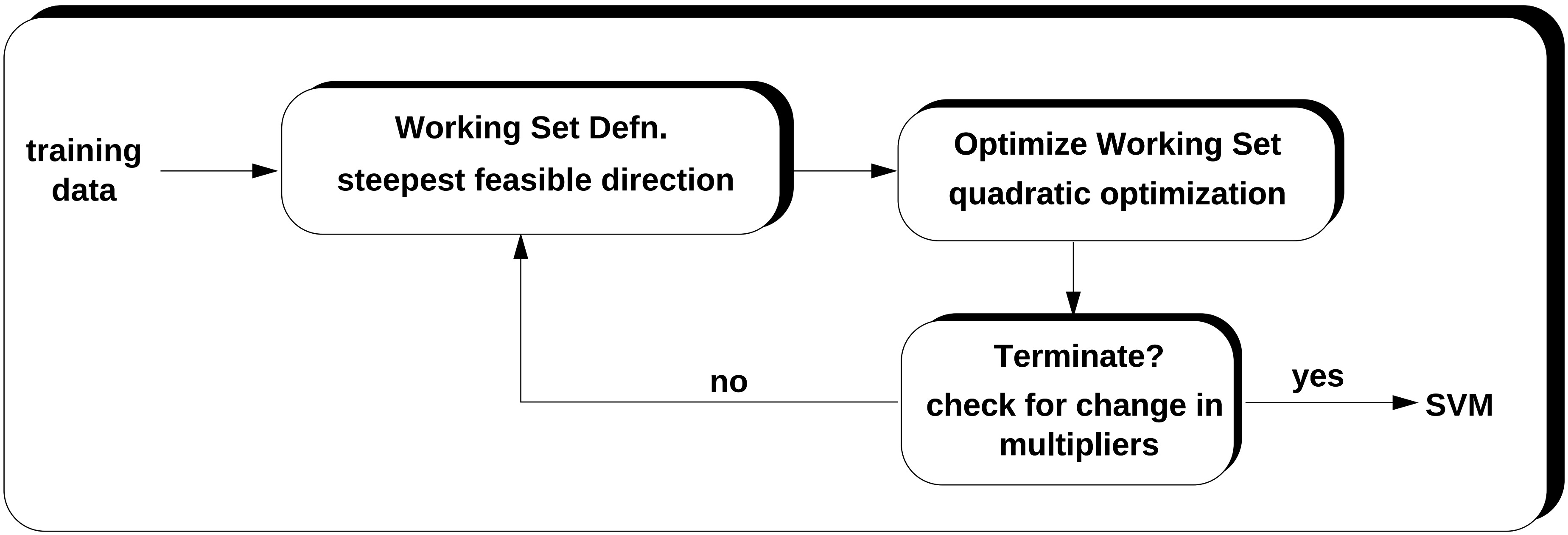
(9,0,0) (0,9,0) (0,9,0) (9,0,0)

$$(x, y) \Rightarrow (x^2, y^2, \sqrt{2}xy)$$

3-dimensional transformed space



Practical SVM Training



- * “Chunking” — proposed by Osuna et al.
- * Guarantees convergence to global optimum
- * Working set definition is crucial

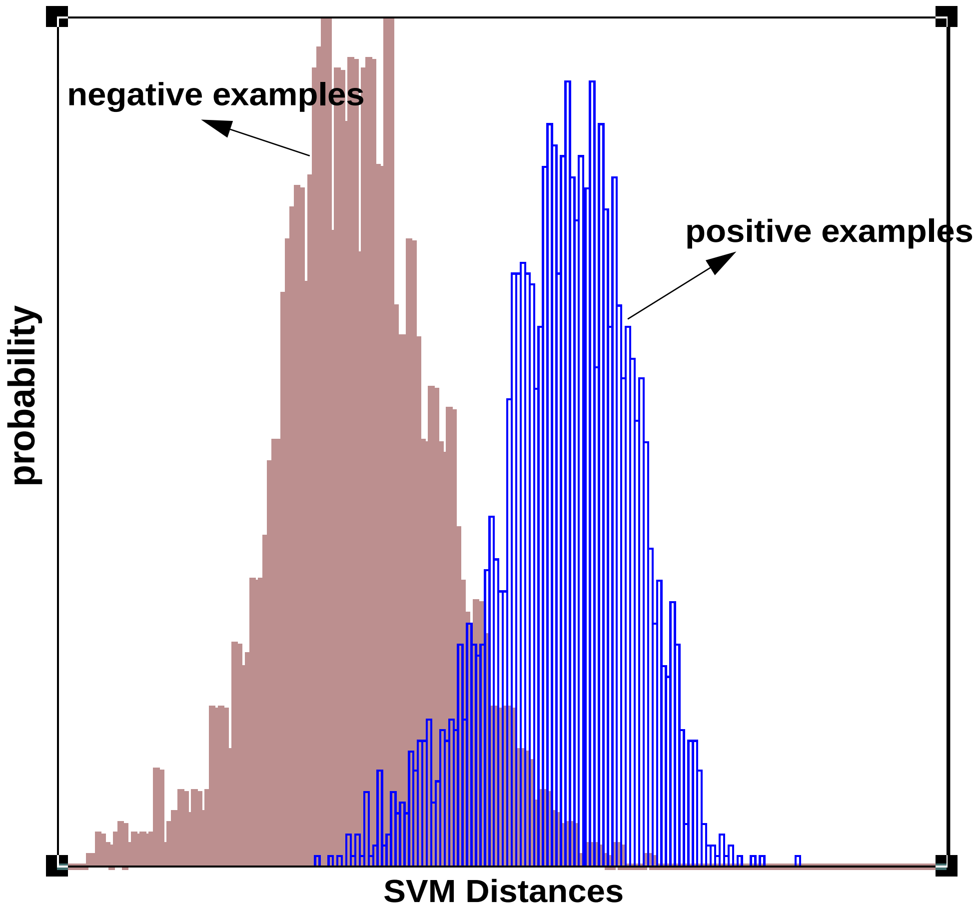
From Classifiers to Recognition

* ISIP ASR system used as the starting point
* Likelihood-based decoding — $\log P(A/M)$ used
* SVMs do not generate likelihoods

$$P(A/M) = \frac{P(M/A)P(A)}{P(M)}$$

* Ignore $P(A)$ and use model priors $P(M)$
* Posterior estimation required
* Feature space needs to be decided — frame level data vs. segment level data
* Use SVM derived posteriors to rescore N-best lists

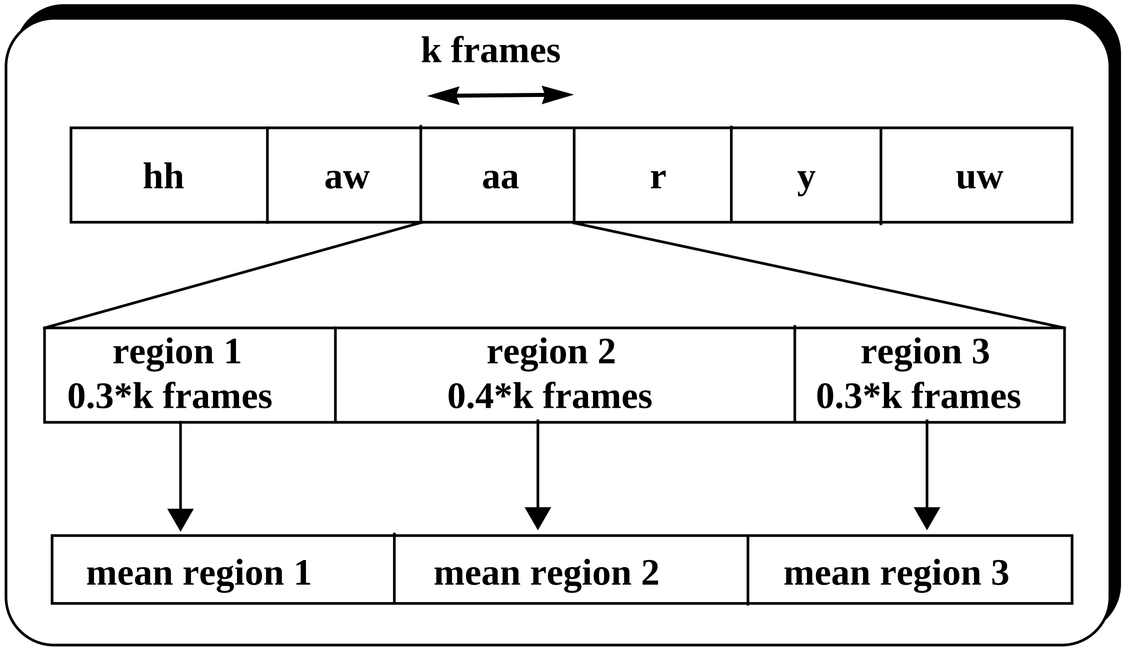
Posterior Estimation



$$p(y = 1 / f) = \frac{1}{1 + \exp(Af + B)}$$

- Gaussian assumption is good for overlap region
- Leads to compact distance-posterior transformation — sigmoid function

Segmental Modeling

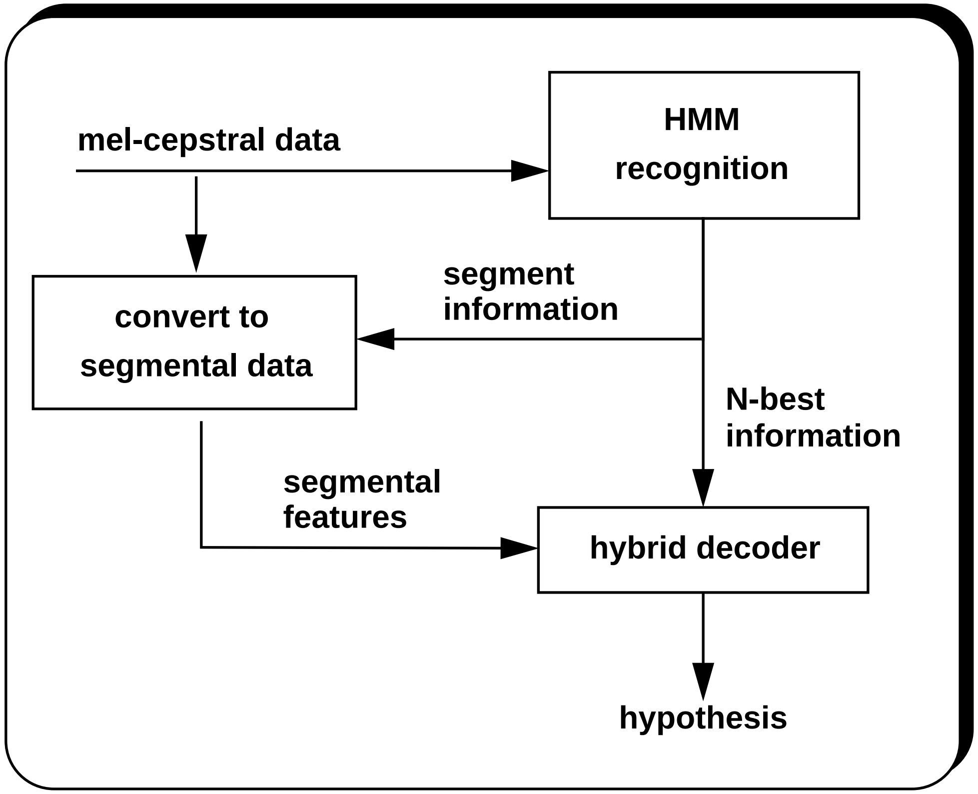


* Allows for each classifier to be exposed to a limited amount of data.

* Captures wider contextual variation

* Approach successfully used in segmental ASR systems where Gaussians are used to model segment duration

Hybrid Recognition Framework



* Gaussian computations replaced with SVM-based probabilities in the hybrid decoder

* Composite feature vectors generated based on traditional HMM-based alignments

Processing Alternatives

* Basic hybrid system operates on a single hypothesis-derived segmentation
  * Approach is simple and saves computations
* Alternate approach involves N segmentations
  * Each segmentation derived from the corresponding hypothesis in the N-best list
  * Computationally expensive
  * Closer in principle to other rescoring-based hybrid frameworks
  * Allows for SVM and HMM score combination

Experimental Data - Deterding Vowel

* Often used for benchmarking non-linear classifiers
* 11 vowels spoken in a “h*d” context
* Training set consists of 528 frames of data from 8 speakers
* Test set composed of 476 frames from seven speakers
* Small size of training set makes the dataset challenging
* Best result reported on this dataset — 29.6% error

Results - Static Data Classification

gamma (C=10)	classification error %	C (gamma=0.5)	classification error %
0.2	45	1	58
0.3	40	2	43
0.4	35	3	43
0.5	36	4	43
0.6	35	5	39
0.7	35	8	37
0.8	36	10	37
0.9	36	20	36
1.0	37	50	36
		100	36

* Best SVM performance: 35% classification error with RBF kernels

* Polynomial kernels perform worse — best performance was a 49% classification error

Experimental Data - OGI Alphadigits

* Telephone database of 6-word strings
* Training Data
  * 52000 sentences
  * 1000 sentences as cross-validation set to estimate sigmoid parameters
* Test data
  * 3329 sentences — speaker independent open-loop test set
* Number of phone classifiers — 30
* 39-dimensional MFCC features used

OGI Alphadigits (AD): Effect of Segment Proportion

Segmentation Proportions	WER (%) RBF kernel	WER (%) polynomial kernel
2-4-2	11.0	11.3
3-4-3	11.0	11.5
4-4-4	11.1	11.4

* Previous research suggests 3-4-3 proportion (Glass, et al.)
* For SVM classifiers, segment proportion does not have any significant impact on classifier accuracy or system performance, especially with RBF kernels
* 3-4-3 proportion used for all further experiments

AD — Effect of Kernel Parameters

RBF gamma	WER (%) hypothesis Segmentation	WER (%) Reference Segmentation	polynomial order	WER (%) hypothesis Segmentation	WER (%) Reference Segmentation
0.1	13.2	9.2	3	11.6	7.7
0.4	11.1	7.2	**4**	**11.4**	**7.6**
0.5	11.1	7.1	5	11.5	7.5
0.6	11.1	7.0	6	11.5	7.5
0.7	**11.0**	**7.0**	7	11.9	7.8
1.0	11.0	7.0			
5.0	12.7	8.1			

* RBF kernels perform better under both the fair and oracle experiments
* Best performance: 11.0% WER vs. 11.9% baseline
* Using single segmentation does not reduce N-best list size significantly

AD — Error Modalities

Data Class	HMM (%WER)	SVM (%WER)
a-set	13.5	11.5
e-set	23.1	22.4
digits	5.1	6.4
alphabets	15.1	14.3
nasals	12.1	12.9
plosives	22.6	21.0
Overall	**11.9**	**11.8**

- Common word class groups used for error analysis
- N-segmentations used for rescoring
- SVM and HMM classifiers seem to have complementary strengths
- Combining the system outputs seems reasonable

AD - Likelihood Combination

Normalization Factor	HMM+SVM (%WER)
100000	11.8
10000	11.4
1000	10.9
500	10.8
200	**10.6**
100	10.7
50	10.8
0.0001	11.9

$$\textit{likelihood} = \textbf{SVM score} + \frac{\textbf{HMM Score}}{\textbf{norm factor}}$$

* Score combination improves overall performance

* Improvement consistent over all error modalities

Experimental Data — SWB

* Telephone database of conversational speech
* Challenging task for ASR systems — casual speaking style with large perplexity
* 114,000 utterance training set
* 2,427 utterance speaker-independent test set
* 42 phones used to model pronunciations
* 39-dimensional MFCC features used
* Variance-normalized data used

SWB - Baseline and Experiments

* Baseline HMM system uses cross-word context-dependent triphone models
* 12 mixture Gaussians per state
* Baseline performance of 41.6% WER
* 90,000 utterances used for estimation of SVM classifiers
* 24,000 utterances used as cross-validation set
* Segment proportion of 3-4-3 used
* Rescoring with hypothesis-based segmentation results in 40.6% WER using RBF kernels

Oracle Experiments

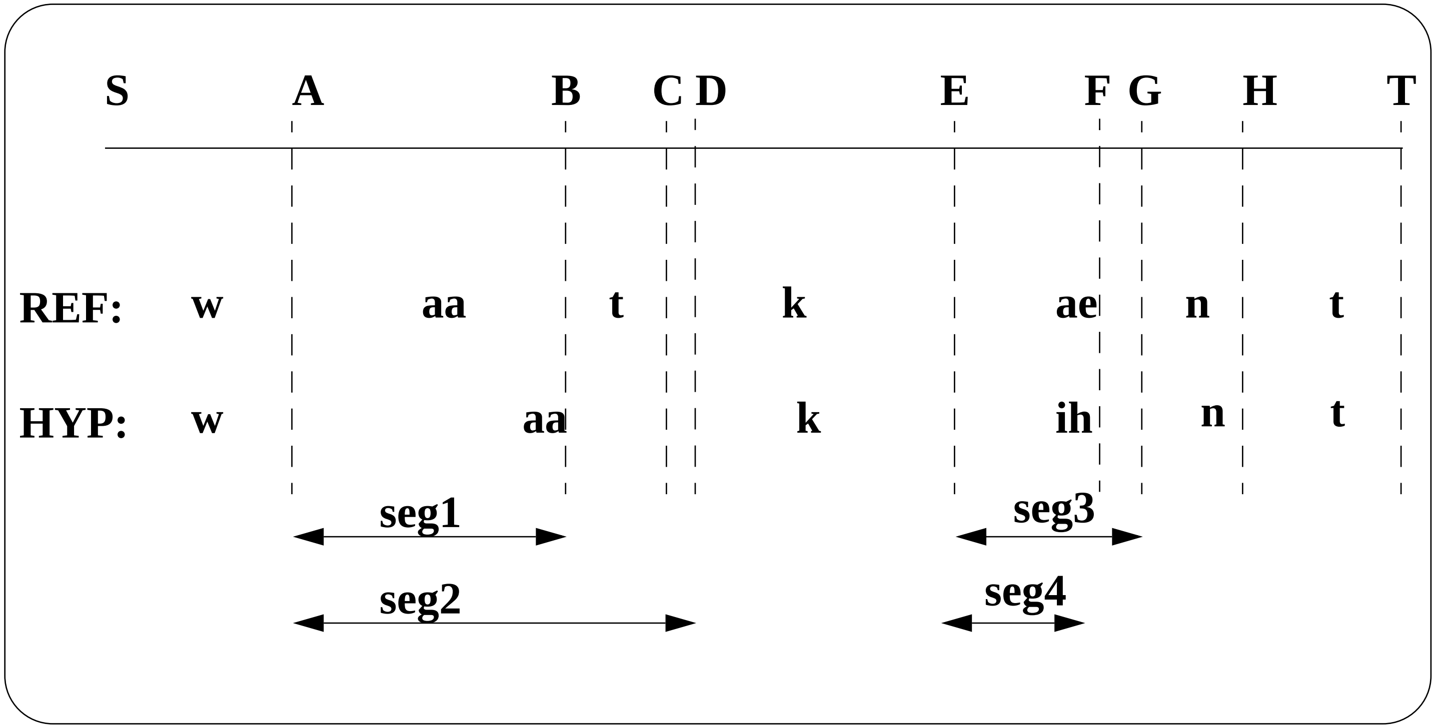
S. No.	Information Source		HMM		Hybrid	
	Transcription	**Segmentation**	**AD**	**SWB**	**AD**	**SWB**
1	N-best	Hypothesis	11.9	41.6	11.0	40.6
2	N-best	N-best	12.0	42.3	11.8	42.1
3	N-best + Ref.	Reference	—	—	3.3	5.8
4	N-best + Ref.	N-best + Ref.	11.9	38.6	9.1	38.1

* Improvement possible from good segmentations and rich N-best lists studied by including reference segmentation and transcription

* Expt. 4 indicates that SVMs do a better job than HMMs when exposed to good segmentations

* Drop in improvements by hybrid system, in comparing expts. 1 and 2, needs further investigation

Segmentation Issue



* Type-A errors: seg1 vs. seg2
Type-B errors: seg3 vs. seg4

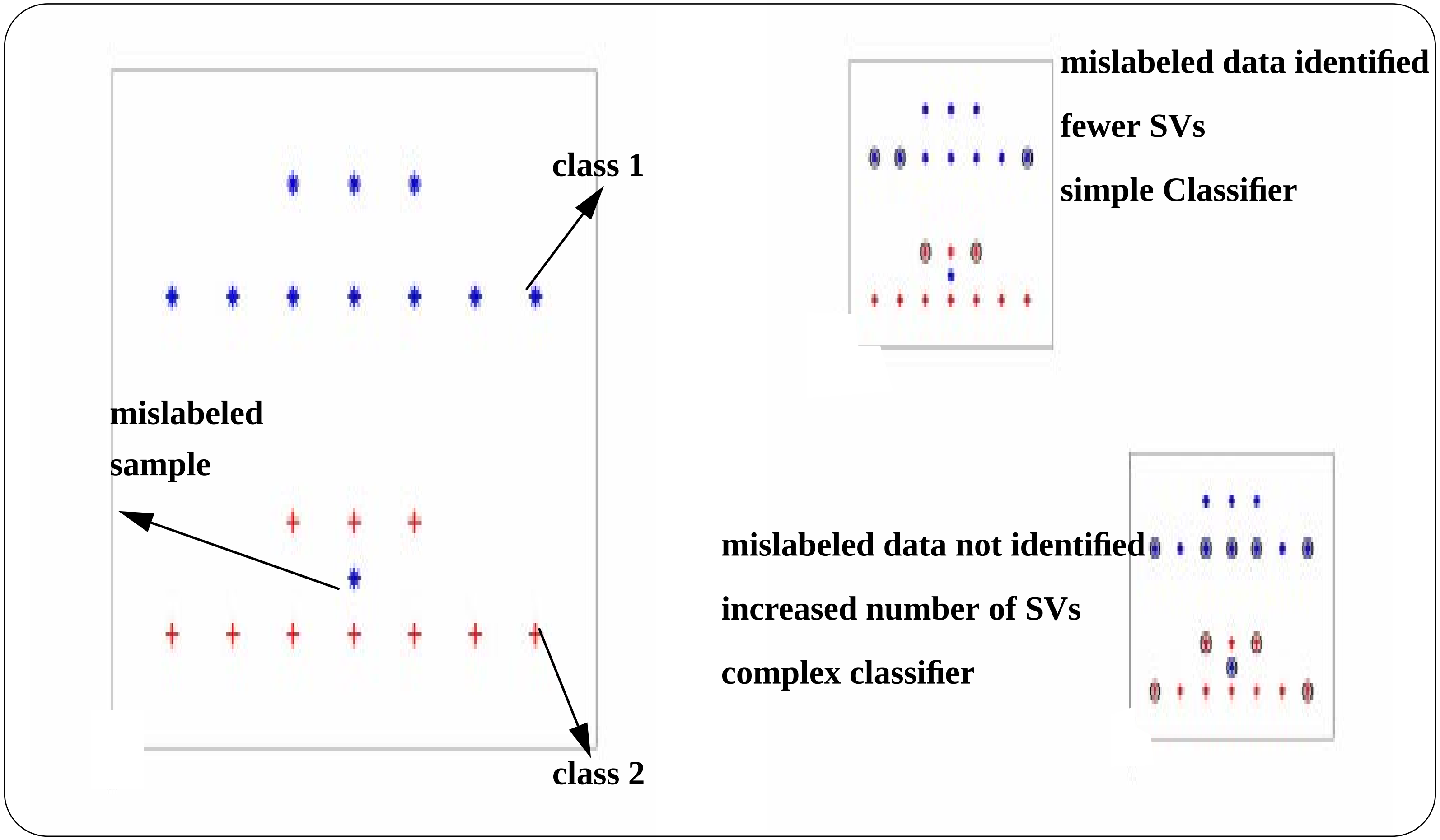
* N-best lists — Type-B errors common

* SWB N-best lists — Type-A errors also significant

Identification of Mislabeled Data

* Chunking converges faster when the working set is composed of examples that violate the Karush-Kuhn-Tucker optimality conditions
* Several support vectors with multipliers at the upper bound (C) — they form the BSVs
* If example identified as a BSV for several iterations, the example is probably mislabeled
* Faster convergence and better classifiers by eliminating mislabeled data
* A “large enough” value for C must be chosen

Synthetic Data Example



* Identifying mislabeled data results in compact classifiers

Summary of Experiments

* Static classification task — Deterding vowel data
  * achieved 35% classification error
* Continuous speech recognition — AD and SWB
  * AD — 11.0% WER vs. 11.9% baseline
  * SWB — 40.6% WER vs. 41.6% baseline
* Score combination improves performance further
* Oracle experiments — reference segmentation and augmented N-best lists
* Segmentation is a primary issue in limited success of the hybrid system

Dissertation Contributions

* First successful attempt to integrate SVMs into a complex recognition system

* Developed a simple hybrid HMM/SVM framework

* Significant performance improvements on small vocabulary task and marginal improvements on large vocabulary task

  * 11.9% to 11.0% on Alphadigits

  * 41.6% to 40.6% on SWB

* Exploration of segment level information

* Concept of identifying mislabeled data

Future Work

* Role of posterior estimation in the hybrid framework
* Use ability of SVMs to identify mislabeled data for data clean up and confidence measures
* Iterative SVM parameter update as part of HMM estimation
* Access to alternate segmentations during SVM estimation
* Fisher kernels and alternate hybrid approaches
* Bayesian approaches for parameter estimation to avoid need for a cross-validation set

Acknowledgements

I would like to thank Dr. Joe Picone for all the mentoring and guidance he has provided during the course of my Ph.D. I would also like to thank Jon Hamaker for the comments he provided during the experimentation and the writing of this dissertation.

Related Publications

1. A. Ganapathiraju, J. Hamaker and J. Picone, "Continuous Speech Recognition Using Support Vector Machines" submitted to *Computers, Speech, and Language*, October 2001.

2. A. Ganapathiraju, J. Hamaker and J. Picone, "A Hybrid ASR System Using Support Vector Machines," *Proceedings of the International Conference of Spoken Language Processing*, vol. 4, pp. 504-507, Beijing, China, October 2000.

3. A. Ganapathiraju and J. Picone, "Support Vector Machines for Automatic Data Cleanup," *Proceedings of the International Conference of Spoken Language Processing*, vol. 4, pp. 210-213, Beijing, China, October 2000.

4. A. Ganapathiraju, J. Hamaker and J. Picone, "Hybrid HMM/SVM Architectures for Speech Recognition," *Speech Transcription Workshop*, College Park, Maryland, USA, May 2000.

5. A. Ganapathiraju, J. Hamaker and J. Picone, "Support Vector Machines for Speech Recognition," *Proceedings of the International Conference on Spoken Language Processing*, pp. 2923-2926, Sydney, Australia, November 1998.