Minimum Classification Error Networks

Based on book chapter 9, by Shigeru Katagiri

Jaakko Peltonen, 28th February, 2002

1

Introduction

- Speech recognition: map dynamic speech instantiation to a **class**
- Adjust recognizer **parameters** for best future recognition accuracy
- **Bayes approach**: decision rule is
  $$C(u_1^{T_0}) = C_i \text{ iff } i = arg\,max_j\, p(C_j | u_1^{T_0})$$
- Estimate class probability as precisely as possible
- Direct estimation with ANNs not successful
- Typical **modules**: feature extractor, classifier (language model, acoustic model)

2

Introduction

- Feature extraction is crucial
  - feature sets: LPC, cepstrum, filter–bank spectrum
- ML problems: form of class distribution unknown, likelihood not directly linked to classification error
- Discriminative training: **discriminant function** in place of conditional class probability, trained to minimize loss
- More closely linked to classification error, but optimality often not proven well, limited versatility
- No interaction between feature extractor and classifier, no guarantee of optimal feature classification

3 Discriminative Pattern Classification

- **Bayes decision theory** (static samples):
  feature pattern $\boldsymbol{x} \in C_k$
- Individual loss: $l_k(C(\boldsymbol{x}))$ (classification $C(\boldsymbol{x})$),
- Expected loss: $L(C(\boldsymbol{x})|\boldsymbol{x}) = \sum_k l_k(C(\boldsymbol{x}))\, p(C_k|\boldsymbol{x})$
- Overall risk: $L = \int L(C(\boldsymbol{x})|\boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}$
  (defines accuracy) $= \int_{\boldsymbol{x},k} l_k(C(\boldsymbol{x}))\, p(C_k, \boldsymbol{x})$
- Error count loss:

$$l_k(C(\boldsymbol{x})) = \begin{cases} 0 & (C(\boldsymbol{x}) = k) \\ 1 & otherwise \end{cases} \qquad L(C(\boldsymbol{x})|\boldsymbol{x}) = 1 - p(C(\boldsymbol{x})|\boldsymbol{x})$$

- Minimize resp. overall risk $\rightarrow$ minimum error rate
  $\rightarrow$ simulate expected loss: ML approach (difficult)

Discriminative Pattern Classification

- **Discriminative training:**
  evaluate error count loss accurately

- Classification criterion:

$$C(\boldsymbol{x})=C_i \quad \text{iff} \quad i=arg\,max_j\, g_j(\boldsymbol{x};\Lambda)$$

- Training characterized by
  - discriminant function: pattern type specific
  - design objective (loss): e.g. n. of class errors
  - optimization method: heuristic/proven
  - consistency/generalization: discriminant choice

- Ideal overall loss: $L(\Lambda)=\sum_k P(C_k)\int l_k(\boldsymbol{x};\Lambda)\,p(\boldsymbol{x}|C_k)\,d\,\boldsymbol{x}$

- Empirical average loss: $L_0(\Lambda)=\frac{1}{N}\sum_k \sum_n l_k(\boldsymbol{x}_{\boldsymbol{n};}\Lambda)\,1(\boldsymbol{x}_{\boldsymbol{n}}\in C_k)$

5

Discriminative Pattern Classification

- Other loss forms:
  - perceptron loss, squared error loss, mutual information
  - suboptimal, results may be inconsistent with minimum classification error (MCE)
- Optimization: batch/sequential, error correction, stochastic approximation, simulated annealing, gradient search
- Purpose: accurate classification for task at hand, not just training data – additional information needed
  - ML introduces a parametric probability function
  - discriminative training is more moderate: consistency from choice of discriminant function

6

Generalized Probabilistic Descent

- Problems with existing recognizers:
  - –lack of optimality results (LVQ, corrective training)
  - –minimal squared error or maximal mutual information $\neq$ minimal misclassifications
  - –concentrate on acoustic modeling, not overall process

- Generalized probabilistic descent (GPD): approximate classification error count loss using sigmoidal functions and L_p norm

GPD Basics

- Input sample: sequence of T F–dimensional vectors

$$\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}}=f(u_1^{T_0})=(\boldsymbol{z}_{\boldsymbol{1}},\ldots,\boldsymbol{z}_{\boldsymbol{T}})$$

- Classifier: M classes, B_j trainable prototypes each

$$\Lambda=\{\lambda_j\}=\{\{\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{b}}\}_{b=1}^{b=B_j}\}_{j=1}^{j=M}$$

- Decision rule based on **distances** g_j :

$$C(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}})=C_i \text{ iff } i=arg\,min_j\, g_j(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T};}\Lambda)$$

- Distances are **minimal paths** to closest prototypes:

$$g_j(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}};\Lambda)=min_{b,\vartheta}\{D_\vartheta(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}},\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{b}})\}$$

$$D_\vartheta(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}},\boldsymbol{r}_{\boldsymbol{j}}^{\boldsymbol{b}})=\left\|\boldsymbol{z}_{\omega(\boldsymbol{j},\boldsymbol{b},\boldsymbol{tau},\vartheta)}-\boldsymbol{r}_{\boldsymbol{j},\tau}^{\boldsymbol{b}}\right\|^2$$

GPD Basics

- Design target: find optimal Λ , adjust based on individual loss
- Adjustment is **gradient** –based, but classification error is not differentiable w.r.t. parameters
- Solution: replace discriminant by

$$g_j(\boldsymbol{x}_1^T;\Lambda)=\left[\sum_{b=1}^{B_j}\{D(\boldsymbol{x}_1^T,\boldsymbol{r}_j^b)\}^{-\zeta}\right]^{-1/\zeta}$$

$L_{-\zeta}$ –norm over classes

$$D(\boldsymbol{x}_1^T,\boldsymbol{r}_j^b)=\left[\sum_{\vartheta=1}^{\Theta_j^b}\{D_\vartheta(\boldsymbol{x}_1^T,\boldsymbol{r}_j^b)\}^{-\xi}\right]^{-1/\xi}$$

$L_{-\xi}$ –norm over paths

- Misclassification measure:

$$d_k(\boldsymbol{x}_1^T;\Lambda)=g_k(\boldsymbol{x}_1^T;\Lambda)-\left[\frac{1}{M-1}\sum_{j,\,j\neq k}\{g_j(x_1^T;\Lambda)\}^{-\mu}\right]^{-1/\mu}$$

Probabilistic Descent Theorem

- If classifier parameters are adjusted by

$$\delta \Lambda(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}}(t), C_k, \Lambda) = -\epsilon U \nabla l_k(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}}(t); \Lambda)$$

- Then the overall loss decreases on average:

$$E[\delta L(\Lambda)] \leqslant 0$$

- Parameters converge to a local loss minimum if

$$\sum_{t=1}^{\infty} \epsilon(t) \to \infty \quad , \quad \sum_{t=1}^{\infty} \epsilon(t)^2 < \infty$$

- Functions are smooth, so gradient adjustment can be used
- Adjustment is done for all paths, all prototypes
- loss function: sigmoidal function of $d_k(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{T}}; \Lambda)$
  – if probability form known, smooth loss yields MAP

Experiments

- E–set task: classify E–rhyme letters (9 classes)
- modified k–means, 3 prototypes: 64.1% to 64.9%
  MCE/GDP, 3 prototypes: 74.0% to 77.2%
  (up to 84.4% for 4 prototypes)
- P–set task: classify Japanese phonemes (41 classes)
- segmental k–means, 5 prototypes: 86.8%
  MCE/GDP, 5 prototypes: 96.2%

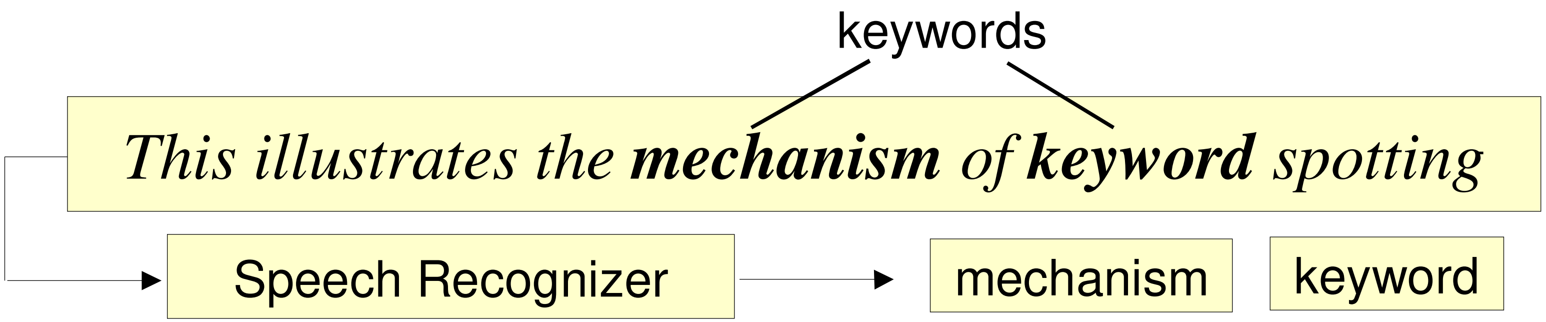
11 Derivatives

- More realistic applications needed: connected word recognition, open–vocabulary speech recognition...
- GPD has been extended to a family of more suitable methods
- **Segmental GPD**: classify continuous speech
  - sub–word models: divide input into segments
  - HMM–based acoustic models
  - discriminant function: class membership of a connected word sample: likelihood measure

$$g_{W_r}(\boldsymbol{x}_1^T;\Lambda) = max_{W \neq W_1,\ldots,W_{r-1}} P(\boldsymbol{x}_1^T, \boldsymbol{q}_{W_r}, W_r | \Lambda)$$

  - misclassification: norm over competing sequences
  - softmax reparameterization

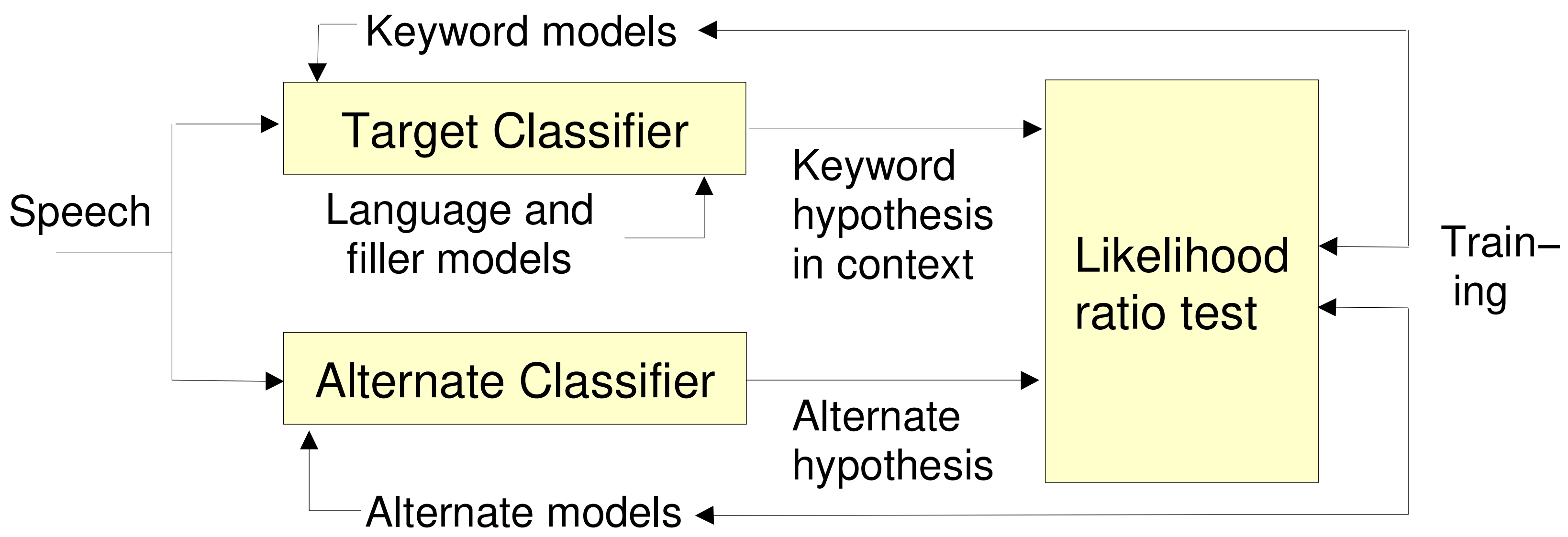
Open–Vocabulary Speech Recognition



- Recognize only selected keywords
- Approaches:
  1) keyword spotting (treshold comparison),
  2) ’filler’ (nonkeyword) model, continuous recognition
- 1) Design model & treshold to minimize spotting error
  - if discriminant is low, keyword exists
  - error types: false detection, false alarm
  - loss function adjustable to emphasize types

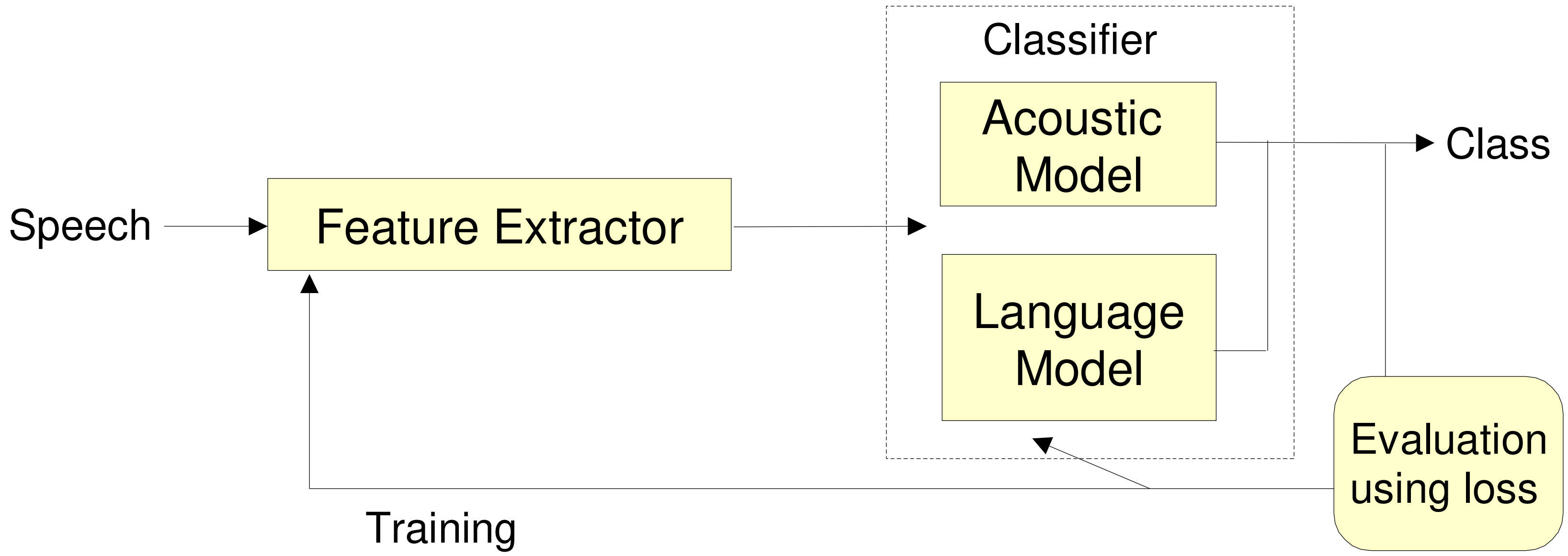
13

Open–Vocabulary Speech Recognition



- 2) Two classifiers: target and ’imposter’
  - target must be more likely than ’imposter’ (ratio above treshold) to accept keyword
  - GPD used to minize false detections and alarms
  - distance: log of likelihood ratio
  - loss selected by error type
  - **speaker recognition** is similar

Discriminative Feature Extraction



- Replace classification rule with recognition rule including feature extraction process T:

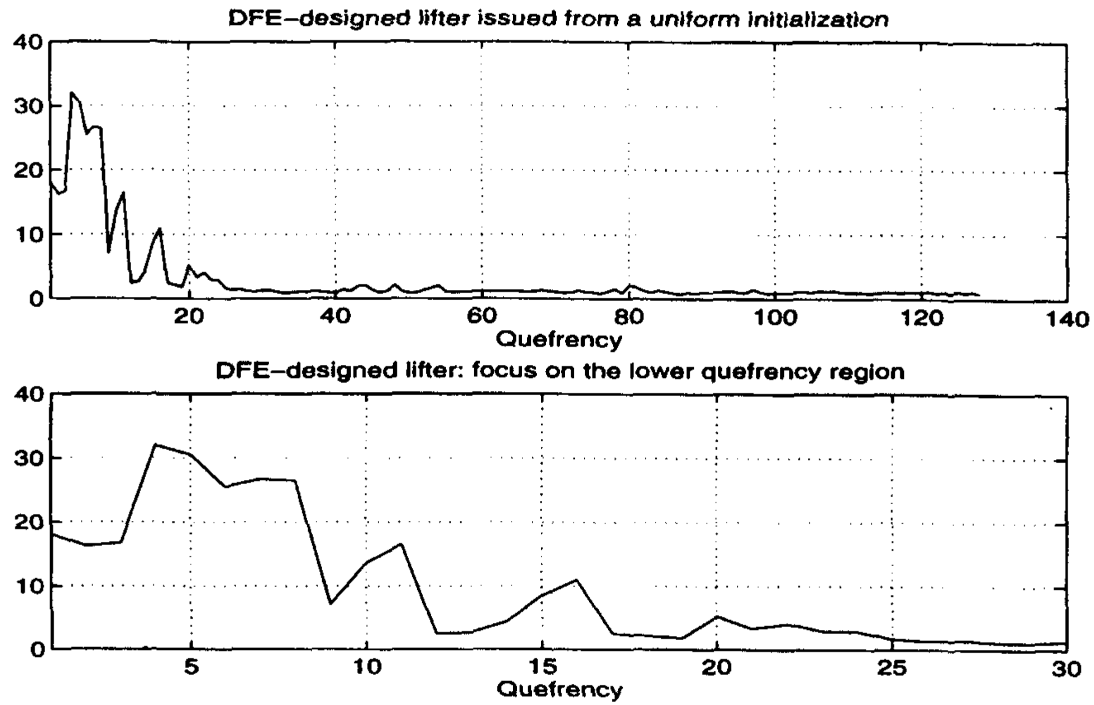
$$C(u_1^{T_0}) = C_i \quad \text{iff} \quad i = arg\,max_j\, g_j(T_\phi(u_1^{T_0}); \Lambda)$$

- Overall recognizer optimized (extraction parameters by chain rule)

- Also applicable to intermediate features

Discriminative Feature Extraction

- Example: Cepstrum–based speech recognition
- Lifter shape: low–quefrency conponents important, shape found by trial and error
- DFE: design lifter to minimize recognition errors – error reduction from 14.5% to 11.3%



Discriminative Feature Extraction

- Discriminative **metric** design:
  - each class has its own metric (feature extractor)

$$C(u_1^{T_0}) = C_i \quad \text{iff} \quad i = arg\,max_j\, g_j(T_\phi^j(u_1^{T_0}); \Lambda)$$

- Minimum error learning subspace method:
  - PCA subspace design does not guarantee low error
  - iterative training better but not rigorous →DFE used

Exercise

- Give example problems (training situations) where these alternate design objectives give suboptimal solutions in the minimum classification error sense:
  - maximum likelihood
  - minimum perceptron loss
  - minimum squared error
  - maximal mutual information