Adaptation

Jeff A. Bilmes

bilmes@ee.washington.edu

http://ssli.ee.washington.edu/~bilmes

Department of Electrical Engineering

University of Washington, Seattle

Adaptation, joint work with:

- Xiao Li
  - Former student at University of Washington, now at Microsoft Research

Adaptation

- Many forms of adaptation: MLLR and its variants, MAP adaptation, hybrids, eigenspace, language model adaptation, etc.
- Adaptation has been and will continue to be crucial to obtaining best ASR performance.
- Adaptation is an idea useful not only to speech recognition, but little attention given to why it works so well.

Traditional Pattern Classification

- Vapnik gave us empirical risk minimization.
- Gave us a theory that we can use to predict, for a given distribution, how many training samples m to we need in order to help predict how poorly we will do.
- He gave us that some form of regularization is almost always necessary (unless we have lots of training data).
- … but this theory assumes that the training and (future) test distributions are identical.

Standard Inductive Learning

- Given
  - a set of *m* samples $(x_i, y_i) \sim p(x, y)$
  - a decision function space $F: X \rightarrow \{\quad 1\}$
- Goal
  - learn a decision function $f \in F$ that minimizes the *expected error*

$$R_{p(x,y)}(f) = \mathrm{E}_{(x,y)\sim p(x,y)}[\mathrm{I}(f(x) \neq y)]$$

- In practice
  - minimize the *empirical error*

$$R_{\mathrm{emp}}(f) = \frac{1}{m}\sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{I}(f(x_i) \neq y_i)$$

  - while applying certain *regularization* strategy to achieve good generalization performance

Why Is Regularization Helpful?

- Learning theory says

$$\Pr\{\, R(f) \le R_{\text{emp}}(f) + \Phi(F, f, m, \delta) \,\} \ge 1 - \delta$$

  - Frequentist: *Vapnik's VC bound* expresses Φ as a function of the VC dimension of F
  - Bayesian: the *Occam's Razor bound* expresses Φ as a function of the prior probability of f
- "Accuracy-regularization"
  - We want to minimize the empirical error as well as the capacity or complexity term.
  - Frequentist: support vector machines, MLPs with weight decay
  - Bayesian: Bayesian model selection, Gaussian Prior.

Practical Work on Adaptation

- Gaussian mixture models (GMMs)
  - MAP (*Gauvain 94*); MLLR (*Leggetter 95*)
- Support vector machines (SVMs)
  - Boosting-like approach (*Matic 93*)
  - Weighted combination of old support vectors and adaptation data (*Wu* 04)
- Multi-layer perceptrons (MLPs)
  - Shared “internal representation” (*Baxter 95, Caruana 97, Stadermann 05*)
  - Linear input network (*Neto 95*)
- Conditional maximum entropy models
  - Gaussian prior (*Chelba 04*)

Adaptation: training/test is different

- Two related yet different distributions
  - Training $p^{tr}(x,y)$
  - target (test-time) $p^{ad}(x,y)$
- Given
  - An unadapted model $f^{tr} = \arg\min_{f \in F} R_{p^{tr}(x,y)}(f)$
  - Adaptation data (labeled) $D_m = \{(x_i, y_i) \sim p^{ad}(x,y)\}_{i=1}^m$
- Goal
  - Learn an adapted model that is as close as possible to our desired model $f^{ad} = \arg\min_{f \in F} R_{p^{ad}(x,y)}(f)$
- Notes
  - Assume sufficient training data but limited adaptation data
  - Training data is not preserved

Why Is Regularization Helpful?

- Learning theory says

$$\Pr\{\ R(f) \leq R_{\text{emp}}(f) + \Phi(F, f, m, \delta)\ \} \geq 1 - \delta$$

  - Frequentist: *Vapnik's VC bound* expresses Φ as a function of the VC dimension of F
  - Bayesian: the *Occam's Razor bound* expresses Φ as a function of the prior probability of f
- "Accuracy-regularization"
  - We want to minimize the empirical error as well as the capacity or complexity term.
  - Frequentist: support vector machines, MLPs with weight decay
  - Bayesian: Bayesian model selection, Gaussian Prior.

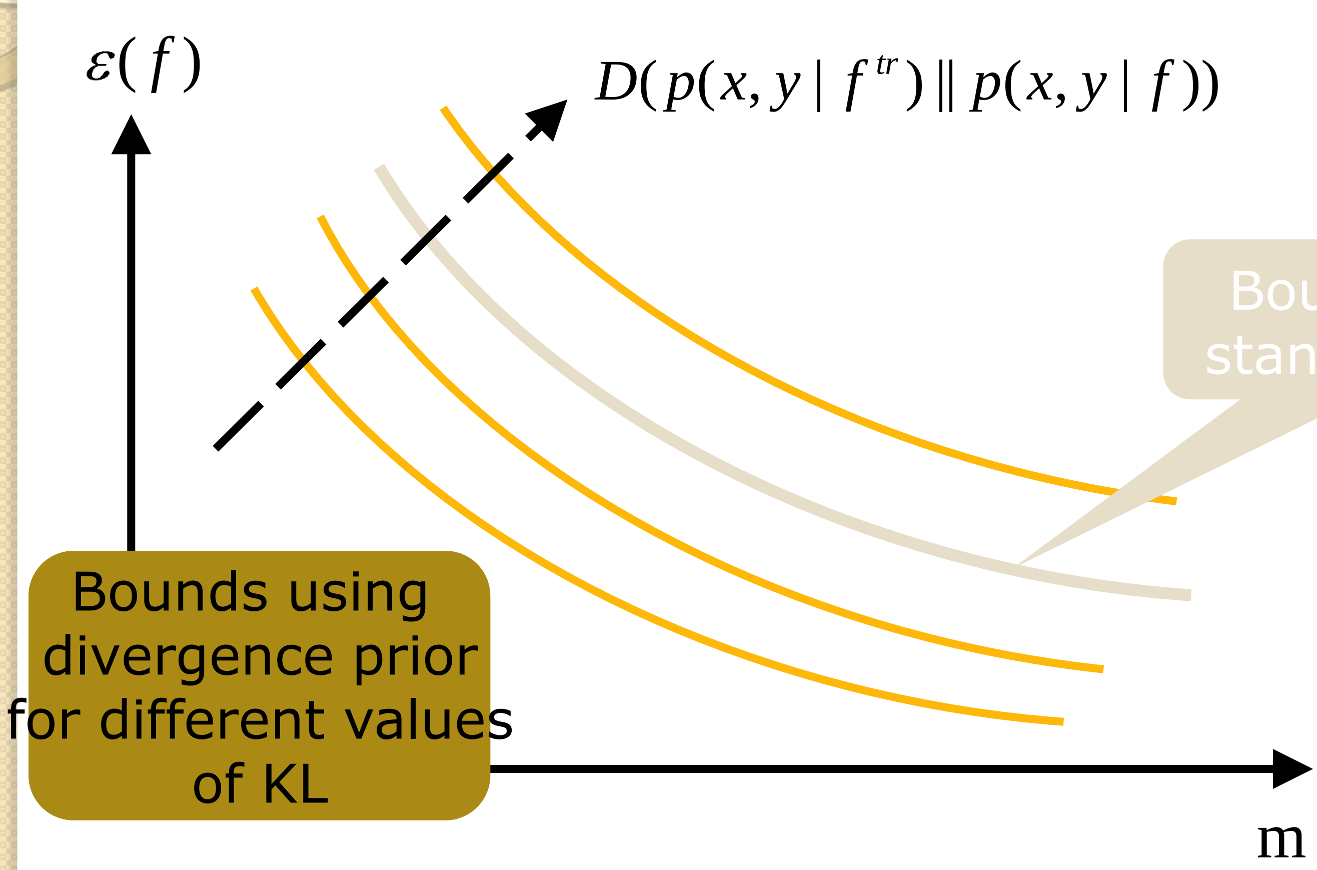
Occam’s Razor Bound for Adaptation

- For a *countable* function space

$$\Pr\left\{ R(f) \le R_{\text{emp}}(f) + \sqrt{\frac{-\ln p_{\text{fid}}(f) - \ln\delta}{2m}} \right\} \ge 1-\delta$$

$$\sqrt{\frac{D(p(x,y\,|\,f^{tr}) \,\|\, p(x,y\,|\,f)) - \beta - \ln\pi(f) - \ln\delta}{2m}}$$

The Critical Value of D()



Bound using standard prior

Bounds using divergence prior for different values of KL

Discriminative Models

- A unified view of SVMs, MLPs, CRFs and etc.
  - Affine classifiers in a transformed space $f = (\ w, b)$
  - Classification $\operatorname{sgn}(w^T \phi(x) + b)$

	$\Phi(x)$	Loss function
SVMs	Determined by the kernel	Hinge loss
MLPs	Hidden neurons	Log loss
CRFs	Any feature function	Log loss

$$p(y \mid x, f) = \frac{1}{1 + e^{-y(w^T \phi(x) + b)}}$$

  - Conditional likelihood (for binary case)

Discriminative Models (cont.)

- Conditional models $p(y \mid x, f)$
  - Classification $\arg\max_y [\log p(x, y \mid f)]$
  - Posterior $p(f \mid x, y) = \dfrac{p(y \mid x, f)\pi(f)}{\sum p(y \mid x, f)\pi(f)}$
  - Assume f^{tr} and f^{ad} are the *true* models generating the training and target *conditional distributions* respectively, *i.e.*

$$p(y \mid x, f^{tr}) = p^{tr}(y \mid x)$$
$$p(y \mid x, f^{ad}) = p^{ad}(y \mid x)$$

Fidelity Prior for Conditional Models

- Again a divergence

$$\ln p_{\text{fid}}(f) = -D(p(y \mid x, f^{tr}) \,\|\, p(y \mid x, f)) + \ln \pi(f) + \beta$$

where $\beta > 0$

  - What if we do not know $p^{tr}(x, y)$
  - We seek an upper bound on the KL-divergence and hence a lower bound on the prior

- Key result

$$D(p(y \mid x, f^{tr}) \,\|\, p(y \mid x, f)) \le R \,\| w - w^{tr} \| + | b - b^{tr} |$$

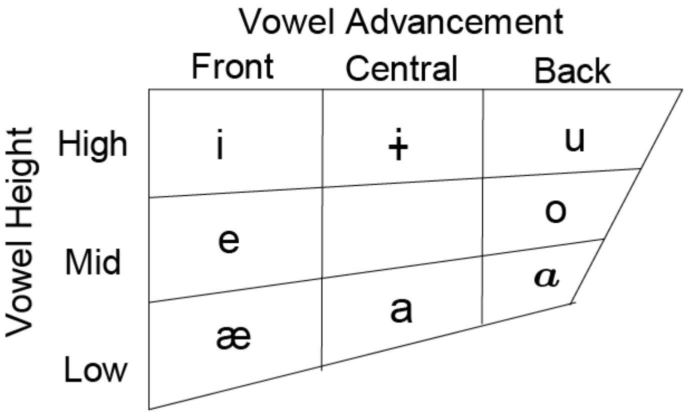
where

$$R = \mathrm{E}[\| x \|]$$

VJ Vowel Dataset

- Task
    - 8 Vowel classes
    - *Frame-level* classification error r
    - Speaker adaptation
- Data allocation
    - Training set – 21 speakers, 420K samples

      For SVM, we random selected 80K samples for training
    - Test set – 10 speakers, 200 samples
    - Dev set – 4 speakers, 80 samples
- Features
    - 182 dimensions – 7 frames of MFCC+delta features



Neural Network Adaptation Procedures

- **Unadapted**
- **Retrained**
  - Start from randomly initialized weight and train *with weight decay*
- **Linear input network** (*Neto 95*)
  - Add a linear transformation in the input space
- **Retrained speaker-independent** (*Neto 95*)
  - Start from the unadapted; train both layers
- **Retrained last layer** (*Baxter 95*, *Caruana 97*, *Stadermann 05)*
  - Start from the unadapted; only train the last layer
- **Retrained first layer** (*proposed here*)
  - Start from the unadapted; only train the first layer
- **Regularized**
  - Note that all above (except retrained) can be considered as special cases of regularized

SVM Adaptation

- RBF kernel (std=10) optimized for training and fixed for adaptation
- Mean and std. dev over 10 speakers; red are significant at p<0.001 level

# adapt. samples per speaker	1K	2K	3K
Unadapted	38.21 4.68	38.21 4.68	38.21 4.68
Retrained	24.70 2.47	18.94 1.52	14.00 1.40
Boosted	29.66 4.60	26.54 2.49	28.85 2.03
Bootstrapped	26.16 5.07	19.24 1.32	14.41 1.26
Regularized	23.28 4.21	19.01 1.30	15.00 1.41
Ext. regularized	28.55 4.99	25.38 2.49	20.36 2.08

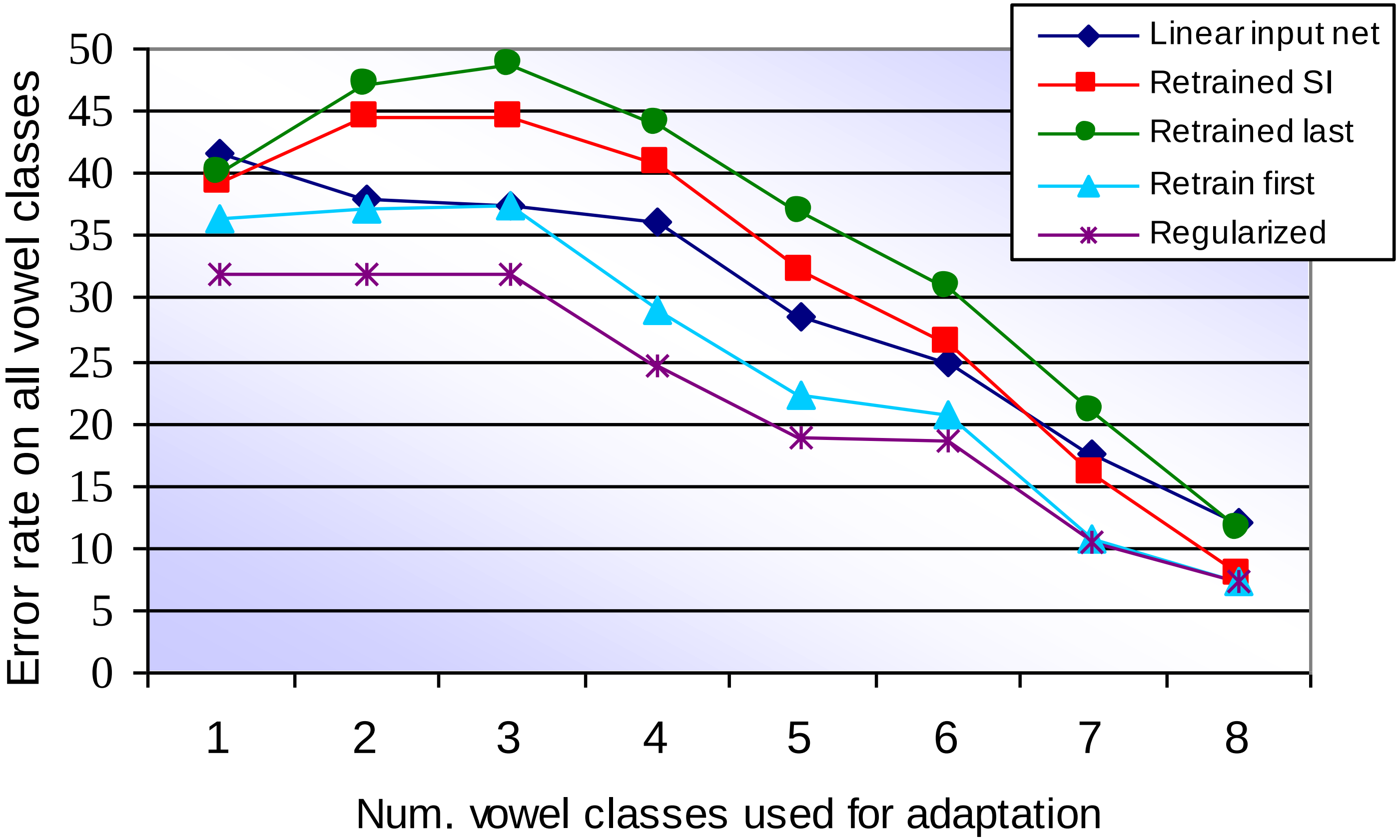
MLP Adaptation (I)

- 50 hidden nodes
- Mean and std. dev over 10 speakers

# adapt. samples per speaker	1K	2K	3K
Unadapted	32.03 3.76	32.03 3.76	32.03 3.76
Retrained (reg.)	14.21 2.50	10.06 3.15	9.09 3.92
Linear input	13.52 2.22	11.81 2.33	11.96 1.77
Retrained SI	12.15 2.70	9.64 2.74	7.88 2.49
Retrained last	15.45 2.75	13.32 2.46	11.40 2.37
Retrained first	11.56 2.09	9.12 3.08	7.35 2.26
Regularized	11.56 2.09	8.16 2.60	7.30 2.40

MLP Adaptation (II)

- Varying number of vowel classes available in adaptation data


Linear input net
Retrained SI
Retrained last
Retrain first
Regularized
Error rate on all vowel classes
Num. vowel classes used for adaptation