論文 / 著書情報
Article / Book Information

Title	Strategies for model training and adaptation based on data dependency control
Author	Takahiro Shinozaki, Sadaoki Furui
Journal/Book name	Proc. APSIPA ASC 2011 Xi ' an, , ,
Issue date	2011, 10

Strategies for Model Training and Adaptation Based on Data Dependency Control

Takahiro Shinozaki* and Sadaoki Furui†
* Chiba University, Chiba, Japan
www.ailab.tj.chiba-u.jp/~shinot
† Tokyo Institute of Technology, Tokyo, Japan

***Abstract*—In speech recognition systems, decoding or data evaluation processes are sometimes performed as a part of a model estimation process. For example, when unsupervised adaptation is performed, parameters are adapted using decoding hypotheses generated by an initial model. In these model estimation frameworks, how to design dependency structures between data and models is an important issue, since it significantly affects recognition performance. This paper overviews model estimation methods that have been proposed from this viewpoint. These include efficient cross-validation (CV) based parameter tuning and structure optimization, domain adaptation based on selecting training subset, and iterative parameter estimation techniques that integrate CV into the process.**

I. Introduction

In speech recognition systems, large and structured statistical models are used. These models need to be accurately trained from a limited amount of training data. Sometimes, there is a mismatch between domains of training and evaluation data and recognition systems are required to be robust for such discrepancies. In order to meet these requirements, several model training and adaptation methods have been proposed based on advanced use of data dependency control. In this paper, these methods are categorized and reviewed.

The basic and simple discipline with statistical model evaluation is that training and evaluation data must be independent. If evaluation data is included in the training data, the evaluation results will be positively biased. The same independence requirement sometimes exists in model training itself. For example, structure optimization needs an unbiased evaluation score. As a general strategy for doing this, a development data is used in addition to the training data. The development data may be a randomly sampled held-out subset of the original training data or may be a data set that are separately recorded from the training data. In either case, the point is that there is no overlap between the training data and the development data, so that the tuning is independent from the model training. An extension of the hold-out strategy is the K-fold Cross-validation (K-fold CV), which can efficiently use limited training data.

The organization of the rest of this paper is as follows. In Section II, the hold-out and cross-validation methods are briefly reviewed and their relationship in terms of expected error is derived. In Section III, CV based model selection and parameter tuning methods are described that efficiently compare a large number of models or optimize continuous variables. In Section IV, a domain adaptation framework and its applications are described that are based on selecting a subset of training data so as to directly maximize an evaluation score on development data. In Section V, parameter estimation techniques are described that integrate CV into an iterative optimization procedure. Finally, summary and conclusion are given in Section VI.

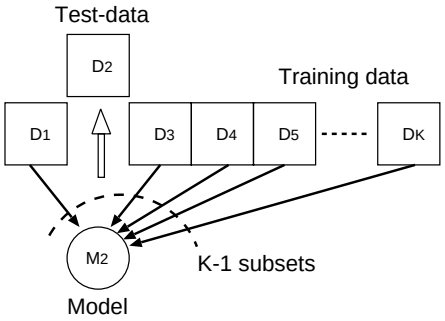

Fig. 1. K-fold CV. The data overlap between model parameter estimation and evaluation is avoided.

II. Hold-out and Cross-validation (CV) Methods

Hold-out method is used to tune some parameters or to evaluate a model using only a training set. For this, a subset of the training set is held out for evaluation and the rest of the data is used for model training. A disadvantage of the hold-out method is that significant amount of the original data need to be used as the hold out subset for meaningful evaluation. This problem is solved by using K-fold Cross-validation (K-fold CV). It works by partitioning the original training data into K subsets. Of the K subsets, a single subset is retained as the validation data for evaluating the model, and the remaining $K-1$ subsets are used to estimate the model parameters. The process is shown in Figure 1. The process is repeated for K times changing the subset for the evaluation. The K evaluation results are then averaged and used as the overall evaluation score.

Since it makes the data exclusive for the model parameter estimation and for the evaluation, the CV evaluation score is mostly unbiased. The data fragmentation problem due to the data partitioning is minimized by choosing large K since each CV model is estimated using $\frac{(K-1)}{K}$ of the original training data. For example, if $K = 10$, 90% of the original training

data is used to estimate each CV model. When K is equal to the number of training samples, it is called leave-one-out CV.

Let $T = \{x_1, x_2, \cdots, x_N\}$ be a training set consisting of independently sampled N samples, M be a statistical model with a specific model structure, $\psi_M(x|T)$ be a log likelihood or more generally any kind of score for a data sample x by M whose parameters are estimated from T. Then, expected error e_{cv} of a leave-one-out CV score $\frac{1}{N}\sum_{x_t \in T} \psi_M(x_t|T \setminus \{x_t\})$ with respect to a model score based on the true data distribution $\mathop{\mathrm{E}}_{x}[\psi_M(x|T)]$ is expressed as Equation (1). By assuming that eliminating up to two training samples from T does not largely change the model score for x that is independent from variables in T, i.e., $\psi_M(x|T) \approx \psi_M(x|T \setminus \{x_i\}) \approx \psi_M(x|T \setminus \{x_i, x_j\})$ for arbitrary x_i and x_j in T, Equation (1) is re-written as Equation (2), which is equal to the expected error e_{dev} of an evaluation score based on an independent development set $D = \{x'_1, x'_2, \cdots, x'_N\}$ as shown in Equations (3) and (4). Therefore, leave-one-out CV has about the same generalization ability in model evaluation as the evaluation using a development set having the same size as the training set. The advantage of the leave-one-out CV method is that it does not actually require such extra data. Similar argument holds for K-fold CV with large K.

While the idea of CV is intuitive, its theoretical aspect does not necessarily well known. Some empirical as well as theoretical results with CV is found in [1].

III. Model trainings based on CV objective function

In the application of CV for parameter tuning and model selection, multiple models are made and evaluated changing some parameters to find the best configuration. Therefore, it is quite time consuming if many models are made from scratch. However, there are some applications where the CV score is efficiently maximized for a large number of possible configurations or even continuous optimization is performed. In the following, the Good-Turing smoothing, CV based decision tree clustering, and CV based GMM structure optimization are explained as such examples.

A. Ngram Smoothing

Ngrams are widely used as a language model in speech recognition. Usually, the number of possible Ngrams are very large and many Ngrams have no example in the training set. These Ngrams get zero probability in maximum likelihood estimation and cause a problem in evaluation. Ngram smoothing is used to solve the problem by discounting probability mass from observed Ngrams and re-distributing it to unseen Ngrams. There are many variations regarding how to discount the probability. Among them, Good-Turing smoothing developed by Good and Turing [2] is a widely known method.

Let's consider Ngram samples in a training set that consist of a word k following a fixed Ngram context c. Let $r = N(k)$ be the number of occurrences of a word k in the samples with the Ngram context and n_r be the number of kinds of word k whose occurrence count is r. With the Good-Turing smoothing, Ngram probabilities are estimated by Equation (5).

$$p_r = \frac{r}{N}\frac{(r+1)\,n_{r+1}}{r n_r}. \tag{5}$$

It has been shown that the Good-Turing smoothing can be regarded as a result of leave-one-out CV based optimization [3] optimizing continuous parameters. In the derivation, it is assumed that all words k with the same occurrence count r have the same Ngram probability $p_r = P(k|c)$. Because p_r is probability, it must satisfy $\sum_r p_r n_r = 1$. When removing a sample with a word k from the Ngram samples and using it as a held-out sample, the occurrence count of k in the remaining samples is one less than the original count. Thus, p_r is used for the word k that has counts $r+1$ in the original samples. Then, the leave-one-out log likelihood is expressed as Equation (6).

$$LL_{leave\_one\_out} = \sum_r (r+1)\, n_{r+1} \log p_r. \tag{6}$$

By maximizing Equation (6) using the method of Lagrange multipliers with the constraint $\sum_r p_r n_r = 1$, Equation (5) is obtained.

B. Decision Tree State Clustering

Context dependent HMM is widely used in large vocabulary speech recognition systems as an acoustic model. Decision tree HMM state clustering [4] is a top-down clustering method to optimize the state tying structure of the context dependent model for robust parameter estimation and for preparing phone contexts that do not appear in the training set.

In decision tree state clustering, a leaf corresponds to a set of HMM states to be tied. The tree growing process begins with a root node that may have all HMM states, or all states associated with a particular phone, etc. Then, a question is selected that divides the set of states into two subsets assigned respectively to two child nodes so that the corresponding new HMM has the largest objective score. The tree is grown in a greedy fashion, successively splitting nodes by selecting questions.

Applying a question corresponds to try an HMM with a specific tying structure. Therefore, a large number of HMM are estimated and evaluated during the clustering, and the computational cost is not practical if they are trained from scratch. In the conventional approach based on maximum likelihood criterion [4], it is assumed to reduce the computational cost that the state alignment does not change with different tying configurations. In this case, the likelihood change due to expanding the parameter set is simply given by the change in observation likelihoods of the impacted states. The model parameters and associated observation likelihoods can be computed efficiently by using the pre-computed sufficient statistics associated with each state in the model.

A limitation of the likelihood objective, however, is that it is guaranteed to increase as parameters are added, e.g. new nodes in the tree, since the splits are trained and evaluated using the same training data. Hence, the tree can potentially grow

$$
\begin{aligned}
e_{cv}(N) &= \mathop{\mathrm{E}}_{T,|T|=N}\left[\left(\frac{1}{N}\sum_{x_t\in T}\psi_M(x_t|T\setminus\{x_t\}) - \mathop{\mathrm{E}}_{x}[\psi_M(x|T)]\right)^2\right], && (1)\\
&\approx \frac{1}{N}\mathop{\mathrm{E}}_{T,|T|=N}\left[\mathop{\mathrm{E}}_{x}\left[\psi_M^2(x|T)\right] - \left(\mathop{\mathrm{E}}_{x}[\psi_M(x|T)]\right)^2\right], && (2)\\
&= \mathop{\mathrm{E}}_{T,|T|=N}\left[\mathop{\mathrm{E}}_{D,|D|=N}\left[\left\{\frac{1}{N}\sum_{x_d\in D}\psi_M(x_d|T) - \mathop{\mathrm{E}}_{x}\psi_M(x|T)\right\}^2\right]\right], && (3)\\
&= e_{dev}(N,N). && (4)
\end{aligned}
$$

until all states are untied. Furthermore, the decisions made in tree growing are unreliable when there are a small number of samples associated with a node. To deal with these problems, empirical thresholds are required such as minimum likelihood difference and minimum occupancy counts of a state [5].

These problems of the likelihood based method are due to the lack of a mechanism of balancing the number of parameters and accuracy of parameter estimation, which can be solved by using CV. CV based decision tree clustering was first proposed for semi-tied HMM [6], which has low computational cost for the clustering since the HMM states are expressed by a weight vector for a shared base Gaussians. However, recognition performance by the semi-tied HMM is limited.

CV based decision tree clustering for more general continuous Gaussian mixture HMM was proposed in [7]. The extension was made possible by estimating a CV score using a set of sufficient statistics similar to the technique used in [8]. Let D be a training set and D_f be a subset for N-fold CV. That is,

$$
D = \bigcup_{f=1}^{N} D_f, \qquad D_i \bigcap D_j = \phi \quad (i \neq j). \tag{7}
$$

For the f-th evaluation, $\bar{D_f} = \bigcup_{f'\neq f} D_{f'}$ is used to estimate HMM parameters and D_f is used to evaluate likelihood. Let S be a set of states, s be a state, $\gamma_s(t)$ be occupancy probability of state s at time t, and $\boldsymbol{x_t} = (x_1(t), x_2(t), \cdots, x_d(t))^T$ be d-dimensional feature vector at time t. Let $A_f^0(s)$, $\boldsymbol{A}_f^1(s)$ and $\boldsymbol{A}_f^2(s)$ be the sufficient statistics of the observations aligned to state s. For diagonal Gaussian distributions, these are:

$$
A_f^0(s) = \sum_{t\in D_f}\gamma_s(t), \tag{8}
$$

$$
\boldsymbol{A}_f^1(s) = \sum_{t\in D_f}\boldsymbol{x_t}\gamma_s(t), \tag{9}
$$

$$
\boldsymbol{A}_f^2(s) = \sum_{t\in D_f}\boldsymbol{x_t^2}\gamma_s(t), \tag{10}
$$

where $\boldsymbol{x}^2 = \left(x_1^2, x_2^2, \cdots, x_d^2\right)^T$.

Using these statistics, the f-th ML estimates of a mean vector $\boldsymbol{\mu}$ and a variance vector $\boldsymbol{v}$ of a state s on $\bar{D_f}$ are:

$$
\boldsymbol{\mu}_f(s) = \frac{\sum_{f'\neq f}\boldsymbol{A}_{f'}^1(s)}{\sum_{f'\neq f}A_{f'}^0(s)}, \tag{11}
$$

$$
\boldsymbol{v}_f(s) = \frac{\sum_{f'\neq f}\boldsymbol{A}_{f'}^2(s)}{\sum_{f'\neq f}A_{f'}^0(s)} - \boldsymbol{\mu}_f(s)^2. \tag{12}
$$

Let L_f be the log likelihood for f-th data fold using Gaussians that have means $\boldsymbol{\mu}_f(s)$ and variances $\boldsymbol{v}_f(s)$, as shown in equation (13), where $\boldsymbol{\Sigma}$ is a diagonal covariance matrix whose main diagonal is $\boldsymbol{v}$. By putting the summation over t inside and utilizing the assumption that $\boldsymbol{\Sigma}$ is a diagonal matrix, equation (13) can be efficiently evaluated using the precomputed statistics A^0, $\boldsymbol{A}^1$, and $\boldsymbol{A}^2$ as shown in equation (14), where $\boldsymbol{v}^{-1} = \left(v_1^{-1}, v_2^{-1}, \cdots, v_d^{-1}\right)$. Finally, the CV likelihood L is obtained by summing the likelihoods for each fold:

$$
L = \sum_{f=1}^{N} L_f. \tag{15}
$$

Originally, the CV based decision tree algorithm was evaluated for speech recognition and it has been shown to outperform conventional maximum likelihood method deciding the model size automatically [7]. It has also been applied for speech synthesis and reported to give better performance than MDL [9]. An extension to variational Bayes based clustering is proposed in [10] and a combination with hierarchical priors are proposed in [11].

C. Gaussian Mixture Optimization

The CV techniques based on the sufficient statistics used in the decision tree clustering can be directly applied to Gaussian mixture optimization. CV likelihood based Gaussian mixture HMM optimization is proposed in [12]. A concern when using CV in such structure optimization algorithms is that the number of models subject to the comparison is much larger than that in the traditional use of CV. While CV can mostly remove the bias in likelihood estimation, the CV score still has variance. Among the large number of models, there will be a model that gives a higher CV score just by chance irrespective of its true performance on new data. This effect increases with the number of models and degrades the model selection performance. To reduce the variance, aggregated cross-validation (AgCV) that introduces a bagging-like [13] idea to the cross-validation framework is proposed and applied

$$L_f = \sum_{t \in D_f} \sum_{s \in S} \log \left\{ \frac{1}{\sqrt{(2\pi)^d \left| \boldsymbol{\Sigma}_f(s) \right|}} \exp\left(-\frac{1}{2} \left(\boldsymbol{x}_t - \boldsymbol{\mu}_f(s) \right)^T \boldsymbol{\Sigma}_f(s)^{-1} \left(\boldsymbol{x}_t - \boldsymbol{\mu}_f(s) \right) \right) \right\} \gamma_s(t) \tag{13}$$

$$= -\frac{1}{2} \sum_{s \in S} \left\{ \log\left((2\pi)^d \left| \boldsymbol{\Sigma}_f(s) \right| \right) A_f^0(s) + \left(\boldsymbol{v}_f(s)^{-1} \right)^T \boldsymbol{A}_f^2 - 2 \left(\boldsymbol{\Sigma}_f(s)^{-1} \boldsymbol{\mu}_f(s) \right)^T \boldsymbol{A}_f^1 + \left(\boldsymbol{v}_f(s)^{-1} \right)^T \boldsymbol{\mu}_f(s)^2 A_f^0 \right\} \tag{14}$$

to Gaussian mixture HMM optimization [14]. It is reported that AgCV based optimization gives better performance than CV based optimization.

IV. Direct Selective Adaptation Methods

In the application of speech recognition, sometimes it is required to train a high performance model using a domain independent large training data and a domain dependent small adaptation data, which is a problem of adaptation or transfer learning [15]. For example, in a deployment of speech recognition systems, it is often required to train a high performance model using an existing large domain independent data and a small in-domain data. Other situation is that large amount of noisy data is available on the Internet together with a small amount of in-domain data.

Assuming that the training data is partitioned to many blocks by some means, a strategy for this problem is to select or weight the blocks by measuring their usefulness based on the adaptation data. Each block of the training data can be a sentence, a document, or a speaker, etc. Here, we refer such selective use of training data for adaptation as selective adaptation. The overall framework of selective adaptation is analogous to parameter tunings using a held-out set. Therefore, some techniques for this problem are related to the ones described in Section III as shown in the following.

With the selective adaptation, one way of selecting the training data is to make a model using the adaptation data. Each block is evaluated using the model and only blocks with likelihood higher than a threshold are selected. Then, a subset of the training data is formed gathering the selected blocks and a model is estimated using the subset. An example of this strategy is found in [16]. However, this strategy has an essential problem that the most frequent pattern in the adaptation data is excessively emphasized in the selected training subset. In other words, the distribution of the events in the selected subset does not match the distribution of the ones in the adaptation data [17]. This problem is due to the direction of model training and evaluation is opposite in the block selection and the final model training.

Another way is to consider all or many combinations of the blocks to enumerate possible subsets, estimate a model for each subset, evaluate the models for the adaptation data, and select the best model among all. In this strategy, a model that gives the highest score for the adaptation data is selected. We refer this strategy as direct selective adaptation.

Instances of this direct selective adaptation have been proposed both for acoustic modeling [18] and for language modeling [19]. A closely related method to [19] is the relative entropy method [17] in which the selection is based on relative entropy evaluated between language models trained on the subset and on the adaptation data. An extension of [19] is proposed in [20] introducing a weighting factor to compensate for unseen Ngrams. In these strategies, a common problem is that the number of subsets is very large. Therefore, it is not always feasible to train and evaluate a model from scratch for all the subsets. To reduce the computational cost, the HMM sufficient statistics are used in [18] and the Ngram counts are used in [20]. In [18], it is reported that the selective method gave better performance than MLLR [21] and MAP [22] adaptation. It is reported in [19], [17], [20] that these methods tend to make very compact Ngram models with superior performance than their baseline.

V. CV Evaluation Inside of Iterative Parameter Estimation

CV is usually used to select models or to evaluate model performance. However, some iterative parameter estimation algorithms include the evaluation process as a subroutine. There are some cases that it is useful to integrate CV into the parameter estimation process to improve model estimation performance. In this section, CV adaptation [23] and CV-EM [24] are explained that integrate CV in their parameter estimation process.

A. CV Adaptation

In speech recognition, unsupervised adaptation refers a semi-supervised learning where model parameters are adapted using unlabeled input speech utterances. Final recognition output for the utterances is made using that adapted model. The most popular way of doing this is the iterative unsupervised batch-mode adaptation.

Figure 2 shows the procedure. The first step is to decode a set of target utterances from a speaker using either an initial model if it is the first iteration or otherwise using an adapted model made in the previous step. By running a speech decoder, a hypothesized transcript is obtained. The second step is to perform model parameter updates based on the hypothesis. The details of how to update the parameters depend on underlying adaptation techniques, such as MLLR. For higher performance, this process is iterated several times [25], [26]. The final recognition result is obtained by outputting the hypothesis made in the last decoding step.

In this procedure, the over-fitting problem cannot be avoided; once a model parameter is biased to a specific data sample, the bias is reinforced in the subsequent adaptation iterations since the same data is used in the decoding step

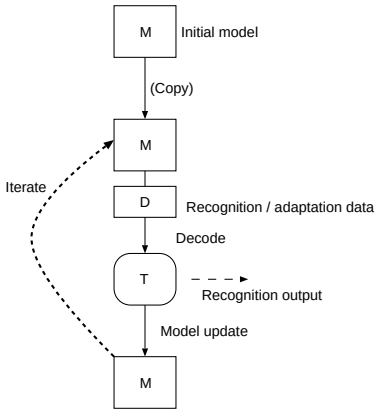

Fig. 2. Batch-mode unsupervised adaptation. M is a model, T is a recognition hypothesis, and D is an adaptation data. The hypothesis T is made from the data D using the model M. Using that hypothesis, parameters of the model M is updated. The updated model is used to recognize the same data in the next iteration.

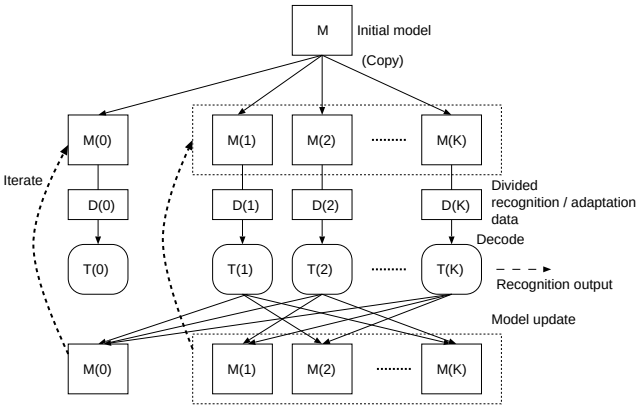

Fig. 3. Unsupervised CV adaptation. M is a model and D is a target data. $M(k)$ is a k-th CV model, $D(k)$ is a k-th exclusive data subset, and $T(k)$ is a recognition hypothesis of that subset using $M(k)$. The data used for the decoding step and for the model update step are separated. $M(0)$ denotes a global CV model and $T(0)$ denotes a hypothesis by $M(0)$.

and the model update step. Moreover, it is unavoidable to have recognition errors in the recognition hypothesis. These errors are also reinforced during the iteration. These problems decrease the efficiency of the adaptation.

The CV adaptation reduces the problems of the batch-mode adaptation by effectively separating the data used in the decoding step and in the model update step based on the K-fold CV, as shown in Figure 3. In this procedure, the target utterances are divided into K exclusive subsets $(D(1), D(2), \cdots, D(K))$ so that each subset has roughly the same size. The first decoding step is basically the same as the batch-mode adaptation and the K subsets are processed using the same initial model. Then, given the K recognition hypotheses $(T(1), T(2) \cdots, T(K))$, K CV models $(M(1), M(2), \cdots, M(K))$ are made by excluding one of the recognition hypotheses, instead of making a single model. As an initial model to estimate the k-th CV model, the k-th CV model of the previous stage is used. Each model is used in the next decoding step to make a new hypothesis for the data subset that has been excluded from the parameter estimation of that model. The decoding step and the model update step are repeated as in the conventional batch-mode adaptation and the final recognition hypothesis is obtained by gathering the hypotheses of the K subsets made in the last decoding step.

With this procedure, the data used for the decoding and for the model parameter estimation are effectively separated minimizing the undesired effect of reinforcing the bias. Because the utterances used for a model estimation are not decoded by that model, it is unlikely that the same recognition error is repeated by that model. The data fragmentation problem is minimal for large K, since $(K-1)/K$ of the data is used for the parameter estimation of each CV model. Optionally, a global CV model $(M(0))$ can be made in the update step together with the CV models by using all recognition hypothesis. The global CV model is useful when a single adapted model is required as an output of the adaptation process. It is reported in [23] that CV adaptation gives significantly better recognition performance than the conventional batch mode adaptation.

If $K = 2$, CV-adaptation is similar to the cross-adaptation [27]. The difference is that CV-adaptation is performed on a single recognition system whereas cross-adaptation requires two. While cross-adaptation uses transcripts from two systems representing different views of the same data based on different features and/or decoding algorithms etc, CV-adaptation uses the different data of the same view.

CV-adaptation is also similar to the co-training [28]. It requires two views of the same data as in the cross-adaptation, and it first learns a separate classifier for each view using labeled examples. The most confident predictions of each classifier on the unlabeled data are then added as part of the labeled data with the automatic classification results. This process is iterated several times. Co-EM [29] is a variation of Co-training where all the predictions are used with weightings. Actually, cross-adaptation might be regarded as a kind of variation of Co-training/Co-EM. However, there is a difference in motivation. Co-training/Co-EM are used when there are only small amounts of labeled data and large amounts of unlabeled data. On the other hand, cross-adaptation is used with a general model trained on large database and recognition data that is not necessarily very large. In cross-adaptation, the recognition data is also used as adaptation data.

B. CV-EM

Expectation Maximization (EM) algorithm [30] is an iterative maximum likelihood parameter estimation algorithm that is widely used when a model contains hidden variables. It consists of Expectation step (E-step) and Maximization step (M-step) that are repeated reciprocally. The expectation step is based on probabilistic inference on the hidden variables given observations. Therefore, it is roughly speaking a decoding process. CV-EM is an algorithm that integrates CV in the framework of EM to reinforce the decoding process in the E-steps [24].

The CV adaptation can be regarded as an special case of CV-EM with some constraints to variables and with Viterbi-training [31] like approximation. The procedure of CV-EM is similar to a MMI training algorithm that incorporates cross-validation [32]. The difference is that while Gradients are estimated using CV in the MMI training, sufficient statistics are estimated in CV-EM. CV-EM has been applied to speech recognition in [24] and it has been shown to be very robust for over-fitting and gives higher recognition accuracy than conventional EM. Some theoretical analyses on CV-EM can be found in [33] about how CV-EM prevents the overfitting.

VI. Conclusions

Controlling data dependency is an important principle in statistical model training and evaluation. In order to correctly evaluate model performance and to tune some type of model parameters, a data set that is independent from the training data is needed. For this purpose, a separate development data or hold-out/CV methods are widely used.

In this paper, model training methods are reviewed that integrate the hold-out or the CV methods in an advanced manner. These include the Good-Turing smoothing, CV based decision-tree HMM state clustering, CV based Gaussian mixture optimization, CV adaptation, and CV-EM. It has also been shown that the selective adaptation methods that select a subset of training data based on a development set share a similar framework as the hold-out method and thus they share some techniques used in the CV methods. These methods are effective to improve speech recognition performance. Future works include application of these techniques to other problems such as speaker recognition and model based feature extraction. Theoretical analysis of these methods will be useful to further improve the recognition performance.

References

[1] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[2] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3/4, pp. pp. 237–264, 1953.

[3] H. Ney, U. Essen, and R. Kneser, "On the estimation of 'small' probabilities by leaving-one-out," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 1202–1212, December 1995.

[4] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.

[5] S. Young *et al.*, *The HTK Book*, Cambridge University Engineering Department, 2005.

[6] I. Rogina, "Automatic architecture design by likelihood-based context clustering with crossvalidation," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 1223–1226.

[7] T. Shinozaki, "HMM state clustering based on efficient cross-validation," in *Proc. ICASSP*, Toulouse, 2006, vol. I, pp. 1157–1160.

[8] M. Ostendorf and H. Singer, "HMM topology design using maximum likelihood successive state splitting," *Computer Speech and Language*, vol. 11, pp. 17–41, 1997.

[9] Z. Yu, Y. Zhi-Jie, and F. K. Soong, "Cross-validation based decision tree clustering for hmm-based tts," in *ICASSP*, march 2010, pp. 4602–4605.

[10] K. Hashimoto, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "Bayesian context clustering using cross validation for speech recognition," *IEICE Transactions on Information and Systems*, vol. 94, no. 3, pp. 668–678, 2011.

[11] H. Zen and M. J. F. Gales, "Decision tree-based context clustering based on cross validation and hierarchical priors," in *Proc. ICASSP*, May 2011, pp. 4560–4563.

[12] T. Shinozaki, S. Furui, and T. Kawahara, "Gaussian mixture optimization based on efficient cross-validation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 3, pp. 540–547, June 2010.

[13] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[14] T. Shinozaki, S. Furui, and T. Kawahara, "Aggregated cross-validation and its efficient application to Gaussian mixture optimization," in *Proc. Interspeech*, 2008, pp. 2382–2385.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, October 2010.

[16] R. Nisimura, K. Komatsu, Y. Kuroda, K. Nagatomo, A. Lee, H. Saruwatari, and K. Shikano, "Automatic n-gram language model creation from web resources," in *Proc. Eurospeech*, 2001, pp. 2127–2130.

[17] A. Sethy, P. Georgiou, and S. Narayanan, "Text data acquisition for domain-specific language models," in *Proc. EMNLP*, 2006, pp. 382–389.

[18] T. Cincarek, T. Tomoki, H Saruwatari, and K. Shikano, "Utterance-based selective training for the automatic creation of task-dependent acoustic models," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 3, pp. 962–969, 2006.

[19] D. Klakow, "Selecting articles from the language model training corpus," in *ICASSP*, 2000, vol. 3, pp. 1695 –1698.

[20] T. Shinozaki, Y. Kubota, S. Furui, E. Utsunomiya, and Y. Shindoh, "Sentence selection by direct likelihood maximization for language model adaptation," in *Proc. Interspeech*, 2011, To appear.

[21] C. J. Leggetter and P. C. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in *Proc. Eurospeech*, 1995, pp. 1155–1158.

[22] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[23] T. Shinozaki, Y. Kubota, and S. Furui, "Unsupervised acoustic model adaptation based on ensemble methods," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1007 –1015, Dec. 2010.

[24] T. Shinozaki and M. Ostendorf, "Cross-validation and aggregated EM training for robust parameter estimation," *Computer speech and language*, vol. 22, no. 2, pp. 185–195, 2008.

[25] P. C. Woodland, D. Pye, and M. J. F. Gales, "Iterative unsupervised adaptation using maximum likelihood linear regression," in *Proc. ICSLP*, oct 1996, vol. 2, pp. 1133–1136.

[26] S. Nakagawa, T. Watanabe, H. nishizaki, and T. Utsuro, "An unsupervised speaker adaptation method for lecture-style spontaneous speech recognition using multiple recognition systems," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 3, pp. 463–471, 2005.

[27] H. Soltau, B. Kingsbury, L. Mangu, D. Povery, G. Saon, and G. Zweig, "The IBM 2004 conversational telephony system for rich transcription," in *Proc. ICASSP*, 2005, vol. I, pp. 205–208.

[28] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," *Proceedings of the eleventh annual conference on Computational learning theory COLT 98*, vol. 98, pp. 92–100, 1998.

[29] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the ninth international conference on Information and knowledge management*. 2000, CIKM '00, pp. 86–93, ACM.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. of the Royal Statistical Society*, , no. Series B 39, No. 1, pp. 1–38, 1977.

[31] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[32] N. S. Kim and C. K. Un, "Deleted strategy for MMI-based HMM training," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, pp. 299–303, 1998.

[33] T. Takenouchi and K. Ikeda, "Theoretical analysis of cross-validation(cv)-em algorithm," in *Proceedings of the 20th international conference on Artificial neural networks: Part III*, 2010, ICANN'10, pp. 321–326.