Abstract

Recently, an implicit trajectory model using temporally varying weight regression (TVWR) was proposed and achieved promising gains using ML training criteria. In the original TVWR, each component weight is modelled as a constrained linear regression function with respect to the monophone posterior feature. Due to the high dimensionality of the posterior feature, many free parameters were introduced into the TVWR system. Compared to the standard HMM system, such increment of system complexity could potentially cause two issues: over-training and slow decoding. In order to avoid these two issues, parameter clustering for TVWR is proposed to estimate cluster specific instead of original component specific regression parameters. In this paper, both knowledge-driven and data-driven approaches are introduced to define the cluster. Parameter re-estimation of clustered regression parameters is also derived. Experiments are conducted on the clean data of Aurora 4 corpus and systems are evaluated on Nov’92 5k closed vocabulary recognition task. Results show that comparable performance can be obtained and decoding time improves by more than 20% after significant reduction of system complexity.

Index Terms: trajectory modelling, complexity control, linear regression, clustering

1. Introduction

Recently, a new implicit trajectory model called Temporally Varying Weight Regression (TVWR) [1] was proposed to relax the Hidden Markov Model (HMM) assumption [2], which is that the current observation is independent of the other states and observations given the current state. Original TVWR [1] tried to model each component weight as a constrained linear regression function with respect to a monophone posterior feature. Since the monophone posterior feature used in TVWR is predicted based on a long span of context observations, rich temporal varying information is able to be properly regressed to represent time-varying Gaussian mixture model (GMM) [3].

Like many other trajectory models [4, 5, 6, 7] that have time-varying property with high dimensional posterior features, significant performance improvements over the conventional HMM are obtained at the cost of greatly increasing number of parameters.

The main additional computing cost of TVWR system is from the regression process, which consists of predicting the monophone posterior feature and linear regression. The monophone posterior feature can be predicted from any probabilistic classifier by taking a long span of context observations as input. For a better accuracy, powerful classifiers, such as neural networks or GMM with full covariance matrix, are preferred. In case of using neural networks in the original TVWR [1], there is no way to reduce the computing cost given the fixed input dimension and neural network configuration. If GMM is applied, Gaussian clustering and selection approach used in fMPE [4, 5] may be applied for efficiency. Although reduction of the posterior feature dimension can decrease the system complexity, it could also lose a lot of temporal contextual information and decrease the system performance. Since time-varying weight of each component in TVWR is a linear function of the posterior feature, the number of regression parameters can be tens of thousands in a state-of-the-art system, which could lead to inefficient decoding. On the other hand, too many free parameters may cause over-training issue as the training data and computing resources can always be limited. Considering the possibility of that some components may share similar contextual information, this paper proposes parameter clustering to estimate a cluster specific regression parameter tied to multiple components for a more compact TVWR model. In detail, clustered parameter re-estimation using the maximum likelihood criterion is derived; both knowledge-driven and data-driven clustering approaches are introduced.

The remaining of this paper is organized as follows. Section 2 gives an overview of previously proposed TVWR. Section 3 introduces the idea of parameter clustering and how to estimate clustered regression parameter. Experimental results are reported in Section 4.

2. Overview of TVWR

TVWR was originally proposed to relax the HMM assumption by explicitly modelling the correlation of limited successive observations into the time-varying weight. Mathematically, TVWR is formulated by factorizing a standard HMM model using a long span of observations as features such as the output probability of state $j$ in TVWR is shown as:

$$p(o_t, \tau_t | j) = \sum_{m=1}^{M} P(m|j)p(\tau_t|o_t, j, m) p(o_t | j, m)$$  \hspace{1cm} (1)$$

where $\tau_t$ is a limited context of observation $o_t$, denoted as $\tau_t = \{o_{t-\delta}, \ldots, o_{t-1}, o_{t+1}, \ldots, o_{t+\delta}\}$, where $\delta$ is the context expansion size. We name the static weight $P(m|j)$ scaled by the conditional probability of the context variable $\tau_t$ as the time-varying weight $c_{jmt}$. The advantage of such factorization is that the conventional component emission probability $p(o_t | j, m)$ remains with efficient parameter estimation formulae but its weight is expressed to be time-varying or context dependent. Since the conditional probability of $\tau_t$ is very expensive to model due to the high dimensionality of $\tau_t$, two ap-
proximations are made such that:
\[
p(\tau_i | \alpha_i, j, m) \approx p(\tau_i | j, m) = \sum_{i=1}^{N} p(\tau_i | j, m) P(i | j, m) \approx K_i \sum_{i=1}^{N} p(i | \tau_i) P(i | j, m)
\]
where \( K_i = p(\tau_i) / P(i) \) is the component independent normalization term, which can be ignored during likelihood calculation, and \( i \) is the latent variable to partition the space of context observations and uniform prior \( P(i) \) is assumed during the above derivation. \( p(i | \tau_i) \) is the posterior feature predicted by any probabilistic classifier using the context observation as input, \( P(i | j, m) \) is the so-called regression parameter to estimate during the conventional TVWR training. Note that due to the use of dynamic features, the independence assumption between \( \alpha_i \) and \( \tau_i \) may not be good, however, this assumption simplifies the model derivation. Typically, \( i \) is defined as monophone such that a neural network or GMM can be trained by supervision for a better quality of the posterior feature. For convenience, \( h_{ji} = p(\tau_i) \), \( w_{jmi} = P(i | j, m) \), and \( c_{jm} = P(m | j) \) are defined for future reference. As can be seen, each component \( jm \) requires \( N \) parameters to linearly regress the incoming posterior feature, which introduces too many parameters.

In the original TVWR [1], monophone classes are used to represent the latent variable \( i \), whose number, \( N = 40 \) is close to the typical feature dimension. Therefore, in a typical TVWR system with diagonal covariance matrix, if the total number of states is \( J \) and number of component per state \( M \), the additionally introduced parameters in TVWR would be \( JMN \) (Note that the transition and posterior generator related parameters and calculation are not considered in this paper), which is about 50\% more parameters to its initial standard HMM system. One simple way to reduce the system complexity is to lower the dimensionality of the posterior feature. However, since IMPE [4, 5] used thousands dimensional Gaussian posterior feature to learn the time-varying property, further reduction of the monophone posterior feature used in TVWR may lose a lot of important temporal information. It is noted that the regression parameter in Eq-4 tries to learn the component specific characteristics of the context variable. Considering that there is high possibility that some components share some similar acoustic context information, the same regression parameter may be sharable by multiple components without invalidating the sum-to-one constraint.

3. Parameter Clustering

The basic idea of parameter clustering for TVWR is to estimate a cluster specific regression parameter to be applied to all the components in that cluster. Therefore, two issues need to be solved in this section, including re-estimating tied regression parameter from the training data and determining which group of components should be tied to a particular cluster. Before discussing those details, regression parameter of cluster \( r \) is defined as \( w_{jmi}^{(r)} \) with following property:
\[
w_{jmi}^{(r)} = w_{jmi} \hspace{1cm} \forall jm \in r
\]
If there are totally \( R \) clusters, there would be \( RN \) additional parameters in TVWR, which could significantly reduce the system complexity given the fact that \( R \ll JM \). Note that the constraint still applies for clustered regression parameter such as:
\[
\sum_{i=1}^{N} w_{i}^{(r)} = 1, \hspace{0.5cm} w_{i}^{(r)} \geq 0 \hspace{1cm} \forall r
\]

3.1. Parameter Estimation

Since the objective function of training cluster and unclustered TVWR system will show how parameter clustering will affect the training criteria, or the performance in the training data, we will introduce the parameter estimation algorithm first. In this paper, only maximum likelihood estimation for TVWR systems will be applied. Given approximations in Eq-2 and Eq-4, the auxiliary function w.r.t. \( w_{jmi}, c_{jm} \) for unclustered TVWR system can be derived based on the conventional EM algorithm:
\[
Q = \sum_{i,j,m} \gamma_{jm}(t) \log c_{jm}
\]
\[
\approx \sum_{i,j,m} \gamma_{jm}(t) \log \left( c_{jm} \sum_{i} w_{jmi} p(\tau_i | j, m) \right)
\]
\[
\geq \sum_{i,j,m} \gamma_{jm}(t) \left( \log w_{jmi} + \log p(\tau_i | j) \right)
\]
\[
+ \sum_{i,j,m} \gamma_{jm}(t) \log c_{jm}
\]
where \( \gamma_{jm}(t) \) is the component occupancy given the current TVWR model, and
\[
\gamma_{jm}(t) = \frac{w_{jmi} p(\tau_i | j, m, \hat{\lambda})}{\sum_{\tau} w_{jmi} p(\tau_i | j, m, \hat{\lambda})}
\]
\[
\approx \frac{w_{jmi} p(\tau_i | j, \hat{\lambda})}{\sum_{\tau} w_{jmi} p(\tau_i | j, \hat{\lambda})}
\]
\[
= \gamma_{jm}(t) \frac{w_{jmi} h_{ji}}{\sum_{\tau} w_{jmi} h_{ji}}
\]
\( \hat{\omega}_{jmi} \) are the current model parameters. After dropping the term independent of regression parameter \( w_{jmi} \), the auxiliary function for each component becomes:
\[
G^{(jm)} = \sum_{i} \beta_{jmi} \log w_{jmi}
\]
where the sufficient statistics are:
\[
\beta_{jmi} = \sum_{t} \gamma_{jm}(t)
\]
Closed form update formula can be derived using Lagrange multiplier method such that:
\[
w_{jmi} = \frac{\beta_{jmi}}{\sum_{jmi} \beta_{jmi}}
\]
After tying regression parameter to some predefined clusters, the objective for each cluster \( r \) becomes maximizing the following auxiliary function:
\[
G^{(r)} = \sum_{jm \in r} \sum_{i} \beta_{jmi} \log w_{i}^{(r)}
\]
subject to the constraint given in Eq-6. Again, Lagrange multiplier method can be applied to derive the update formula for clustered regression parameters:
\[
w_{i}^{(r)} = \frac{\sum_{jm \in r} \beta_{jmi}}{\sum_{jm \in r} \sum_{i} \beta_{jmi}}
\]
As can be seen, the accuracy loss in the training data by tying regression parameter can be measured by the likelihood loss after switching objective function from Eq-11 to Eq-14. On the other hand, update formulae in Eq-13 and Eq-15 also tell that the components in the same cluster should have similar component specific sufficient statistics $\beta_{j, m}$, such that the likelihood loss can be minimized.

3.2. Knowledge-driven Clustering

Knowledge-driven clustering is based on the possible connection of components in terms of time-varying weights to empirically determine which group of components could be possibly tied together. The very first approach is to tie all the components into one global cluster. In that case, all the components will share the same time-varying regression value, however, this does not offer useful information to distinguish different components or states given different context information. Therefore, TVWR with one global cluster will degenerate to be a standard HMM system. In case of using monophone class to represent the latent variable $i$, components with the same central monophone may learn some similar time-varying information characterized by monophone posterior feature. It is possible to tie these components such that $R = N$ and we name this approach as model-wise clustering. However, model-wise clustering may not be consistent to the typical triphone system using left and right phone as context, which may lead to a poor context model. One alternative is to cluster all the components within the same triphone state such that each state has its own regression parameter. In that case, we have $w_{j, m} = w_{j, m}^*$ and $R = J$, and name it as state-wise clustering. Since there is no need of any prior knowledge about TVWR systems, these approaches can be performed without pre-training of TVWR.

3.3. Data-driven Clustering

Since knowledge-driven clustering is not able to perform the clustering at the component level, data-driven clustering is motivated. For a better measurement of the clustering quality, the auxiliary function Eq-14 of clustered TVWR system is employed as the objective function of data-driven clustering algorithm. However, it is computationally expensive to evaluate this objective function since the sufficient statistics $\beta_{j, m}$ may change after tying regression parameter. Therefore, forward-backward alignment of clustered TVWR is assumed to be fixed during the clustering process. Since the update step is already shown in Eq-15, remaining problem of this clustering algorithm is how to assign each component to a particular cluster. Given a component $j, m$ with its sufficient statistics $\beta_{j, m}$, the assignment can be performed by selecting a cluster $r$ which contributes most likelihood to the auxiliary function Eq-14 such as:

$$j, m \in \arg \max_r \left\{ \sum_i \frac{w_{j, m}^*}{w_{j, m}} \log w_i^{(r)} \right\}$$

Finally, the whole data-driven clustering algorithm can be summarized as follows:

1. Perform a regular TVWR training to accumulate sufficient statistics in Eq-12 for each component.
2. Estimate a GMM with the desired number of components (clusters) to describe the distribution of regression parameters $(w_{j, m}^*)$ and use the component mean as the initialization of the cluster centroids.
3. Perform the assignment step for each component according to Eq-16 given the current cluster centroids.
4. Update the centroid of each cluster according to Eq-15 using the fixed sufficient statistics.
5. If the auxiliary function in Eq-14 converged or the maximum number of iterations reached, goto step 6; otherwise, goto step 3.
6. Re-estimate the cluster centroids through the training data using the update formula in Eq-15 by few iterations of clustered parameter estimation.

3.4. Runtime Analysis

Caching the cluster dependent $\sum_i w_{j, m}^* h_{ij}$ is the most important step to really take advantage of the compressed TVWR system. Therefore, in a clustered TVWR system, likelihood calculation of one component has chance to directly apply the regressed value from its cluster. If pruning threshold and beam search are disabled, for a utterance with $T$ frames, unclustered TVWR requires $O(TMN)$ more operations than that of its initial standard HMM system. After clustering, only $O(TRN)$ more operations are required. In a typical state-of-the-art ASR system, the total number of Gaussian components can be tens of thousands, if a relative small number of clusters can maintain the similar gain over the standard system, a lot of computing time can be saved. Moreover, for a fast implementation, regression value is usually calculated on demand but always cached for possible reference by other components, which is another key point to really improve the efficiency for both training and decoding.

4. Experimental Results

In this section, experiments are conducted on the clean data of Aurora 4 [8] corpus, which contains about 14 hours training data, including 7138 utterances. The evaluation set is 5k closed vocabulary recognition task (NIST Nov'92 WSJ0), including 330 utterances. Decision tree state-clustered triphone system with $J = 3226$ distinct states is built with each triphone modelled by a 3 state left-to-right HMM. The acoustic features used here is 39 dimensional MFCC, including 12 static coefficients, zero-th coefficient and the first two differentials. The recognition includes a bigram full decoding followed by a trigram lattice-rescoring using HTK [9].

The temporally varying posterior feature, $h_{tr}$, is obtained using multiple GMM models. Specifically, $N = 40$ monophones are employed to represent the latent class $i$, each of which is modelled by a GMM with 8 components and full covariance matrix. The long span context observations include a sequence of MFCC features spanning a window of 8 frames. Since the linear relationship exists between the static and dynamic parameters, which will lead to singular full covariance matrix, only static parameters are applied with 13*8 total dimensions.

<table>
<thead>
<tr>
<th>Models</th>
<th>$R=1$</th>
<th>$R=40$</th>
<th>$R=3226$</th>
<th>$R=51616$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER(%)</td>
<td>6.35</td>
<td>6.00</td>
<td>5.98</td>
<td>3.75</td>
</tr>
</tbody>
</table>

Table 1: Comparison of recognition performance for knowledge-driven clustered TVWR and baselines.

Recognition results of baselines and knowledge-driven clustered systems are compared in Table 1. The standard HMM
system \((R = 1)\) with \(M = 16\) components per state shows 6.35\% Word Error Rate (WER). Note that no improvements have been found using more mixtures per state under the current setup. Starting from this baseline, the standard TVWR \((R = 51616)\) system shows a significant (tested at 5\% level using NIST SCTK scoring toolkit) improved performance, 5.75\% WER. However, as the component dependent regression parameter dimension, 40 is comparable to the Gaussian mean variance, it has increased the system complexity by 50\%. When model-wise clustering \((R = 40)\) is performed, the clustered TVWR system achieved slightly better recognition accuracy than the standard HMM. This is expected as the monophone posterior feature performed as a secondary classifier to help the speech recognition. When state-wise clustering \(R = 3226\) is performed, this system with much more parameters just performs similar to the model-wise clustering. Since both knowledge-driven clustering methods cannot significantly outperform the standard HMM as the regular TVWR, it is necessary to study the component level clustering in a data-driven fashion.

Figure 1: Comparison of "Likelihood" (Top), Word Error Rate (Middle) and HTK Runtime factor (Bottom) for various data-driven clustered TVWR systems with or without re-estimation. Knowledge-driven clustering based results are also marked as * at \(R=40\) and 3226.

Various criteria are compared with different number of clusters based on data-driven approaches in Fig.1. In the top figure, average likelihood with the same normalization term offset in all TVWR systems is compared. First observation is that re-estimation after clustering consistently increased the “likelihood”, which tells that parameter estimation algorithm of clustered TVWR is well implemented. After increasing the number of clusters, “likelihood” constantly increased and soon converged to be close to the unclustered TVWR. This implies clustered TVWR systems are potentially capable of describing the same characteristics of the contextual information like the unclustered TVWR. On the other hand, TVWR with more than 1024 clusters begin to show better training likelihood than the regular TVWR, which tells that TVWR may be more compactly modelled after tying parameters given the limited training data. Finally, when comparing knowledge-driven clustering approaches, data-driven method show consistently better “likelihood”.

Recognition results are compared in the middle figure. Clustered TVWR systems without further re-estimation perform consistently better than the standard HMM system but worse than the unclustered TVWR system. After 8 iterations of re-estimation after clustering, most clustered TVWR systems show promising improvement over both the standard HMM system and their starting points. It is also found that re-estimation did not always improve the system with fewer clusters. Since the forward-backward alignment is assumed to be unchanged during clustering, this assumption could be weakly made for those with fewer clusters possibly due to greater model changes. Therefore, data-driven clustered TVWR systems with fewer clusters may not be improved by re-estimation in terms of recognition performance. On the other hand, knowledge-driven approaches using \(R=40\) and 3226 did not show clear difference, however, data-driven approach can potentially gain more than the knowledge-driven approach. This tells that intra-state regression parameters can be very different such that data-driven approach with more flexible clustering choices can perform better. Eventually, the clustered TVWR system with 1024 clusters shows comparable performance to the unclustered system, which implies that many regression parameters can be shared without affecting the recognition performance.

In the bottom figure, HTK runtime factor, i.e. computing time per second speech, is compared. As can be seen, all the clustered TVWR systems are much more efficient than the unclustered system, and re-estimation also showed consistently improvement. Both data-driven and knowledge-driven approaches perform similarly efficient at \(R=40\) and 3226. Note that since beam search is always enabled for efficient decoding, more clusters does not always lead to more computing time. Eventually, the clustered TVWR system with the best recognition performance, i.e. \(R=1024\) also shows more than 20\% relative improvement in terms of decoding speed.

In summary, data-driven clustered TVWR with 1024 clusters has shown better performance than knowledge-driven approaches in terms of various criteria. It is also found that 1024-cluster TVWR system is able to maintain the recognition accuracy of the unclustered TVWR system and improve decoding efficiency. Although development set is missing for determining the proper number of clusters, it is empirically found that parameter clustering algorithm should be performed by choosing a minimum cluster number which can catch up the “likelihood” of unclustered TVWR system.

5. Conclusions

This paper has proposed parameter clustering using the maximum likelihood criterion to compress the recently proposed Temporally Varying Weight Regression (TVWR) model in the model space view. Parameter estimation of clustered TVWR system through the training data is derived; two clustering methods are also introduced. Experimental results on Aurora 4 corpus show that both “likelihood” and Word Error Rate can be retained, and decoding efficiency was significantly improved using data-driven clustering followed by parameter re-estimation.
6. References


