

CS753 : Automatic Speech Recognition

Assignment 3

Arka Sadhu
140070011

Abstract

The problem statement considered here is that given an audio recording from a radio station we want to isolate the parts of the recordings that correspond to human speech. This will be useful for further downstream tasks like sentiment classification, audio summarization etc. The dataset we have includes audio clips annotated with intervals that contain human speech. Here we present two methods for this task. First is the use of Conditional Random Fields (CRF) which uses a discriminative model as opposed to the generative model used by Hidden Markov Models (HMM) and also circumvents bias problems inherent in other discriminative models like Maximum Entropy Markov Models (MEMMs). Second is the use of Segmental Recurrent Neural Networks which defines a joint probability distribution over segmentations of the input and labellings of the sequence.

Problem Statement

We define the observation sequence as observation vectors upto time T (which is assumed to be fixed) as $O = [o_1, o_2, \dots, o_T]$. We also assume that we already know the mapping from the observation vector index to the actual speech duration which is to say that we know some observation vector o_i would correspond to the duration $t_i : t_j$. Therefore our only task is to label the observation sequence vectors. The label space we consider here is $l = \{l_h, l_n\}$ with l_h corresponding to human speech and l_n corresponding to anything other than human speech which we are not bothered with. So our task reduces to predicting the label sequence $Y = [y_1, y_2, \dots, y_T]$ such that each $y_i \in l$. Another formulation we introduce is to explicitly group together consecutive same labels. In this case we would like to predict the sequence $Z = [z_1, z_2, \dots, z_p]$ and $Y = [y_1, y_2, \dots, y_p]$ such that $z_i \in \mathbb{Z}_+$ and $y_i \in l$ where z_i represents the duration of a segment and y_i is the label corresponding to the segment. The first formulation is used in the conditional random fields and the second formulation is used in the segmental recurrent neural networks. The training data which is available to us is audio clips with intervals of annotated human speech. Thus for training we can get the data for either formulation.

Methodology

Here we describe both methods Conditional Random Fields (CRF) (Lafferty, McCallum, and Pereira 2001) and Segmental Recurrent Neural Networks (sRNN) (Kong, Dyer, and Smith 2015) in more detail.

Conditional Random Field (CRF)

Conditional Random field is a discriminative framework which is able to model the sequence of data better in the sense of relaxing strong independence conditions in Hidden Markov Models (HMM) and overcoming the label bias problem inherent in Maximum Entropy Markov Models (MEMMs).

The independence condition in HMM is required because we are modeling the joint probability distribution $P(Y, X)$. Here Y is the label sequence and X is the input sequence. It is necessary to make the independence assumption for the probability distribution to be tractable. CRF overcomes this by instead modelling $P(Y|X)$ which is the probability distribution of Y conditioned over X and not explicitly modeling the distribution $P(X)$. CRF gets away by not modeling $P(X)$ because at test time the observations are anyway fixed. This also allows for context dependent information to be captured because the label sequence Y is no longer needed to be decided only by the current observation.

The paper also compares CRF to other discriminative framework like MEMMs and claims like models like MEMMs are inherently biased towards states which have smaller number of output branches. The reason for this is that when it is at a particular state the competition of the next state is only from the current state. In other words we are only comparing transitions to the next state from the current state instead of looking at every other transition from any state to any other state (including self loops as well). To avoid this either the finite state machine can be determined but this can lead to explosion of number of states or to start with fully connected model but that doesn't allow us to use our domain knowledge or priors. This problem is circumvented in CRF because the individual transitions do not explicitly choose the transition rather only amplify or dampen the probability mass they receive.

Actual Model Let X be the random variable over data sequence and Y be the random variable over label sequence.

Even though X and Y are jointly distributed we will try to get the probability distribution $p(Y|X)$ and we choose not to model explicitly the distribution $p(X)$ as it is already fixed during runtime.

We consider a graph $G = (V, E)$ characterized by vertex set V and edge set E and let $Y = (Y_v)_{v \in V}$ be indexed by the vertices of the vertex set. Then we define (X, Y) to be a conditional random field if the random variables Y_v when conditioned on X obey the Markov property with respect to the graph: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \in \text{nb}(v))$. Here $\text{nb}(v)$ is the set of vertices one-hop distance away from the vertex v or in the 1-neighbourhood of v .

An important thing to note is that the CRF is globally conditioned on the variable X . Since our primary task is to build the ASR system to extract human speech we consider a specific case of the graph G which is a simple chain with T vertices and each consecutive vertices are joined by a single edge. Let $X = (X_1, X_2, \dots, X_T)$ and $Y = (Y_1, Y_2, \dots, Y_T)$. Now using the fundamental theorem of random fields we know :

$$p_\theta(y|x) \propto \exp\left(\sum_{e \in E} \lambda_k f_k(e, y_e, x) + \sum_{v \in V} \mu_k g_k(v, y_v, x)\right) \quad (1)$$

Here x is the whole sequence data, y is a label sequence and $y_{v/e}$ is the set of components associated with the vertex or edge in other words the subgraph consisting of the vertex. For edge it will be simply the two vertices being connected. We assume that f and g are fixed and the parameter set is the $\{\lambda_i, \mu_j | i \in E, j \in V\}$ which will be estimated from the training data.

We note here that specific choice of f which depends only the transition probabilities and g which depends on the observation probability, we can get HMMs that is to say, HMMs can be considered as a special class of CRF and in general CRF is much more expressive. We also note that CRFs have inherently convex loss functions.

As mentioned earlier CRF looks at the set of all transition probabilities at once. While in HMM the transition probabilities considered at a particular state is deciding which is the next state to go, in the case of CRF it is considering all transitions from any state x to any other state y . For convenience we consider first and last state as start and stop state respectively. So we have a matrix of transition probabilities at each vertex (our graph is a chain) as M_i which is of size $m \times m$ where m is the number of labels. We define

$$M_i(x) = [M_i(y', y|x)]$$

which is the probability of the particular transition happening at vertex i .

$$M_i(y', y|x) = \exp(\Lambda_i(y', y|x))$$

$$\Lambda_i(y', y|x) = \sum_k \lambda_k f_k(e_i, y_{e_i}=(y', y), x) + \sum_k \mu_k g_k(v_i, y_{v_i}=y, x)$$

The second formula follows from (1). Clearly the normalizing factor to make it a probability distribution will be the product of all the matrices

$$Z_\theta = \prod_{i=1:n+1} M_i(x)$$

Here we use θ to denote that there are parameters to be learnt. This gives us

$$p_\theta(y|x) = \frac{\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|x)}{\prod_{i=1}^{n+1} M_i(x)}$$

Segmental Recurrent Neural Networks (sRNN)

Task

Here we describe how the two models explained can be used for the speech extraction task described in the problem statement.

Conditional Random Field (CRF)

The task to be accomplished is described in the problem statement. For CRF we use the first formulation that is we have the observation vector $x = (x_1, \dots, x_T)$ and we need to get the label sequence $y = (y_1, \dots, y_T)$. We consider both training and testing.

Training The parameters of the CRF are the M matrices which consequently depend on the parameters $(\lambda_i, \mu_j | i \in E, j \in V)$. The training data consists of segmented data with corresponding labels of being a human speech or not.

To learn the parameters we use the algorithm Improved Iterative Scaling Algorithms (IIS). The exact derivations of these are slightly involved and hence it is not being covered in the report. The overall idea is that the scales that are used in the updates is obtained by equating the expectation of a particular feature with the observed expectation. This step requires the calculation of the **Total feature count** and this comes in the exponent and calculating the exponential sums in often problematic.

Thus there are two algorithms working together Algorithm S and T. Algorithm S has a new feature called the slack feature so that we don't have to keep track of the exact total feature count and yet get away with it. The slack feature is given simply by

$$s(x, y) = S - \sum_i \sum_k f_k(e_i, y_{e_i}, x) - \sum_i \sum_k g_k(v_i, y_{v_i}, x)$$

and S is chosen so that $s(x^{(i)}, y) \geq 0$ for $x^{(i)}$ seen in training data. The slack feature is global feature and doesn't belong to any one vertex. Algorithm T on the other hand remembers the partial T totals.

Now quite similar to the Baum-Welch algorithm, we define forward and backward vectors to deal with the large number of parameters. This is done for efficient computations of the matrices $M_i(x)$ for each i using dynamic programming. Combining both the forward and backward passes along with the matrices M_i and the features f_k and g_k it is possible to get the weight updates for the parameters λ and μ .

An important point to be noted is that a single iteration of the Algorithm S and Algorithm T has nearly the same time and space complexity as that of Baum-Welch.

A complication that can occur during the train time is that the feature s considered in the Algorithm S contains the constant S which is sometimes quite large as it is proportional

to the length of the sequence. In such cases keeping track of feature totals for each observation sequence separately is helpful.

Testing At test time we are given the whole sequence. Then the problem reduces to simply finding the label sequence with the largest probability. We can directly use the MLE on $p(y|x)$ and this will give us the best label sequence data and each x_i for $0 \leq i \leq T$ will have a corresponding y_i and each $y_i \in l$ which is either human speech or not.

Segmental Recurrent Neural Networks (sRNN)

Benefits and Shortcomings

Here both Benefits and Shortcomings of the proposed models are given.

Conditional Random Field (CRF)

Benefits CRF comes with very obvious benefits some of which have already been discussed.

- It is able to relax the strong assumption made in the HMM about independence of observation from previous observation. In CRF it takes in the whole sequence and therefore sees the whole context.
- It also circumvents the label-bias problem inherent in many other discriminative framework like MEMMs by considering all the transition possible rather than considering only the transitions from the present state.

Shortcomings CRF while is an excellent tool for speech recognition, there are some shortcomings.

- Firstly the transition matrices can depend as $O(n^2)$ on the number of labels required. In our case it is not a problem since there are only two labels, but in the case with large number of labels this can be an obvious shortcoming.
- The other is related to its inherent model in that it requires to see the whole sequence and therefore it may miserably fail with only partial observability like seeing only the first 3 observation vectors would not allow the model to make any good decision.

Segmental Recurrent Neural Networks (sRNN)

References

- Kong, L.; Dyer, C.; and Smith, N. A. 2015. Segmental recurrent neural networks. *CoRR* abs/1511.06018.
- Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, 282–289. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.