Informedia at PDMC

Wei-Hao Lin WHLIN@CS.CMU.EDU
Alexander Hauptmann ALEX@CS.CMU.EDU
Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

1. Introduction

Our *Digial Human Memory* project (Lin & Hauptmann, 2002) aims to collect and index every aspect of human daily experiences in digital form. By wearing a spy camera, microphones, and a BodyMedia armband, the wearer can collect rich records in a unobtrutive fashion, and many applications can build on top of such multimodal collections. For example, digital human memory can serve as a memory prosthesis to help the wearer recall past events; the habits or anomalies of the wearer can be analyzed from digital human memory. The physiological recordings recorded by a Bodymedia armband provides complementary dimensions of the wearer's experiences, and play an important role in identifying wearer's context and activities.

In this year Physiological Data Modeling Contest, we build a baseline system that models the gender and context tasks as simple binary classification problems using only unambiguous annotations. In addition, we explore two issues. First, instead of ignoring ambiguous annotations and unlabeled data, we attempt to disambiguate them into positive and negative annotations such that the learner can incorporate them in the training phase. Second, we exploit sequence relationship because context activities do not appear randomly but usually in the consecutive minutes as a cluster. A conditional model is built to incorporate the sequence information.

2. Baseline System

In the baseline system, we approach both gender and context tasks as binary classification tasks on a per minute basis. Each minute of instances are assumed to be independently drawn from a identical distribution, and data are represented as a set of feature and label tuples, denoted as $\{(x_i, y_i)\}_{i=1}^{n}$, where x_i and y_i are the feature vector and the label respectively of the i^{th} example in the data set of size n. Since gender should be the same within a session, different gender predictions within a session are resolved by majority votes.

Feature vectors are consisted of nine sensors and two characteristics, and thus the dimension of x_i is eleven. Numeric values of features are scaled. In the gender task, all training data are fully labeled and y_i are set to 1 or 0 correspondingly. In the context tasks, we separate annotations into three set: **positive** (3004 in context 1, 5102 in context 2), **ambiguous** (0, 3003, 5199, 5101 in context 1, 0, 5103, 2901, 2902 in context 2), and **negative** (annotations are in neither Positive nor Ambiguous). Classifiers for both context tasks in the baseline systems are trained only with Positive and Negative annotations.

We choose Support Vector Machine (SVM) (Cristianini & Shawe-Taylor, 2000) as the binary classification learner. The kernel is radial basis function, and model selection is based grid-searching on two parameters (γ from the kernel and C cost parameter) in 10-fold cross-validation.

We evaluate the baseline system on the training set in 10-fold cross-validation, as shown in Table 1. The metric for the gender task is the balanced error rates, and the metric for the context tasks is weighted formula as suggested in the official instruction. We also list the random baseline which guesses every session as gender 0 in the gender task, every minute as 0 (negative) in the context tasks.

	Gender	Context 1	Context 2
SVM Baseline	0.9572	0.7548	0.8711
Random Baseline	0.5	0.7	0.7

Table 1. The performance of out baseline system in 10-fold cross-validation on the training set

3. Annotation Disambiguation

One drawback in the baseline system for the context tasks is that ambiguous annotations are totally ig-

nored. While training data may be easier to discriminate, it runs the great risk of confusing ambiguous labels in the testing phase because the learner never observes the ambiguous data in the training phase. Moreover, the proportion of unlabeled data (annotation 0) is huge ($404872/580264 \approx 0.698$) in the training data, which makes the problem even worse.

We consider the following possible strategies to utilize ambiguous annotations:

1. Ambiguous annotations are either positive or negative with equal probability. Without any prior information, the only reasonable assumption we can make is that it is equally probably for an ambiguous annotation to be positive or negative.

2. Ambiguous annotations are all negative. Although ambiguous annotations may be either positive or negative, we can make a stronger assumption that very few of them contain true positive annotations. By treating all ambiguous annotations as negative hopefully the learner can acquire the "negativity" of the ambiguous labels.

3. Efforts are spent to disambiguate ambiguous annotations. Consider the annotations in question, the context tasks is closer to "semi-supervised" learning where we have both positive, negative, and unlabeled data and multiple-label framework(Jin & Ghahramani, 2003) is applicable. In the multiple-label framework, each data is associated with multiple labels and only one of them is correct label. In order to disambiguate the multiple labels, we can randomly initiate the label distribution $P(y_i|x_i)$ to train a classifier, then use the learned classifier to update the label distribution in an iterative fashion. In context tasks, we fix the label distribution of examples with positive and negative annotations, and only update the distribution of the examples with ambiguous annotations. In order to prevent overfitting, we select the iteration number with the best performance on 10-fold cross-validation.

The above strategies are implemented via sampling. Each training example is assigned a label distribution according to the strategy, and the learner samples positive and examples according to the label distribution. In strategy 3, SVM decision values are transformed into label probability distributions via fitting logistic regression. The performance of the above strategies on the training set is shown in Table 2.

,

	Context 1	Context 2
Strategy 1	0.7625	0.8834
Strategy 2	0.6957	0.8559
Strategy 3	0.7613	0.8707
SVM Baseline	0.7548	0.8711
Random Baseline	0.7	0.7

Table 2. The performance of three annotation disambiguation strategies in 10-fold cross-validation on the training set

4. SVM-Based Markov Models

Besides ignoring ambiguous annotations, the other piece of information the baseline system that is not exploited is the sequential relationship. It is instantly recognizable that positive annotations do not appear randomly within an session. It is more likely to see a positive annotation in the followin minute after seeing a positive annotation and so does negative annotation. Inspired by (McCallum et al., 2000), we create a conditional markov model based on SVM to make predictions on a session (sequence) level. Given a sequence of observations (feature vectors) $x_1, x_2, \ldots, x_m$, the context task can be formulated as finding the annotation sequence of states (annotations) $y_1, y_2, \ldots, y_m$ that maximizes the posterior probabilities, where λ is the model parameters,

$$\arg \max_{y_1, y_2, \ldots, y_m} P(y_1, y_2, \ldots, y_m | x_1, x_2, \ldots, x_m; \lambda) \quad (1)$$

Like other markov models, we make the assumption that current state depends on only the previous state but not earlier states; unlike Hidden Markov Models, we do not model the joint probability of states and observations. Instead, we let the current state depend on not only the previous state but also the current observations. Therefore, Eq. 1 can be rewritten as follows,

$$\arg \max_{y_1, y_2, \ldots, y_m} P(y_1 | x_1; \theta) \prod_{t=2}^{m} P(y_t | x_t, y_{t-1}; \lambda) \quad (2)$$

Conditional on y_{t-1}, we train each conditional probability model $P(y_t|x_t, y_{t-1}; \lambda)$ using SVM, and probability are obtained by fitting logistic regression on SVM decision values.

It turns out that we can have Viterbi-like algorithm to find the most probable annotation sequence given a series of observations. Follow the notation in (Rabiner, 1989), define $\delta_t(i)$ as the highest probability along a single path at time t and the state equals to q_i, where

q_1 is positive, and q_2 is negative:

$$\delta_t(i) = \max_{y_1, y_2, \ldots, y_{t-1}} P(y_1, y_2, \ldots, y_t = q_i | x_1, x_2, \ldots, x_t; \lambda) \quad (3)$$

Eq. 3 can be solved using Dynamic Programming,

$$\delta_t(i) = \max_j \delta_{t-1}(j) \cdot P(y_t = q_i | y_{t-1}, x_t; \lambda) \quad (4)$$

However, the performance of SVM-based Markov Model does not perform well, worse than random baseline. The problem is due to the high unbalanced positive and negative examples after conditioned on s_{t-1}, as illustrated in Table 3. While the number of positive

	$s_{t-1} = \text{neg}$	$s_{t-1} = \text{pos}$
$s_t = \text{neg}$	575776	75
$s_t = \text{pos}$	75	4338

Table 3. Number of examples for the context 1 task in the training set

examples in the training set is already scare and make the learner difficulty to learn, conditioned on s_{t-1} exasperates the situation.

References

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machine: And other kernel-based learning methods*. Cambridge University Press. 1st edition.

Jin, R., & Ghahramani, Z. (2003). Learning with multiple labels. *Advances in Neural Information Processing Systems 15* (pp. 897–904). Cambridge, MA: MIT Press.

Lin, W.-H., & Hauptmann, A. (2002). A wearable digital library of personal conversations. *Proceedings of the Second ACM/IEEE-CSJoint Conference on Digital Libraries (JCDL)* (pp. 277–278). New York, NY, USA: ACM Press.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy markov models for information extraction and segmentation. *Proceedings of the* 17^{th} *International Conference on Machine Learning (ICML)*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE* (pp. 257–286).