2015 数字语音信号处理实验报告

游沛杰 13307130325

July 6, 2015

Contents

1 相关背景 **2**
1.1 实验目的 . . . 2
1.2 项目环境 . . . 2
1.3 预期方法 . . . 2

2 程序原理 **2**
2.1 端点检测 [6] . . . 2
2.2 MFCC[2] . . . 4
2.3 PCA[3] . . . 5
2.4 模式识别 . . . 6

3 具体实现 **6**
3.1 最终流程 . . . 6
3.2 training set 选取 . . . 7

4 结果和分析 **8**
4.1 Kernel Issue . . . 8
4.2 Dimension Issue . . . 8
4.3 Overfitting Issue . . . 9
4.4 PCA Issue . . . 9
4.5 Personal Issue . . . 11
4.6 遇到问题 . . . 12

5 实验总结 **12**
5.1 项目特色 . . . 13
5.2 需要改进的地方 . . . 13
5.3 Thank You! . . . 13

1 相关背景

1.1 实验目的

本次项目我们将实现一个语音识别小程序，主要对 14 个特定说话人的 20 个孤立词进行识别。通过项目实践增加对数字语音信号相关知识的掌握。

1.2 项目环境

以下是我的开发环境。

Table 1: 开发环境

使用语言	Matlab R2012b
操作系统	Win 8.1(64bit)
笔记本内存	2G
其他工具	VOICEBOX
	LIBSVM

1.3 预期方法

在项目开始前我大概预想了一下本次实验准备使用的算法。

因为这次项目主要是让我们熟悉各种特征的提取和机器学习算法，所以我打算 2-3 个不同的算法 (主要是 SVM, VQ-HMM, transfer learning 等)，然后通过 ensemble learning 整合到一起，或者使用 AdaBoost 之类的做法。这样既能保证实现的内容尽量多，同时测试结果也会比较理想。

但是由于条件限制最后并没有完成全部内容。

2 程序原理

简单介绍一下我本次项目用到的重要算法及其相关原理。

2.1 端点检测 [6]

端点检测就是从原始信号中分离出人说话的语音信号段的过程，使用的是二门限的方法。

首先我们对输入语音分帧加窗，对于每一帧信号提取能量和过零率特征：

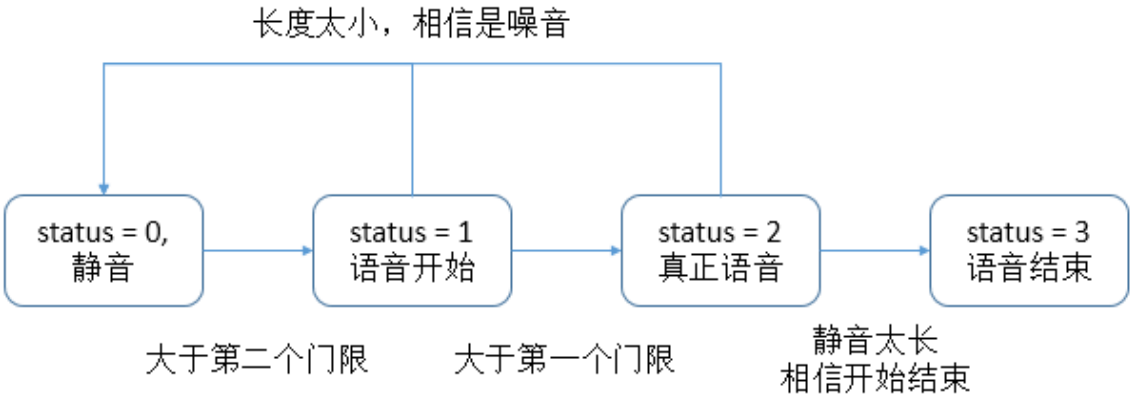
$$\begin{aligned} energy &= \sum x_i^2 \\ zcr &= \frac{1}{2}\sum |\mathrm{sgn}(s_i) - \mathrm{sgn}(s_{i+1})| \end{aligned}$$

根据相关声学原理我们可知，人的发音主要有 3 种:

1. 清音，相当于一个噪音? 过零率比较大，
2. 元音 (浊音)，是一个周期信号，能量比较大，过零率比较小。
3. 爆破音，相当于一个脉冲信号，前部分能量比较大，后部分过零率比较大。

可见如果一帧信号里面能量或者过零率比较大，都有可能是语音信号，我们给这两种特征分别设置两个阈值 (门限)，分别为 $g1, g2$，如果大于 $g1$，我们认为是一定语音信号，如果大于 $g2$ 我们就认为 可能是语音信号。

具体的端点检测程序类似于一个状态机，如下所示。



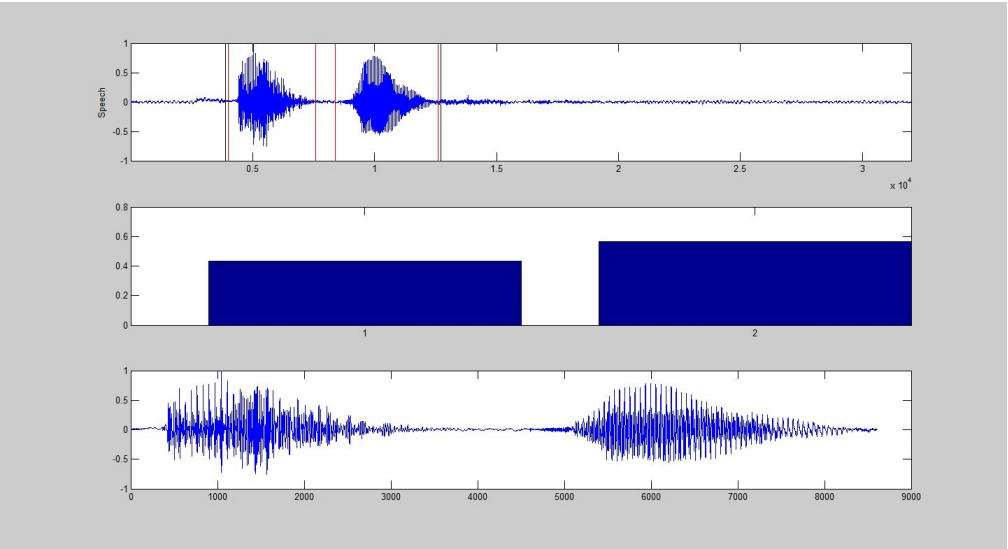
当我们到达状态 3，我们就得到 一段 语音信号。不幸的是，因为同学们录得数据并不符合实际情况，尤其是有的同学停顿过长，使得我们不能很好的划分语音区间，如果门限太大，会漏掉一些音素；如果门限太小，则会包含很多噪音。

我采用的解决方案是提高门限，然后把多段“语音信号”连接起来，流程如下。

- 找出能量最大的一段“语音”区间，这段肯定是语音。(否则数据质量太说不过去了)

- 找与已确认的“语音”信号相邻的区间 (左边和右边相邻)，看谁的能量和过零率比较大，而且离已确认区间得比较近，如果比较明显 (比如能量占全部的 30%) 就确认它是语音信号，否者终止。
- 重复上步骤 2-3 次，因为这里我们识别的都是双音节或者三音节的词，基本不会有其他情况。

最终端点检测效果如下图：



上图是原来数据，原始端点检测发现有 2 个区间符合要求，可能是语音。

中图是他们能量的比例，发现没有一个远大于另外一个的情况，认为是一段声音断开了，开始合并。

下图是合并后结果。

最后我们得到的端点检测结果比较科学，但是有的情况中间静音过长，考虑可以在后续步骤进一步去掉静音段。

2.2 MFCC[2]

MFCC 全称 Mel frequency cepstrum coefficient，梅尔频率倒谱系数，是语音识别中用到的主要声学特征。

我理解的 MFCC 原理如下：

人发声模型可以看成是信号源 (声门波) 和共振腔卷积的结果，需要进行同态解卷来分开，分开之后我们就可以得到信号源的具体信息，从而推断出是发哪个单词的音，而与说话人无关。

同态信号处理应用到我们这个学期学到的抽象代数的知识，考虑两个代数系统 $[X, *]$, $[Y, \cdot]$ 同态:

$$\phi : X \to Y, \phi(x_1 * x_2) = \phi(x_1) \cdot \phi(x_2)$$

或者:

$$\begin{array}{ccc} \mathrm{X} & \xrightarrow{\mathrm{f}} & \mathrm{X'} \\ \downarrow & & \downarrow \\ \mathrm{Y} & \xrightarrow[\mathrm{g}]{} & \mathrm{Y'} \end{array}$$

同态系统易有 $g\phi = \phi f$，所以 $f = \phi^{-1} g \phi$，在我们的例子中 ϕ 就是 z 变换，g 就是对数操作和 Mel 滤波，所以就是

$$f : X = \mathbf{z}^{-1}(m(\ln(\mathbf{z}(x))))$$

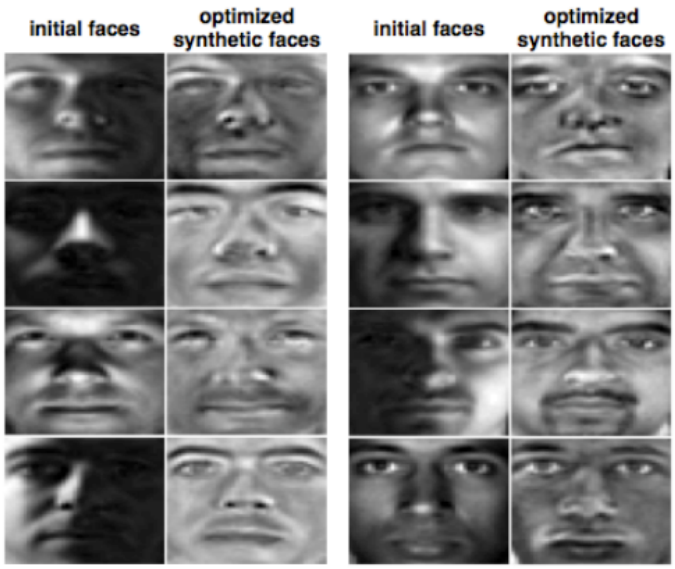
这部分详细过程我们已经在试卷上作答，这里不再重复描述。

2.3 PCA[3]

主成分分析 (Principal Component Analysis) 是机器学习中常见的数据压缩和降维的方法，原理是选出几个数据 variance 最大的方向进行投影。

在这里我之所以会用到 PCA，目的除了降维以外还有去掉与说话人声道，录音设备等无关的特性，是受到前几天参加的一个学术会议的启发。

他们在人脸识别的预处理的时候，为了去掉 光照 这个因素的影响，常常会做 PCA 然后丢掉前 2-3 维的数据，剩下的特征再放进去学习效果更好。



如图有两组数据，每一组左半部分是原始数据，光照影响实现严重，右边是优化过的图像 (详见图中注释)，处理后内容更加清晰可见了 [5]。

同样的在这里我使用了 PCA，得到了 10% accuracy 效果提升，具体做法是 MFCC 系数 PCA 后去掉第二维。

2.4 模式识别

最后我使用的是 SVM，支持向量机。原理是找出最优分割面。

首先这里我们有 20 个类，为了避免 data skew(数据偏斜，正负样本数差异太大)，我采用的是一对一分类，然后每一对中胜出的一类得 1 分统计得分高低，并返回结果。这个过程简单使用 SVM 就可以完成。

还需要考虑维数和核函数的问题，后面再详细展开。

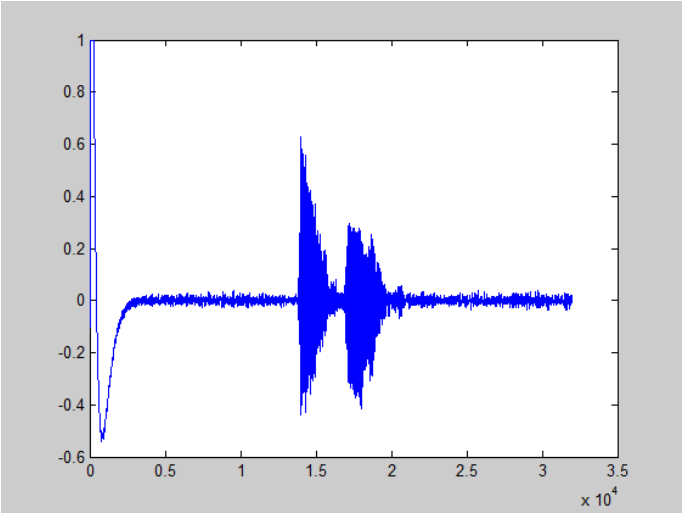
3 具体实现

这里主要讲述实现的语音识别程序流程。

3.1 最终流程

1. 输入测试数据
2. 测试数据端点检测，需要注意的是检测前要先对数据进行 normalize 操作，主要排除设备问题，包括

- 把最前面一段录音用背景音 (取最后一段录音) 代替，因为有的同学开始的时候会有如下波形



  前面一段是不可避免的系统噪音，所以我用语音的最后一小段 (认为是背景音) 来代替了。
- 去掉极个别的最大值，放缩使得最大值为 1

3. 和训练数据一起提取特征 (因为要一起 PCA) 特征由 66 维向量组成，包括如下：
   - 语音段能量
   - 语音段长度
   - 能量长度比例
   - MFCC 中均匀采样成 50*16=800 维向量，PCA 后去掉某几个差异特别大的参数，再降到剩下 63 维 (直接取前几个分量)。
4. 用 training set 训练
5. 用 test set 测试并返回标号结果。

3.2 training set 选取

这里有一位同学 (lzj) 录音效果非常不好，所以我把他所有数据都去掉了，拿剩下的 8 位同学作训练集。

4 结果和分析

4.1 Kernel Issue

在 SVM 中使用不同的核函数，结果也不一样。

本次实验我们主要实验了 3 种核函数，包括线性核，多项式核 (3 次)，RBF(高斯核)。实验结果如下：

Table 2: 同样条件下不同的 kernel function 效果

线性核	**80%**
多项式核 (3 次多项式)	70%
RBF	8%

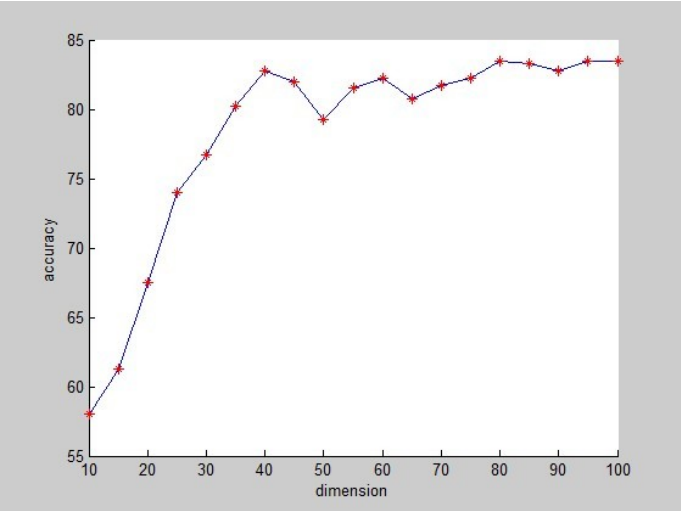
易见效果是线性核 > 多形式核 >RBF，其中 RBF 并不比随机猜测 (random guess) 好，可能是我 LIBSVM 没设置好，但是我换了几个参数效果都不变，只能放弃 RBF。

线性核效果最可观，这从侧面也反映了我们提取的特征比较科学，而且每一维参数相关性比较小?

下面的实验除特殊说明外统一使用线性核,采用封江涛同学的作为测试数据，其他人的作训练数据。

4.2 Dimension Issue

根据常识易知，当特征维数从小变大的时候，程序性能先快速提升后稳定不变甚至下降，我们试验了不同的维数大小来找出最终需要的 dim 值。



多次试验表明在 60 维附近就已经能取得比较好的解了，所以在测试的时候我们令 $dim = 66$

4.3 Overfitting Issue

使用 SVM 进行学习的时候我们发现 training error 为 0%，这让我们不得不考虑 overfitting 的情况。

我们发现，当 training set 中说话人和 test set 中说话人有重叠时，正确率会达到 97% 以上，而当 test set 的说话人是全新的时候，正确率却只有 60-80% 。这说明我们的程序 variance 比较高，可能有过拟合的情况。

所以我们看训练结果的时候也要采取类似的方法，用某几个同学的数据拿去训练，其他同学的数据用作测试。在这里同样不能采用传统的随机混合后分割数据的交叉验证 (cross validation) 做法，应该以每一个说话人作为 test set 来验证程序。

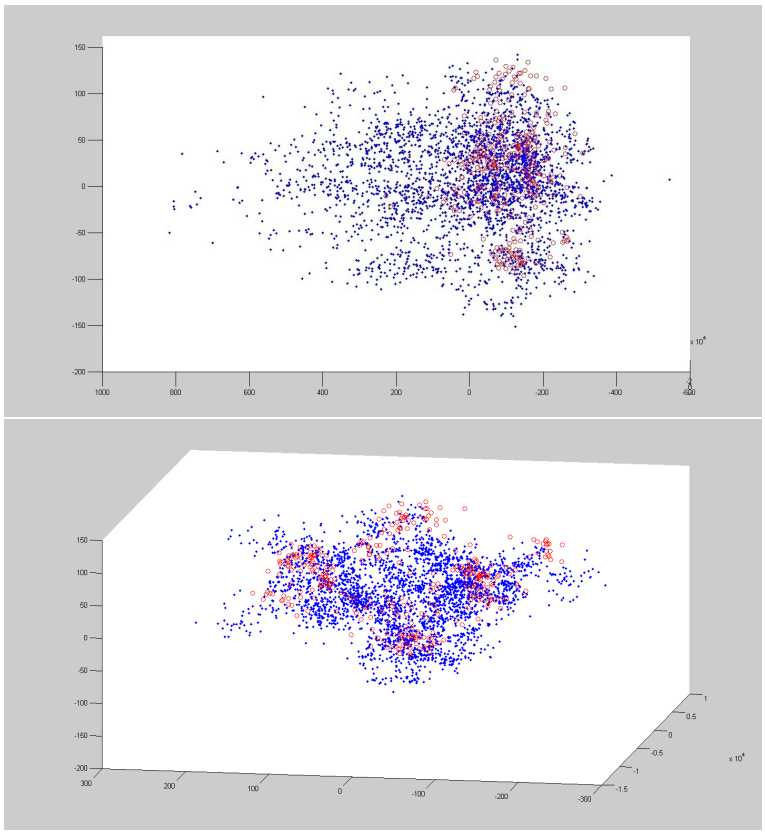
另外我们可以考虑迁移学习 (transfer learning) 的方法 (trAdaBoost, tr-SVM 等)，因为如果有目标域数据的话，即使数据较少，在 SVM 中也可以同样大幅提升性能 (10 个数据即可提升 5%-7%)，所以可以考虑做类似于 one shot 的学习方法。

4.4 PCA Issue

之前看论文的时候也遇到过类似的小技巧，本次实验把 PCA 应用到特征提取中，得到较好效果。

在上面说到，如果 training set 和 test set 能充分混合，那么训练结果就比较理想，基于这个思路我们对数据有如下分析。

例子如下图：



左图中，我们发现在横坐标方向上两者差异比较大 (红色数据主要集中在有半部分，而蓝色数据布满整个坐标轴)，因为说话的内容是相同的，所以认为是人引起的特征不同。右图是去掉这一维特征后，两者充分混合。

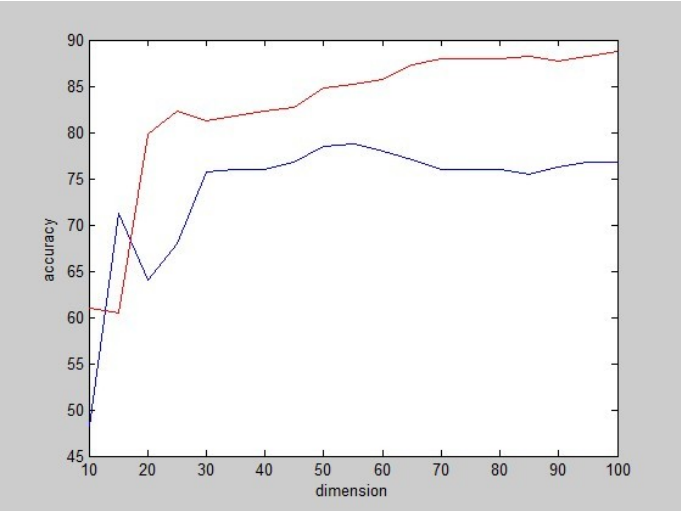
首先我们计算 PCA 结果中源域 (就是 training set) 和目标域 (test set) 的每一个维度中心距离 d，如果两组数据之间的距离比数据内部的标准差大，那么认为是与说话人声音特性有关，而与说话内容无关的特征，舍弃之。

具体判别函数如下:

$$\frac{d^2}{var_{source} * var_{target}} \geq \theta$$

因为是对于某一个维度求上述表达式的值，所以结果都是标量，排序后舍弃最大的 1-2 个。

结果如下，明显看出性能提升：



4.5 Personal Issue

在交叉验证中我们发现同学们的测试结果参差不齐，大多在 75-80% 正确率之间，有的同学甚至达到了 90%。

Table 3: 同学结果

s1(ypj, 我)	s2(wqf)	s3(zzd)	s4(wwy)	s5(zmy)	s6(lq)	s7(zjd)	s8(fjt)
86.75%	73%	72%	78%	**87%**	71%	70%	**87.25%**

因为这里 SVM 是确定性分类器，无局部最优解，所以基本没有误差。

另外我们还拿到了一组北方同学和南方同学的数据 (不是课堂上的学生)，测试结果如下：

Table 4: 南北对比

北方	南方
64.5%	93%

发现南方同学识别正确率比较高，虽然作者我也是南方同学，但是在训练的时候并没有加入针对自己进行优化，只能说是地域口音有差异，或者是录音同学个体引起的差异 (比如实验中的北方同学的 file 读不准)。

4.6 遇到问题

- 数据问题，同学们录音的时候可能为了后面处理方便，竟然一字一顿的说话，中间停顿相比起来十分长，并不符合我们日常生活的说话习惯，甚至比慢速说话还要慢上好几倍。

- 时间问题，本学期下来我总过做过 5 个课程项目，每个做了至少一个星期，加上其他事情最后留下来做语音项目的时间和精力比较少，没有做完我开始预期的所有内容，比较遗憾，特别是 ensemble learning，因为通常加入以后程序会有进一步提升空间。

- *Fitting* 问题，因为训练错误率为 0，所以无论怎样修改方法，对结果影响都不明显，难以实施计划，特别是预想中的 ensemble。SVM 是一个很强的分类器，可能并不适合 AdaBoost，而且 AdaBoost 需要在加权的数据上分类，而使用 LIBSVM 包并不能简单解决这个问题。

- 不会使用 *HMM* 工具包，导致最后尝试 HMM 算法很困难。

5 实验总结

本次项目是我第一次自己写的机器学习程序，学到不少东西。

这次项目相当于一个语音识别比赛，把我平时对 machine learning 竞赛的相关知识都用了进来 (PCA, feature engineering...)，效果也都很明显。最后实现的是相当于一个 handcrafted feature+SVM 的做法，并没有很高的理论价值，但是也有 70-80% 正确率，只是不知道最后测试效果如何。

最让我以外的是参加了 tutorial 以后第二天就能应用到这次的项目中来，并取得很好结果，看来多和学术界的老师同学交流对日后研究或者工作都有很大启发和帮助。

通过这次实验加深了我对语音识别的认识，还需要好好学习。语音领域还是比较有趣，希望以后有机会能再次用到相关知识，特别是在推荐系统中。同时在这里也感谢老师一个学期的付出，我感觉真的学到了很多东西。

再过几天就要验收程序了，而大后天还要计算机原理期末考，并没有开始复习，虽然老师布置的比较早，但是由于种种原因实际写程序的时间只有 3-4 个星期，我表示这次项目，我已经尽自己努力了。这是我选的第一个 B 组专业选修课，感觉很有趣，同时也有很多部分和其他课程重叠，让我增加了对下学期学好其他选修课的信心。

Matlab 这种语言与我们平时用的 C++, Python 有点不一样，用起来不是十分顺手，经常一个函数就要调一天。尽管如此但是我们还是完成了项目的全部内容，也算是一种成就。

相关文献引用，实验结果详细说明有需要可以后再补上。

5.1 项目特色

- 根据我的理解修改了上课讲的二门限做法，同时加入合并区间操作，得到的端点检测效果比较好。
- 除了加窗 (enframe), 频率单位转换 (mel2frq, frq2mel), SVM 以外，全部内容都是我手写的，特别是 MFCC，花费了许多时间。
- 创新的使用了 PCA 算法，提升了程序性能。
- 方法简单，效果显著。使用统一的模型在多个关键词上有可观的效果，避免了人手针对特定词进行优化的过程。

5.2 需要改进的地方

我认为这次项目的成果并不是很理想，总结一下可能需要改进的地方。

主要是可以加入 HMM 和 ensemble。同时可以尝试别的方法，特别是 DNN-HMM, GMM-HMM 等等，会对我了解相关领域知识有很大帮助。

下次应该一开始果断做 HMM 模型，这样就不用在 SVM 上浪费时间了，效果应该会更好，这是项目开始的时候策略不当。

5.3 Thank You!

谢谢阅读！

View this project on GitHub[1]

Report is written in LaTeX

Final result, accuracy =

References

[1] https://github.com/kjkszpj/SpeechR

[2] https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

[3] https://en.wikipedia.org/wiki/Principal_component_analysis

[4] https://en.wikipedia.org/wiki/Support_vector_machine

[5] Kusner M, Tyree S, Weinberger K Q, et al. Stochastic neighbor compression[C]//Proceedings of the 31st International Conference on Machine Learning. 2014: 622-630.

[6] http://blog.csdn.net/ziyuzhao123/article/details/8932336