视听信息系统导论课程设计 - 视频分类

李思涵　　李文硕

2016 年 1 月 3 日

目录

1 团队成员

1.1 李文硕

学号 2013011177。负责视频特征的提取，SVM/随机森林分类器的实现，以及实时视频分类/长视频分割和分段分类的实现。

1.2 李思涵

学号 2013011187，负责音频特征的提取，神经网络分类器的实现，以及在线学习与自动更新的实现。

2 文件清单

```
workspace
    collect_votes.m          统计票数
    do_compare.m             分类器分类脚本
    extract_audio_features.m     提取音频特征
    fun_process.m            原版
    fun_process_longvideo.m 长视频分割
    fun_process_online.m     在线学习
    fun_process_realtime.m   实时学习
    get_filenames.m      获取所有文件名
    mmread.m
    my_mfcc.m            MFCC 实现（含参数）
    nn_train_test.m      训练/测试 NN
    processFrame.m
    processFrame.m
    saveFrame.m
    test_longvideo.m     测试长视频分割
    test_online.m        测试在线学习
    trainingCompare.m    比较分类器
    svm.mat              SVM 分类器
    nn_data.mat          NN 训练器训练数据
    nn_full.mat          NN 分类器，全部样本
    valid_data.mat       格式化后的样本数据
    test_longvideo.m     测试长视频分割
    mfcc/                MFCC 第三方实现
```

3 基础部分

3.1 视频特征

3.1.1 动态特征

视频分类与图片分类的主要区别就在于视频包含大量的动态特征，我们选取镜头编辑特征和运动特征两个方面进行研究。

3.1.1.1 镜头编辑特征

我们注意到，各种视频镜头切换的频率和方式都很不相同。在例如体育类的视频中，存在大量镜头位置的变化，而且变换均在瞬间完成（即下文提到的切变）；而在宗教、科技类视频中，视频镜头切换频率较低，而且变换较为缓慢，所以我们使用以上两种特征作为镜头编辑特征。

使用了镜头变换率和镜头变换方式两个特征。镜头变换率定义为镜头变换率 = 镜头变换次数/视频总时长，镜头变换方式定义为镜头变换方式 = 切变次数/总次数，当总次数为 0 时定义为 1。镜头切换的判断使用两帧灰度值差的绝对值判定。

3.1.1.2 运动特征

在音乐等类视频中，存在大量的明暗变化，而在宗教、科技视频中则变化较小。我们选用亮度和色度的差异来衡量前后两帧的色彩亮度变化。

使用了三个特征。特征一定义为 $|\Omega|/N, \Omega = \{\delta^{\mu}|\delta^{\mu} > 1.2 \times mean(\delta^{\mu})\}$，其中 δ^{μ} 为帧间亮度均值差的绝对值，$|\Omega|$ 为 Ω 的元素个数。特征二定义为 $|\Omega|/N, \Omega = \{\delta^{\mu}|\delta^{\mu} < 0.8 \times mean(\delta^{\mu}) \text{and} \delta^{var} < 0.8 \times mean(\delta^{var})\}$，其中 δ^{μ} 为帧间亮度方差差的绝对值。特征三定义为 $\sum_{f=1}^{N} \delta_f^{channal}/N, \delta_f^{channal} = \frac{1}{xy}\sum_x \sum_y \sqrt{(\delta_f^R - \delta_{f-1}^R)^2 + (\delta_f^G - \delta_{f-1}^G)^2 + (\delta_f^B - \delta_{f-1}^B)^2}$，其中 $\delta_f^{channal}$ 为某一帧某颜色通道的值。

3.1.2 静态特征

我们选取视频中三个关键帧进行静态特征的分析，关键帧的选取方法为第一个镜头，中间镜头和最后一个镜头的中间帧。

3.1.2.1 颜色特征

颜色特征显然是视频分类所用的一个重要特征。我们在 HSV 颜色空间提取特征，使用了 6 个特征，分别是颜色直方图的方差、最大值，亮度的均值、大于均值 1.5 倍的值的个数，饱和度的均值、大于均值 1.5 倍的值的个数。

3.1.2.2 纹理特征

由于纹理是由灰度分布在空间位置上反复出现而形成的，因而在图像空间中相隔某距离的两象素之间会存在一定的灰度关系，即图像中灰度的空间相关特性。灰度共生矩阵就是一种通过研究灰度的空间相关特性来描述纹理的常用方法。我们只取了 $a = 1, b = 0$ 的灰度矩阵，即水平扫描灰度共生矩阵，研究其对比度、相似度、能量、熵、相关性五个特征。

3.2 音频特征

视频与图片分量的另一大不同在与，视频往往会同时带有音频信息。在有些情况下，仅仅通过视频信息是很难对一些视频进行分类的。例如，一个新闻节目和 MV（Music Video）可能都包含大量的外景镜头，而二者的音频特征缺差异明显。为此我们从时域特性和频域特性两个方面提取特征。[1]

需要注意到的是，有些视频文件的音频包括左声道和右声道两个部分。在处理这些音频序列的时候，我们将左声道和右声道单独处理，然后对二者取平均，从而避免只取一个声道而带来的损失。

3.2.1 时域特征

3.2.1.1 能量

能量是音频的一个主要时域特征。从音频的能量中我们可以看出音频序列的基本特征。例如，音乐类节目的音频序列在能量极本上一直很强。而相比之下，其它类别的节目的音频序列能量通常具有间歇性，在人物说话的部分才会出现比较集中的能量。这样一来，我们可以通过音频的能量来估计视频的基本形式（访谈为主/音乐为主等）。

要注意的是，由于不同视频的长度不一样，我们需要将能量归一化。即 $E_{\text{norm}} = \frac{1}{N}\sum_{i=1}^{N} x^2[i]$

3.2.1.2 过零率

过零率指的是音频信号经过零点跳变的频率。很容易注意到，在一定范围内，频率越高的信号过零率越高。所以，过零率间接描述了信号的频率。但是由于它是从时域的角度去分析的，不需要 FFT 等复杂的计算，所以也可以作为一种比较经济的方法来提取音频的特征。

3.2.2 频域特征

3.2.2.1 频谱

从频域分析音频序列的最直接方式便是看频谱。为了能展现整个时间序列的频谱，同时考虑到一些声音特征的短时性，我们使用短时傅立叶对原音频进行分析。具体方法为，我们先把音频序列分成长为 20ms 的段，同时为了减少信息的损失，段于段之间留下 50% 的交叠。然后，我们将每一段加窗，并重采样到 5 个点，计算 5 点 FFT。最后我们用最大值和方差两个统计量来描述每个 FFT 系数。

3.2.2.2 MFCC

MFCC 是 Mel-frequency cepstrum 的简称。其使用的是线性的倒频谱表示方法，与人类的非线性听觉系统更为接近，故经常被用在音频分析处理中。MFCC 得到的是一连串系数，我们选用 Hamming 窗作为窗函数，并在每段时间中取 13 个 MFCC 系数，得到一个系数矩阵。

为了描述 MFCC 系数矩阵，我们对每个系数选用四个统计量：方差，最大值最大值与平均值的比值，以及中位数。这样，我们便可以得到 51 个特征。

3.3 分类方法的设计

3.3.1 SVM 分类器

SVM 分类器的工作原理很简单，即将特征向量看作高维空间的点，分类即生成一个核函数的超平面对两类对应的点进行分割，方便易用。

我们使用了 matlab 自带的 SVM 分类器，使用多项式核。需要考虑的一点就是 SVM 是二分式的分类器，所以对于 5 个类别，我们需要设计一个分类策略，来使其正确工作。

3.3.1.1 直接实现法

我们可以使用最简单的直接实现，即使用五个分类器，每个训练时只将其对应的一个类别的 label 置为 1，其他均置为-1。分类时，若只有一个分类器输出 1，就将该类作为输出，否则认为无法分辨，输出 0。但实际实验中发现此方法效果极差，几乎无法得到有效的结果，所以我们抛弃了这种方法。

3.3.1.2 树状二分法

通过查询资料 [2] 我们发现可以用一种树形结构来实现，即对于 5 个类别的视频，实现 $5\times(5-1)/2=10$ 个分类器，每一个都可以对两个类别进行分类。分类时，首先使用 1vs5 的分类器，若结果是 1，则使用 1vs4 的分类器，即后一个数减一；若结果是 5，则使用 2vs5 的分类器，即前一个数加一。如此进行 4 层分类，最后会得到一个确定的结果。但在实际实验中发现，这一方法依然缺点很多，例如收敛性不好，召回率低（无 0 输出）等等。

3.3.1.3 投票法

针对以上问题，我们自己设计了一种分类策略，即实现 $n\times(n-1)/2$ 个分类器，每个分类器对应两个类别输出为 1，其他类别为-1，分类时统计每个类收到 1 的个数，取最多的作为结果，如果有最多的不止一个，认为无法分辨。在实际实验中，该方法取得了较好的结果。

3.3.2 随机森林分类器

随机森林方法即生成多颗决策树，对每颗决策树使用训练集中的一部分进行训练，分类时结果由决策树投票决定。

我们使用的是 opencv 中的 RTrees 类，尝试了上文的投票法和直接由一个随机森林给出结果两种方式。

3.3.3 神经网络分类器

我们使用了 MATLAB Neural Network Toolbox 中的 Pattern recognition network 来完成我们的识别。由于我们一共提取了 102 个特征，而一共有 5 个分类，我们建立一个输入 102 维，输出为 5 维的，有 100 个隐含层的神经网络，其具体结构如下图所示：

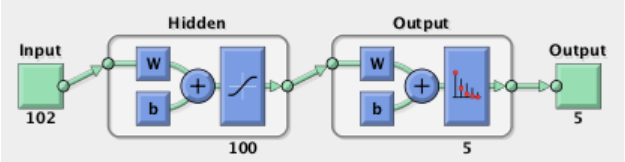

图 1: 神经网络结构

由于神经网络的输出为一个 5 维向量，每个元素的值都是 0-1 之间的 double。为了达到分类的目的，我们建立如下判据：弱神经网络输出最大值与次大值相差小于 0.2，则认为分类成功，分类结果即为最大值对应的分类，否则分类失败。

4 实验结果

在我们的实验中，我们每次随机选取每种类别各 20 个样本作为测试数据，其它视频作为训练数据，进行交叉验证。结果发现，我们的 SVM 分类器和 NN 分类器都能达到不错的分类效果，两分类器最终的 fscore 都在 0.7 上下波动。分类的结果如下：

分类器	SVM	NN	NN-视频	NN-音频
准确率	0.63	0.76	0.56	0.70
召回率	0.75	0.76	0.57	0.70
Fscore	0.68	0.76	0.56	0.70

可以看到，同时使用视频和音频信息确实提高了正确率。各个类别的分类情况如下：

	1014	1019	1016	1017	1020	0
1014	0.65	0.05	0.10	0	0.20	0
1019	0.15	0.60	0.05	0	0.20	0
1016	0	0	0.95	0	0.05	0
1017	0.15	0.20	0.20	0.30	0.15	0
1020	0	0	0.20	0.05	0.75	0

可以看到，1016（政治）和 1020（技术）的识别正确率最高，而 1017 类的正确率很低。同时，1020（技术）的视频较容易被分类成 1016（政治），1019（体育）的视频较容易被分类为 1020（技术）。这可能是由于网络深度较深。

5 提高部分

5.1 在线学习与自动更新

由于我们使用的 Neural Network Toolbox 中的神经网络支持分段学习，我们使用它来实现在线学习和自动更新。也就是，我们通过在识别的过程中更新我们的神经网络，在识别的过程中训练，从而在线提升模型的识别能力。

我们使用的方式是，每次得到一个视频的特征后，从文件中读出当前的神经网络，以及从上一个视频中提取的特征。我们首先使用上一个视频的特征和正确分类来训练我们的网络。然后我们使用更新过的网络来进行识别，得到当前视频的分类结果。最后，我们将更新过的神经网络和本次提取出的特征保存在文件中，以便下次分类使用。具体来说，我们首先预先训练出了一个神经网络，然后使用 `adapt` 函数使其适应新样本，从而达到在线学习和自动更新的效果。

实验中，我们首先每个类别取一个样本，构建初始网络。然后我们将剩余样本随机顺序输入神经网络，并统计正确率。结果如图所示。可以看到，模型的正确率随着识别的进行而逐渐上升。这说明我们成功实现了在线学习和自动更新。

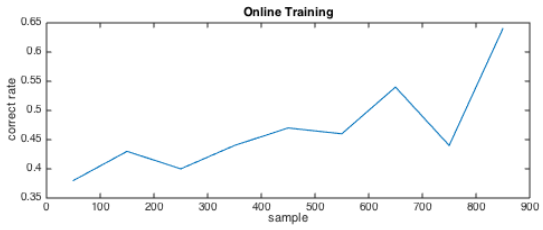

图 2: 在线学习

5.2 实时视频分类

对于视频的实时分类，我们采用流式读取，每次读入视频的一帧（视频每 5 帧取 1 帧进行播放和计算，间隔即为 5/rate 秒），播放图片并计算。分类的策略是实时的计算画面灰度值的变化情况，视频播放到 1000 帧之后（1000 帧之前信息量不足以进行判断），以灰度值变化情况来判断是否有剧烈的镜头变换。当有剧烈的镜头变换时，更新所有特征（大约耗时 200ms，对于主流的 25 帧/秒的视频能做到实时）并进行分类，label 以 title 的形式显示在图片上，并可能根据视频的播放和特征的更新发生变化。但就稳定值来看比较令人满意，对于某些视频，可以在 1000 帧时就稳定得到正确的分类结果。

图 3: 实时视频分类

5.3 长视频分割和分段分类

对于长视频，我们像普通视频一样进行特征提取。但使用的分类器不含有镜头编辑特征和开头结尾关键帧的静态特征。完成提取后对于每一个镜头进行分类，如果相邻镜头分类结果不同，我们就认为此处应该分割。

参考文献

[1] zhouzhouzf. 暴力视音频分类检测相关论文.

[2] 覃丹. 基于多特征组合和 svm 的视频内容自动分类算法研究. Master's thesis, 上海交通大学, 2009.