数据分析的数学导论期末大作业
基于Boosting与Bootstrap方法的Vincent van Gogh画作鉴定分析

胡安然 1200010644　钱鹏宇 1100010699　张维熹 1100010631

2015 年 1 月 21 日

1　引言

在这篇报告中，我们将尝试用统计方法对64幅Vincent van Gogh的真品和15幅赝品进行鉴定。

在文献[1]中，作者首先从包含极为复杂信息的图像矩阵中提取出108个特征，再利用Boosting方法从中选取出其中的4个特征，并将这4个特征的数值作为各个画作的空间坐标，通过把偏离平均位置的程度超过某个阈值的画作判断为赝品，实现了准确率高达83.54% 的分类。这里，阈值的选取是通过交叉验证实现的。

为了提高分类的精度，我们将从两个方向进行改进，并比较相应的结果。其中一个方向是“精化”，即利用Boosting方法进一步从4个特征中选取出更重要的1至3个特征，再分别利用[1]中的分类方法和SVM进行分类。另一个方向则是“粗化”，即将Bootstrap方法应用到Generalized Linear Model中进行重采样，再对回归结果进行分类。

此外，我们还将分别给出“精化”和“粗化”的后验参数选取结果。我们将看到，通过后验方法选取的参数达到的效果是非常好的。当然，因为后验参数选取是“事后诸葛亮”，所以实际应用中并不能直接使用。

最后，需要说明的是，为了使结果具有可比较性，我们的出发点就是4个Geometric Tight Frame特征（而非画作本身）。

2　直观分析

在详细叙述我们的两种方法之前，我们先对4个Geometric Tight Frame特征进行一个简要的分析。

首先，我们对4个特征进行绘图：

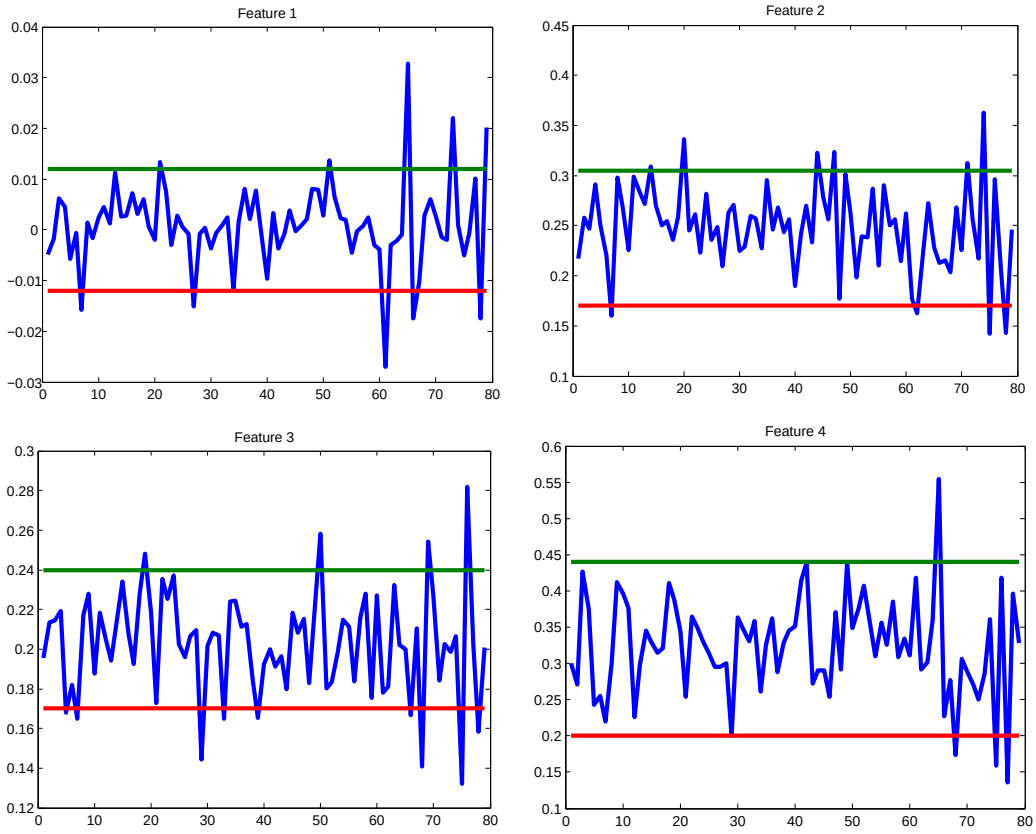


Figure 1: 4 Geometric Tight Frame Features

这里我们对每个特征人为选取了2个阈值。我们看到，仅仅根据这些阈值，将两条线以内的判定为真品，将两条线以外的判定为赝品，我们得到的准确率即如下表：

Table 1: 基于4个特征的各自的Hard Threshold分类结果

Feature	1	2	3	4
TPR	93.75%	90.63%	89.06%	**100%**
TNR	**40%**	26.67%	40%	26.67%
Acc	83.54%	78.48%	79.75%	**86.08%**

当然，这里阈值的选取是有一定后验因素的，即我们是在已知有且仅有后15个是赝品的基础上来划定的阈值。但即便如此，我们也可以看到，即使是非常naive的方法也能得到上述不错的结果。换言之，我们所提出的方法，理应至少在允许后验地选取参数的前提下达到上述效果。

此外，还可以看到，无论根据哪个特征进行分类，TNR永远是表现最差的指标。事实上这正说明了验伪的困难性。而事实上，由于4个特征本身在TNR上的局限，也造成了无论是[1]中还是我们的方法，都很难在TNR上（较大地）超过50%的准确率。

在接下来的两节中，我们就将详细地叙述我们提出的两类改进方法。

3 基于Boosting的特征选择与分类

在这一节中，我们介绍基于Boosting的van Gogh画作鉴定算法。

我们的基本出发点是在于4个Geometric Tight Frame Feature都各自具有较好的分类效果，但4个特征之间的相互关系可能较为复杂，难以用线性的或近似线性的分类器分离。所以我们提取其中的更主要的特征，获取其直观，并据此选择更合理的分类方法。

我们首先利用[1]中的特征选取方法选取2或3个特征，并分别利用(citation)中的方法和支持向量机(SVM) 对数据进行分类。

3.1 基于AUC的特征选择

记训练集的数据矩阵为：

$$X = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_{78}} \end{bmatrix}$$

其中$\mathbf{x_i} \in \mathbb{R}^4, i = 1, \ldots, 78$，为每一幅作品的特征。我们首先对$X$做归一化处理：

$$\bar{X} = X\Sigma^{-1}, \Sigma = diag(\sigma_1, \ldots, \sigma_4),$$

其中$\sigma_i, i = 1, \ldots, 4$为$X$每一列的标准差。我们令集合$\{1, \ldots, 78\} = \mathcal{T}_{vG} \bigcup \mathcal{T}_{nvG}$，其中$\mathcal{T}_{vG}$为所有van Gogh作品的编号集合，$\mathcal{T}_{nvG}$为所有非van Gogh作品的编号集合。

下面我们考虑从4个特征中提取2或3个特征并对数据进行可视化和分类。设提取出来的特征为$\mathcal{F} = \{i_1, \ldots, i_{|\mathcal{F}|}\}$，定义$\bar{X}_{j,\mathcal{F}} = (\bar{X}_{j,i_1}, \ldots, \bar{X}_{j,i_{|\mathcal{F}|}})$，即数据在选定特征子集方向上的投影。我们对每一个$\mathcal{F}$定义**vG中心**:

$$c^{\mathcal{F}} = \frac{1}{|\mathcal{T}_{vG}|} \Sigma_{j \in \mathcal{T}_{vG}} \bar{X}_{j,\mathcal{F}},$$

并以此定义每一个数据点到vG中心的距离：

$$d_j^{\mathcal{F}} = ||\bar{X}_{j,\mathcal{F}} - c^{\mathcal{F}}||_p, \ 1 \leqslant j \leqslant 78.$$

直观地看，如果$\mathcal{F}$是一个好的特征集合，那么对于$\mathcal{T}_{vG}$中的点，$d_j^{\mathcal{F}}$应该较小；对于$\mathcal{T}_{nvG}$中的点，$d_j^{\mathcal{F}}$应该较大。因而我们将分类规则定为：选择一个距离阈值d_ϵ，对于一个未分类样本X_{test}，

$$X_{text} \in \begin{cases} \mathcal{T}_{vG}, \ d_{test}^{\mathcal{F}} \leqslant d_\epsilon \\ \mathcal{T}_{nvG}, \ d_{test}^{\mathcal{F}} > d_\epsilon \end{cases}. \tag{1}$$

对于每一个固定的$\mathcal{F}$，我们将$\{d_j^{\mathcal{F}}\}_{j=1}^{78}$从小到大排序:$d_{j_1}^{\mathcal{F}}, \ldots, d_{j_{78}}^{\mathcal{F}}$，则可以选择的距离阈值为：

$$d_{j_1}^{\mathcal{F}}, \frac{1}{2}(d_{j_1}^{\mathcal{F}} + d_{j_2}^{\mathcal{F}}), \ldots, \frac{1}{2}(d_{j_{77}}^{\mathcal{F}} + d_{j_{78}}^{\mathcal{F}}), d_{j_{78}}^{\mathcal{F}}. \tag{2}$$

利用这些不同的距离阈值，我们可以画出对应于$\mathcal{F}$的ROC曲线（如图2所示），其中横轴为False Positive Rate (FPR):

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}},$$

纵轴为True Positive Rate (TPR):

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}},$$

其中TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative。我们考虑使得AUC（Area Under ROC Curve）最大的$\mathcal{F}$。当$|\mathcal{F}| = 3$时，AUC如表2所示；当$|\mathcal{F}| = 2$时，AUC如表3所示。因而我们考虑$\mathcal{F} = \{1, 2, 3\}$和$\mathcal{F} = \{1, 3\}$的情形。

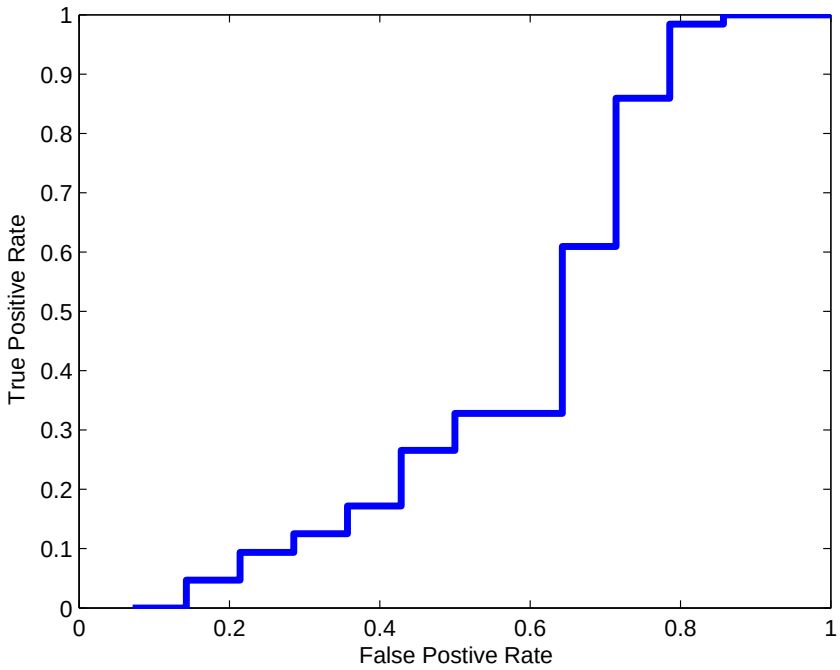


Figure 2: 当选取全部4种特征时的ROC曲线

Table 2: 选取3种特征时的AUC

特征集合	{1,2,3}	{1,2,4}	{1,3,4}	{2,3,4}
AUC	**0.5960**	0.4554	0.4632	0.3996

Table 3: 选取2种特征时的AUC

特征集合	{1,2}	{1,3}	{1,4}	{2,3}	{2,4}	{3,4}
AUC	0.5424	**0.5815**	0.4386	0.4152	0.3438	0.3672

最后，我们强调，上面定义d_j^r时范数p是需要根据点集的直观来确定的。[1]中选取的$p=1$，但我们在数值实验中发现，经过进一步降维（例如选取特征1、3或3、4作为坐标绘图），我们可以得到如下直观：

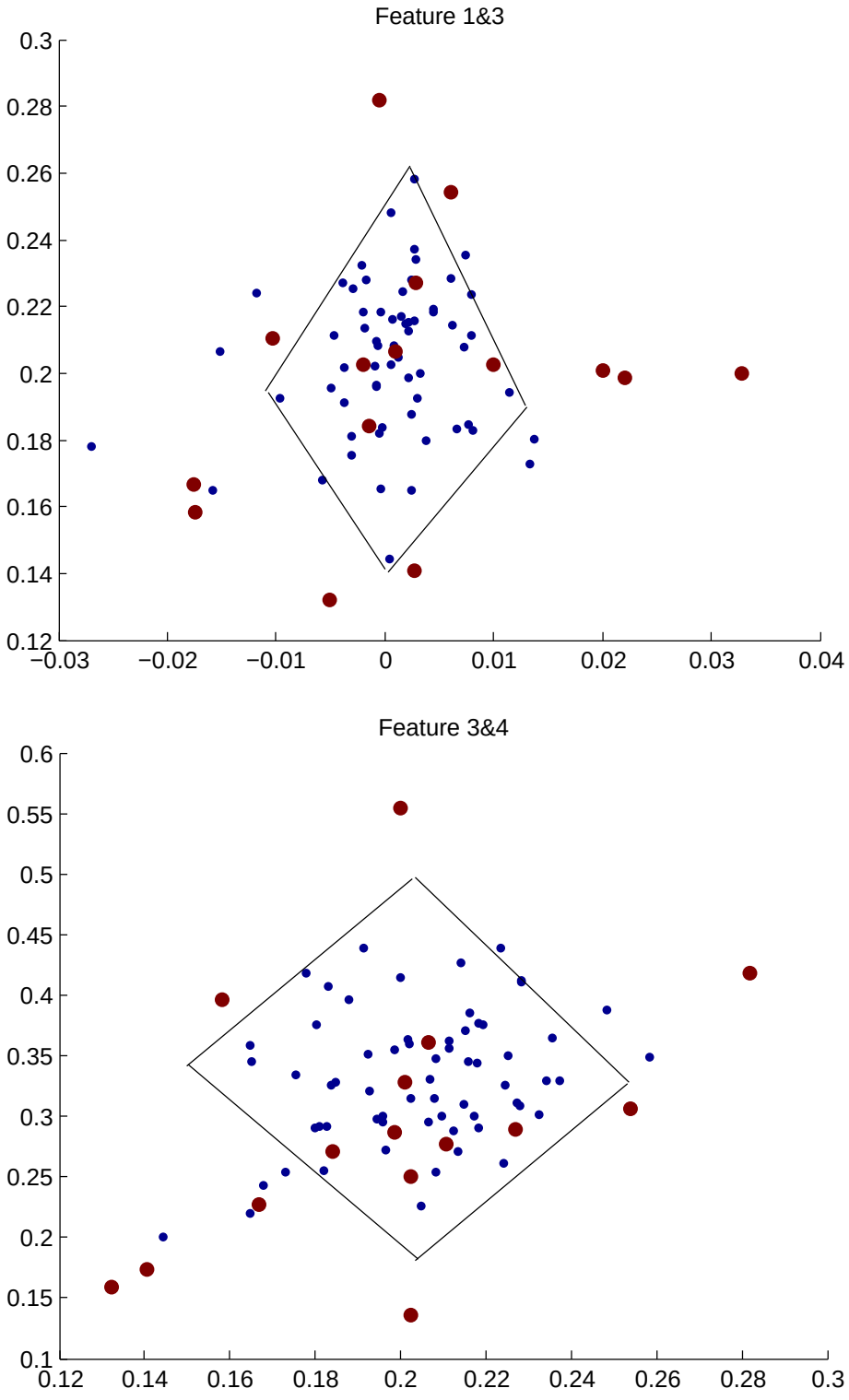


Figure 3: p的选取

由此，我们不难看到，选取$p = 1$得到的菱形邻域能更好地分离真品（蓝色）与赝品（褐色）。

3.2 基于距离阈值的分类

3.2.1 先验分类结果

我们需要在(2)中的距离中选取一个作为距离阈值，并按规则(1)进行分类。对于每一个距离阈值d_ϵ，我们可以计算出其准确度：

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{78},$$

并选择使得准确度最大的d_ϵ作为距离阈值。当$\mathcal{F} = \{1, 2, 3\}$时，leave-one-out交叉验证的分类结果如图4所示，其中红色的点为TP，粉色的点为TN，绿色的点为FP，蓝色的点为FN。当$\mathcal{F} = \{1, 3\}$时，leave-one-out交叉验证的分类结果如图5所示。

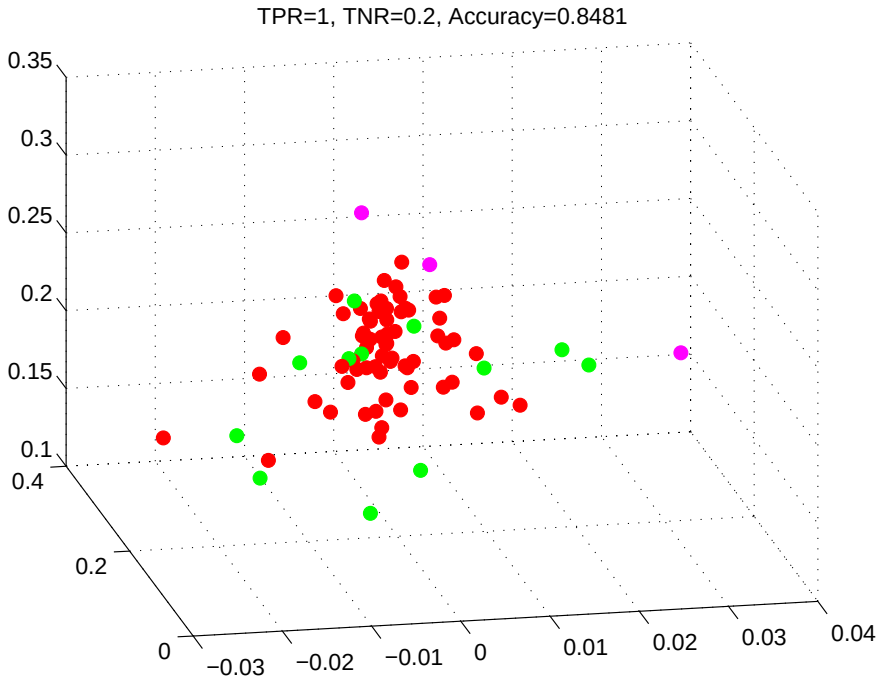


Figure 4: 当$\mathcal{F} = \{1, 2, 3\}$时的基于距离阈值的分类结果

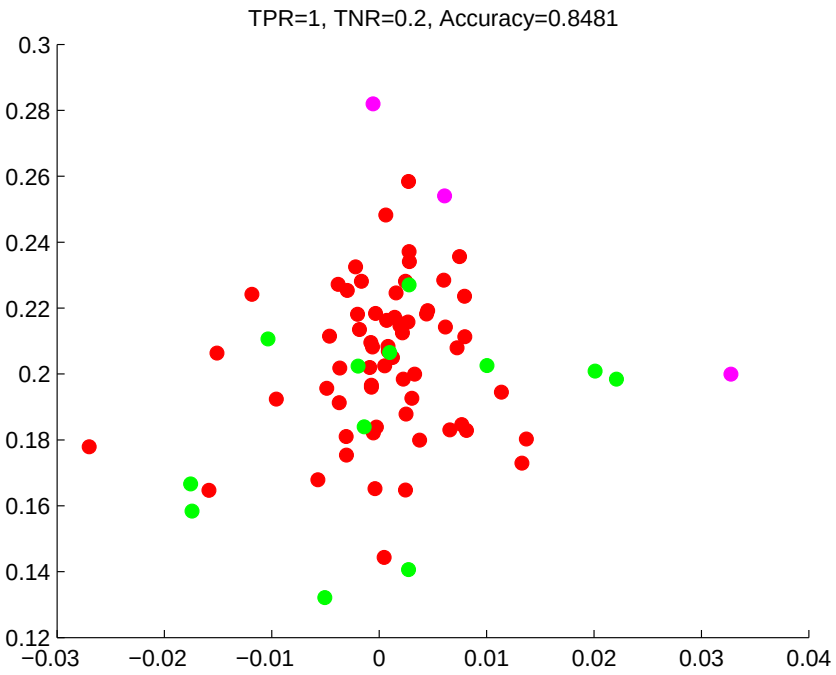


Figure 5: 当$\mathcal{F} = \{1, 3\}$时的基于距离阈值的分类结果

通过简单的观察我们可以发现，虽然上述两种分类策略都能达到84.1%的准确率，但FPR均只有20%，即15幅赝品中只成功识别出3幅。这一现象出现的原因在于，在79幅画作中，只有15幅是赝品，即使将d_ϵ设成∞，那么正确率也能够达到81.0%。

3.2.2 后验参数选取

为了避免将d_ϵ取得过大，我们尝试改变分类规则的评价标准：

$$Acc_{beta} = \text{TPR} - \lambda\text{FPR},$$

当λ较大时，我们选取的距离阈值可能能够得到较大的TNR。通过选取不同的λ，我们绘制出trade-off曲线，当选取3个特征时，如图6所示；当选取2个特征时，如图7所示；当选取1个特征时，如图8所示。从上述图中可以看出，提高TNR会极大地影响TPR，然而在某些情况下适当地提高λ可以同时提高准确率和TNR。

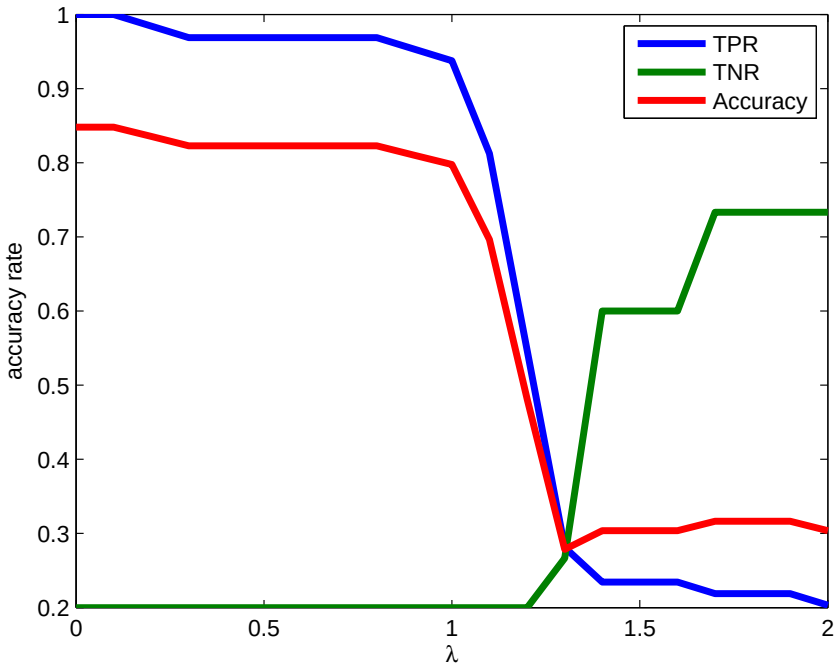


Figure 6: 选取不同λ时的TPR,TNR和准确率，$|\mathcal{F}| = 3$

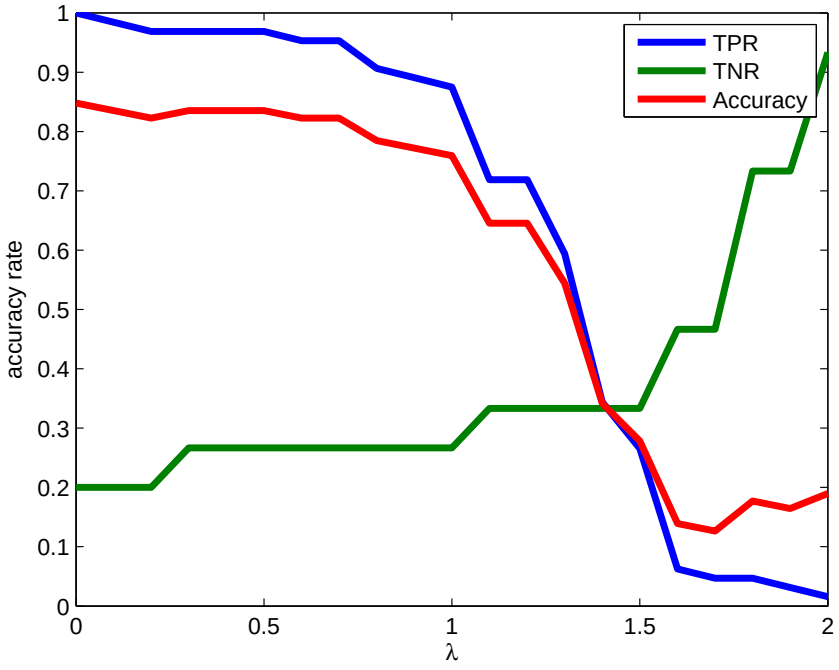


Figure 7: 选取不同λ时的TPR,TNR和准确率，$|\mathcal{F}| = 2$

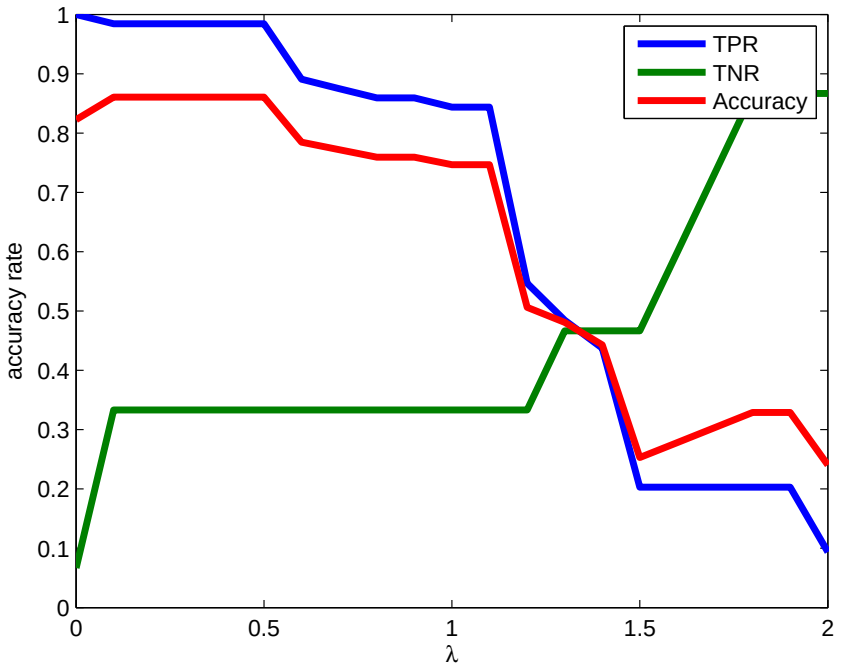


Figure 8: 选取不同λ时的TPR,TNR和准确率，$|\mathcal{F}| = 1$

我们将看到，选取$\lambda = 0.2$时效果会非常不错（见表4）。

但需要注意的是，λ的取值只能后验地确定，这是不够理想的。一个可能的改进是在每次Leave-One-Out实验中，对78个样本组成的训练集本身确定λ，再用于test。这种思想在“粗化”方法中也有所体现。

3.3 基于SVM的分类

除了基于距离阈值的分类方法之外，我们还尝试了利用SVM对原数据和$\mathcal{F} = \{1, 2, 3\}$的情形进行分类。

注意到SVM是线性分类器，但本问题的数据直观上并非线性的，所以一般而言使用SVM是不恰当的。但如果以超过50%的TNR为目标，那么把这一问题近似为线性问题也未尝不可。

事实上，例如选取特征1和2，则我们可以作出如下的分离超平面（见下页首）：

从图中不难看出，此时TNR=46.67%，TPR=92.19%，而Acc=83.54%。这就说明SVM的确是可以考虑的。当然实际运行SVM时，并不一定就能选取到这样较为合适的超平面。

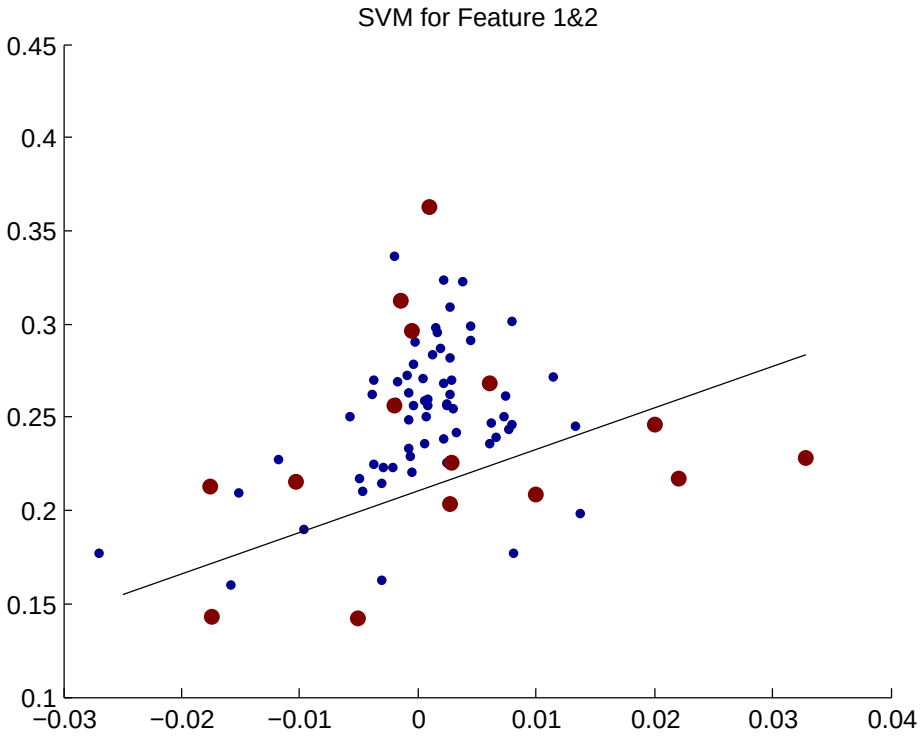


Figure 9: Feature 1和2的一种可能的线性分类结果

下面我们就简要叙述这一方法的数值结果。

当$|\mathcal{F}| = 3$ 时，分类结果如图10 所示。

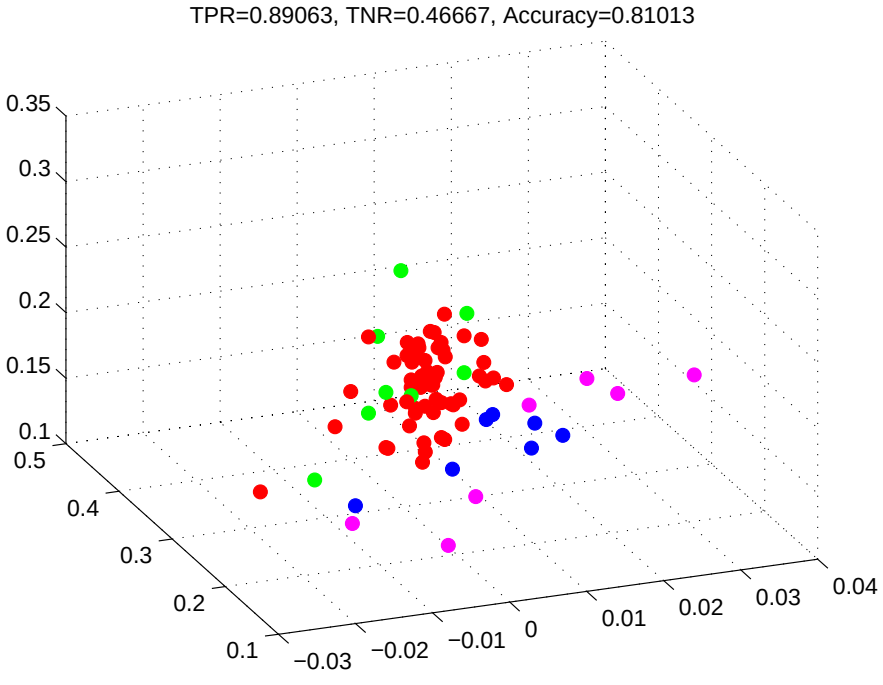


Figure 10: 当$\mathcal{F} = \{1, 2, 3\}$时的基于SVM的分类结果

3.4 小结

我们将上述所有方法的分类准确度总结在表4中。从表中可以看出，Boosting+基于距离($\lambda = 0.2$)的表现相对较好，然而由于λ的选取是后验的，故这一个分类器在一定程度上违反了leave-one-out的交叉验证规则。其他分类器的最好表现为84.81%，超过了84%。

Table 4: 不同分类方法的准确度比较

采用方法	特征数	TPR	TNR	Acc
Boosting+基于距离	4	**100.00%**	6.67%	82.28%
Boosting+基于距离	3	**100.00%**	20.00%	84.81%
Boosting+基于距离	2	**100.00%**	20.00%	84.81%
Boosting+基于距离	1	**100.00%**	6.67%	82.28%
Boosting+基于距离($\lambda = 0.2$)	1	98.44%	44.44%	**86.08%**
Boosting+SVM	4	90.63%	33.33%	79.75%
Boosting+SVM	3	89.06%	**46.67%**	81.01%

4 基于Bootstrap方法的Vincent van Gogh画作鉴定分析

在这一节里，我们将引入基于Bootstrap方法的van Gogh画作鉴定算法。

与上一节中不同，我们不再对特征进行进一步精选以求获得几何直观并据此指导分类器的选择，而是采取一个相反的思路。这主要体现在两个方面：一是增加样本，二是增加随机性。而Bootstrap很好地实现了这两个目标。

4.1 BootLog算法

我们的算法是通过将Bootstrap算法用于loglog Generalized Linear Model得到的，所以以下将其简称为BootLog算法。

我们的基本考虑是所谓“真品”和“赝品”其实都只是人们的一种判断——即使在这里前64幅画是公认的真品，而后15幅画是公认的赝品，也只是判断其为真或假的人多到了一定程度，从而成为了一种权威。

所以我们构造如下的Generalized Linear Model：

- 设Y服从均值为p的Bernoulli两点分布，其中Y取1表示人们认为是真的，取0则表示人们认为是假的；
- 设$\log(-\log p) = X\beta$，即$p = e^{-e^{X\beta}}$；

- 用最大似然估计确定参数β。

我们给出如下评注：

- 上述模型中，对79幅画，Y就是其（公认的）真伪，而X就是4个Geometric Tight Frame Feature；
- 之所以取loglog模型，是因为:
  - 其重尾性较弱，从而可以使得较大的X影响更大，而较大的X对应的是赝品，这正是我们需要重点区分的；
  - 且因为X的量级都很小，所以不会造成β过小的情况；
  - 经过和logit、probit、log、comploglog以及reciprocal等多种模型比较，确定了loglog是表现最好的模型；
- 我们可以直接利用matlab的glmfit函数估计参数β。

接下来，为了更充分地体现出我们已知的关于画作真伪的信息的权威性以及一定的偶然误差，我们认为有必要“增加”样本量。于是我们引入Bootstrap方法，这样一来，从某种意义上来说就在增加总样本量的同时也增加了容易被占据绝大多数的真品掩盖的赝品的数量。

这里需要注意的是，在进行Bootstrapping时，新得到的Y很可能不再是0-1变量。于是我们在Bootstrap过程中采取如下的正态模型：

- Y来自均值为p的正态分布；
- $p = X\beta$。

这样做的合理性可以从**中心极限定理或不变原理**来理解，即均值为p的Bernoulli随机变量在积累足够多之后，其平均行为趋向于均值为p的正态随机变量。所以随着Bootstrap重采样的进行，正态假设就变得越来越合理了。我们将利用matlab的bootstrp函数进行Bootstrapping。

最后，在Leave-One-Out测试中，第i个子测试中可以选取78幅画通过按照上述方法计算出相应的β_i，并将剩下的那幅画的Geometric Tight Frame Feature代入模型，来计算出p_i。这样我们一共可以得到79个不同的p_i。

为了完成分类，我们采用和[1]中类似的方法。具体而言，我们有如下两种版本：

- 后验版：
  - 计算出p_i的均值$\bar{p}$；
  - 计算出p_i到$\bar{p}$的最大距离δ；
  - 凡是$|p_i - \bar{p}| \leqslant \dfrac{\delta}{2}$的都认为是真品，反之认为是赝品。

- 先验版：
  - 利用第i个子测试中的训练集中的一部分画（77幅）进行Leave-One-Out测试，从各个p_j到其均值的距离中选择出最大化Acc的δ_i（这里分类标准同后验版最后一步）；
  - 若$|p_i - \bar{p}| \leqslant \delta_i$，则认为是真品，否则认为是赝品。

我们将上述算法总结如下：

Algorithm 1 BootLog with Leave-One-Out Test Y: 真/假 X: 4个Geometric Tight Frame特征

Require:

Geometric Tight Frame Features X, True/False Response Y

Ensure:

For $i = 1 : 79$

Kick out the i^{th} picture, fit loglog Generalized Linear Model using glmfit:

$m = 1$;

$\beta_m = \text{glmfit}(X, Y, \text{'binomial','link','loglog'})$;

$P_{fit} = e^{-e^{X\beta_m}}$;

$R = Y - P_{fit}$;

Bootstrapping:

Sampling with replacement the components of R^0(residuals):

$\hat{R} = resampling(R)$;

$\hat{Y} = P_{fit} + \hat{R}$;

Fit loglog Generalized Linear Model with Y replaced by $\hat{Y}$:

$m = m + 1$, Repeat all the above steps k times(e.g. k=30);

Compute $\beta = \frac{1}{k}\sum_{m=1}^{k}\beta_m$ and $p_i = e^{-e^{X\beta}}$;

Determine the Threshold parameter using cross-validation:

Choose 77 pictures from the 78-pictures' training set;

For $j = 1 : 77$

Compute $p_j^{(i)}(j = 1, 2, \ldots, 77)$ using the above procedures;

Let $\bar{p}_i = \frac{1}{77}\sum_{j=1}^{77} p_j^{(i)}$, δ_0 be the former threshold parameter;

$$\delta_i = \begin{cases} \text{argmax Acc}(\delta = |p_j^{(i)} - \bar{p}_i|) & \text{if Acc}(\delta_i) \geqslant \text{Acc}(\delta_0) \\ \delta_0 & \text{if Acc}(\delta_i) < \text{Acc}(\delta_0) \end{cases}$$

Classification by p_i and δ_i:

The i^{th} picture is True if $|p_i - \bar{p}_i| \leqslant \delta_i$, Fake if otherwise.

end

上述算法有两处需要说明：

- 选择77幅画，即从训练集的78幅画中去掉一幅，去掉的原则是如果外层循环中已经剔除了一幅真画则剔除一幅假画，反之则反；
- 之所以要再多去掉一幅画，是为了使得真假画的比例和原来更近似，从而使得均值和极差都更接近总体；
- Acc(δ)即按照以$|p_j^{(i)} - \bar{p}_i| \leqslant \delta$作为判断真的标准计算出来的准确率；
- 在$i = 1$时，没有δ_0，此时就不做判断直接取argmax即可；
- 上述算法为**先验版本**，**后验版本**的区别只在于将确定阈值的步骤替换为取$\bar{p}_i = \bar{p} = \frac{1}{79}\sum_{i=1}^{79} p_i$，$\delta_i = \max|p_i - \bar{p}|/2$。

4.2 数值结果

4.2.1 先验版本

在这一小节，我们将给出先验版本BootLog算法的数值结果。

我们首先说明使用Bootstrap后对原始的loglog Generalized Linear Model的改进。

事实上，如果直接对loglog Generalized Linear Model进行拟合，则利用Leave-One-Out测试得到的各个画作的p如下（取Bootstrap次数$k = 30$）：

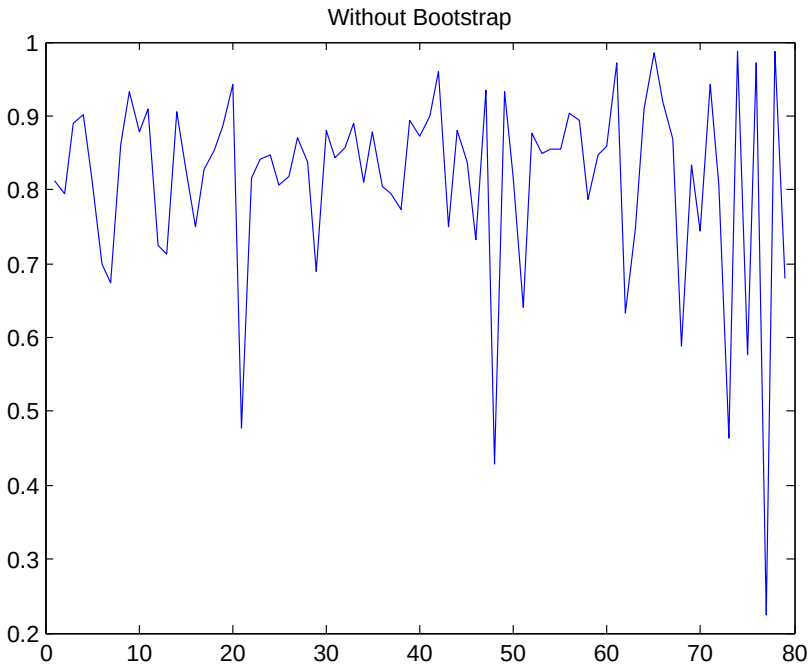


Figure 11: 不进行Bootstrap

但如果用BootLog算法，则得到的p如下：

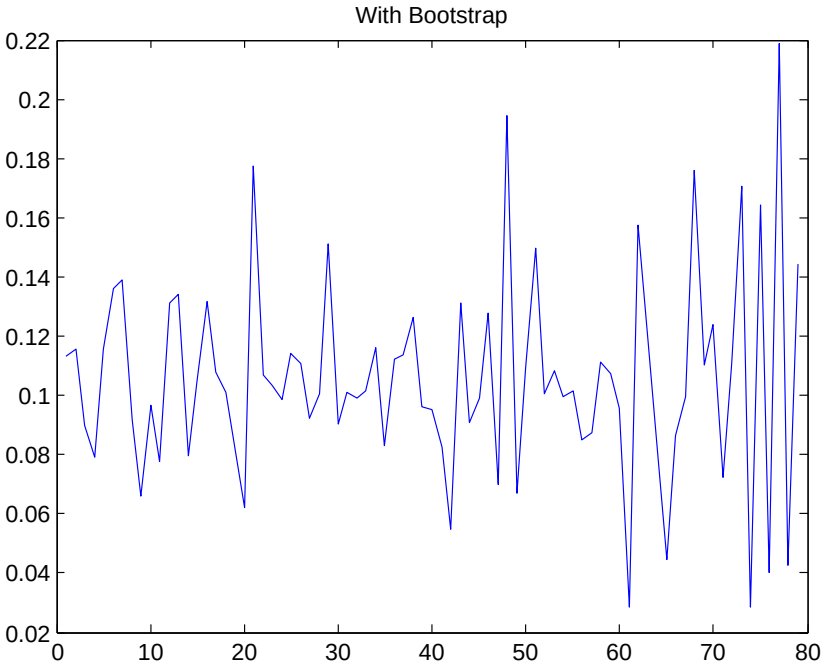
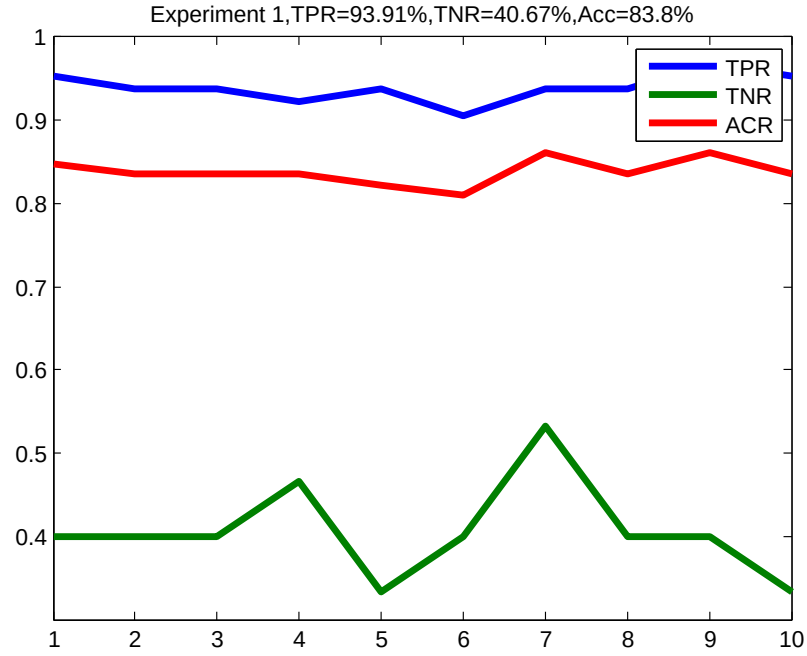


Figure 12: 不进行Bootstrap

可以看到，进行Bootstrap之后，p的第65-79分量（相对于均值）明显波动得更厉害了（且正负波动更均匀），而这自然有助于将对应的赝品分离出来。这正说明了使用Bootstrap的合理性。

接下来，我们考察BootLog算法的实际表现。鉴于Bootstrap方法具有随机性，所以我们需要进行若干次数值实验来衡量其表现（以下TPR,TNR,Acc均取算术平均值）。

我们首先取Bootstrap次数$k = 30$，进行两组（每组10次）实验，得到如下结果：



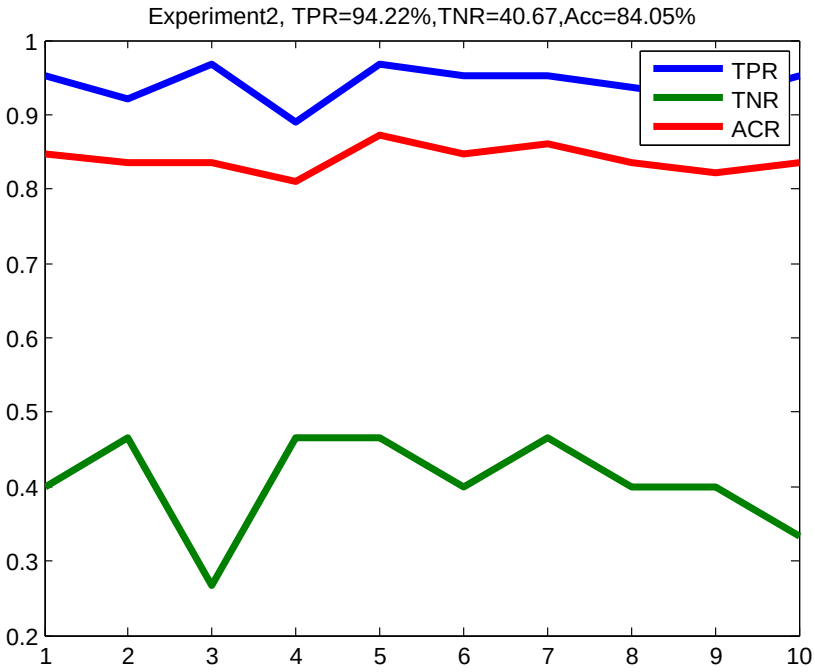


Figure 13: $k = 30$，两次实验

可以看到，虽然有随机性，但至少从平均效果上来说，BootLog已经略微超过了[1]中的方法的83.54%的准确率（在4-Features的前提下）。需要注意的是，BootLog的TNR比53.33%小，从这一点上考虑，BootLog的表现并没有真正超过[1]。

还应该指出的是，这里TNR已经达到了40% 以上，同时TPR也大于[1]中的90.62%。不仅如此，如果增大k到40，则我们可以得到如下更好的结果：

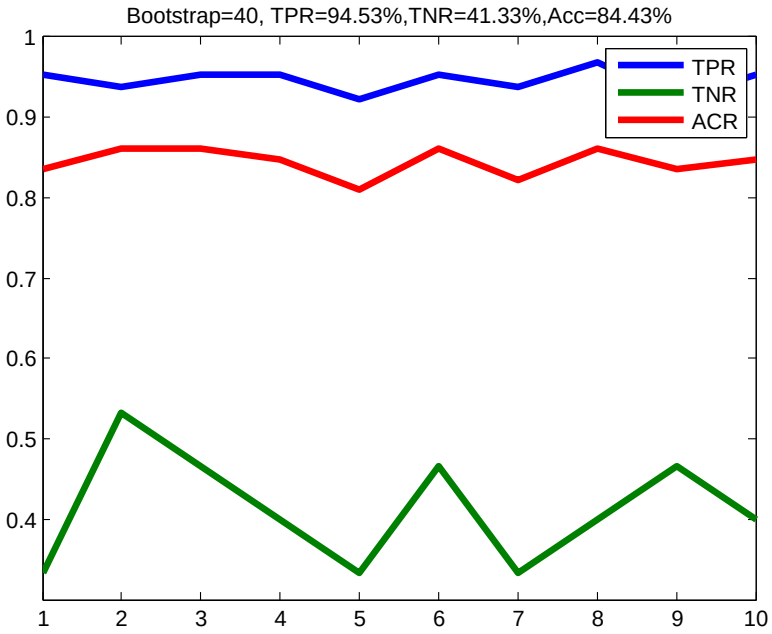


Figure 14: $k = 40$，更好的结果

可以推测，如果时间允许，我们还可以增大k来获得更高的TNR和Acc，从而有可能真正的全面地超越[1]。但即便如此，我们仍然需要考虑如何提高算法的运行效率，因为就目前的实现形式而言，增大k的代价还是很大的。

4.2.2 后验版本

我们再给出后验版本的数值结果。注意到后验版本实际上利用了所有画的真伪的信息，所以即使结果很好，也并不能直接投入使用。

我们的数值实验表明，运行10次BootLog（后验版本），取$k = 30$，最优情形可以达到$TPR = 95.31\%, TNR = 53.33\%, Acc = 87.34\%$，较差的情形则可能只有$TPR = 85.94\%, TNR = 53.33\%, Acc = 79.75\%$或$TPR = 95.31\%, TNR = 40\%, Acc = 84.81\%$或$TPR = 87.50\%, TNR = 60\%, Acc = 82.28\%$。

一组（10次）实验整体结果如下：

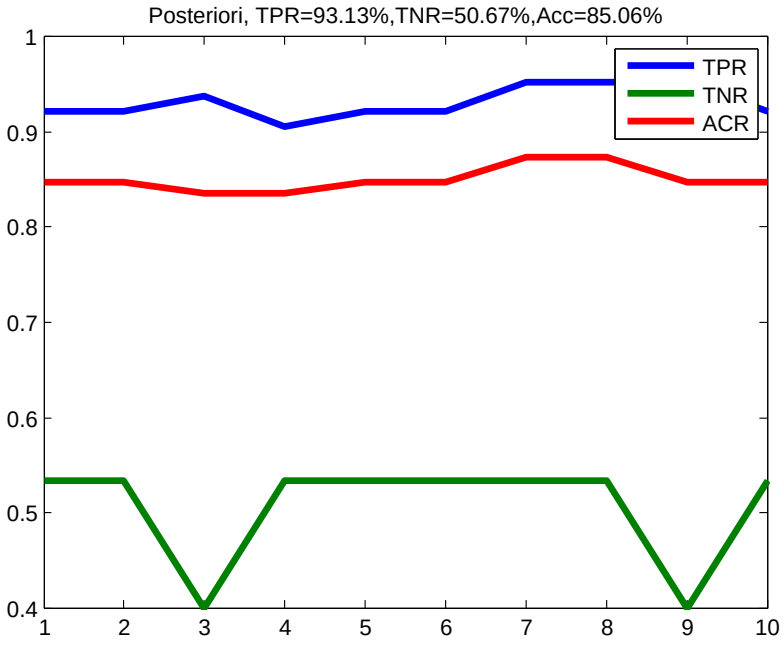


Figure 15: 后验版本的结果

可以看到，结果是较为理想的，并且时间代价也比先验方法小很多。

为了给出一个**相对实用的后验方法**，我们特别地根据经验近似取定$\bar{p} = 0.1, \delta = 0.05$，则可以得到如下结果：

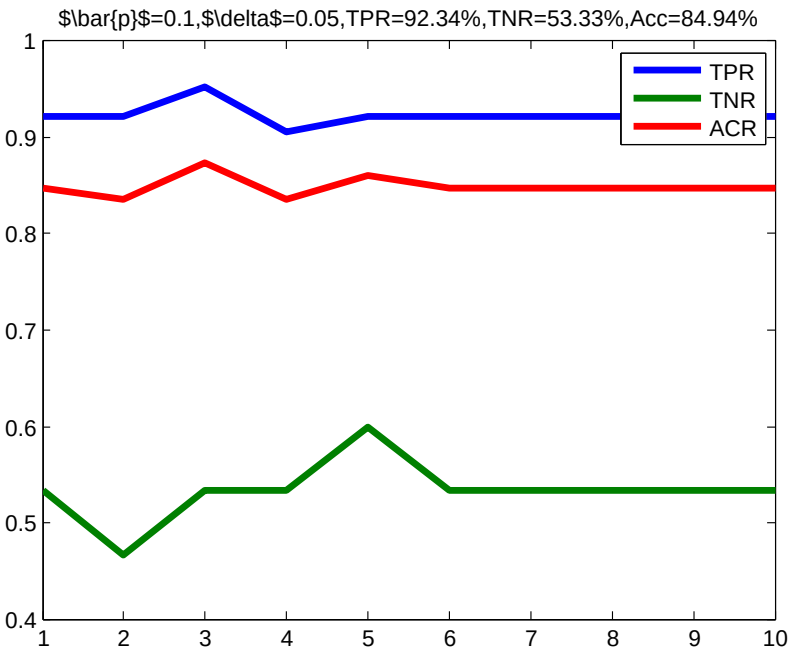


Figure 16: 固定$\bar{p}$和δ

这个结果事实上在一定程度上（如果更重视验伪的话）已经超越了普通的后验方法，和先验方法相比的优越性则更加明显。这是有些出乎我们意料的，但也说明，BootLog算法是有相当的稳健性的。

所以在实际问题中，如果有一定的经验或先验地知道一些关于$\bar{p}$和δ 的大致范围的信息，我们就可以直接取定这组值，然后同样得到较为理想的结果。

4.3 小结

我们将BootLog算法的结果总结如下（这里均取$k = 30$，且均按照多组实验取一个相对合理的值给出）：

Table 5: BootLog算法不同情况结果比较

采用方法	TPR	TNR	Acc
BootLog+先验+平均	94.22%	40.67%	84.05%
BootLog+先验+最优	**95.31%**	40%	84.81%
BootLog+后验+平均	93.13%	50.67%	85.06%
BootLog+后验+最优	**95.31%**	53.33%	87.34%
BootLog+实用后验+平均	92.34%	53.33%	84.94%
BootLog+实用后验+最优	**95.31%**	**60%**	**88.61%**

可以看到，实用后验BootLog的最优情形已经完全超越了[1]，并且甚至不输[1]中的最优情形（此时不受4-Features的限制）。

5 总结

在这篇报告中，我们提出了两类算法来改进[1]的结果。其中，第一类算法的特点是“精化”和确定性，第二种方法的特点是“粗化”和随机性。

从这些算法中，我们可以总结出如下结论：

- “精化”算法在“验真”上更有优势，并且普遍时间代价较小（除了后验选取λ），并且结果是确定的，即不会出现波动，表格中所列出的值是可以不断重复的；
- “粗化”算法在“验伪”上更有优势，并且普遍时间代价较大（除了实用后验版本），并且结果是随机的，即会出现波动，表格中所列出的值只是偶尔才能遇到，会有一定变化；
- “粗化”算法的随机性是一把双刃剑，一方面我们存在得到非常好的结果的可能性（如表5最后一行），但另一方面也存在得到非常差的结果的可能性，可谓是牺牲稳定而追求“极致”的一种体现；
- 后验算法虽然一般不具有实用性，但如果具有一定的先验知识，仍然是可用的，并且效果可能是最好的（如实用后验BootLog算法）；
- “验伪”永远是一个更具有挑战性的任务，所以“打假斗士”还是挺值得尊重的。

6 分工

在本次小课题中，胡安然和钱鹏宇共同负责了课题的选定及文献的查找。钱鹏宇负责了文献[1]的阅读和数值实验的重复，并设计了基于Boosting的算法（第3节），完成了相应的数值实验。胡安然则在与钱鹏宇讨论后提出了BootLog算法，并完成了相应的数值实验。在报告撰写过程中，钱鹏宇主要负责了3小节的撰写，胡安然主要负责了4小节及1、2两个较短小节的撰写，并对3小节做了一点补充。5小节由胡安然和钱鹏宇共同完成。

此外，我们还要感谢姚远老师和熊杰超学长一学期以来的悉心指导，并感谢姚远老师的文献[1]给我们带来的灵感。

参考文献

[1]Haixia Liu, Raymond H.Chan, Yuan Yao, Geometric Tight Frame Based Stylometry for Art Authentication of van Gogh Paintings, 2014.