计算机视觉 问题包-3

黄榕基
519030910026

Problem 1

证明:

点 (u, v) 关于直线 $ax+by+c=0$ 的垂线方程为 $bx-ay-bu+av=0$（该表达式包括了斜率不存在和斜率为 0 的情况）。联立两条直线方程:

$$\begin{cases} ax+by+c=0 \\ bx-ay-bu+av=0 \end{cases}$$

解得两条直线交点即垂点的坐标为 $(\frac{b^2u-abv-ac}{a^2+b^2}, \frac{-abu+a^2v-bc}{a^2+b^2})$。点到直线的距离即为点到该点关于该直线的垂点的距离，因此点到直线的距离为:

$$\begin{aligned} d &= \sqrt{(u-\frac{b^2u-abv-ac}{a^2+b^2})^2+(v-\frac{-abu+a^2v-bc}{a^2+b^2})^2} \\ &= \sqrt{(\frac{a^2u+abv+ac}{a^2+b^2})^2+(\frac{abu+b^2v+bc}{a^2+b^2})^2} \\ &= \sqrt{\frac{a^2(au+bv+c)^2}{(a^2+b^2)^2}+\frac{b^2(au+bv+c)^2}{(a^2+b^2)^2}} \\ &= \sqrt{\frac{(a^2+b^2)(au+bv+c)^2}{(a^2+b^2)^2}} = \sqrt{\frac{(au+bv+c)^2}{a^2+b^2}} \\ &= \frac{|au+bv+c|}{\sqrt{a^2+b^2}} \end{aligned}$$

当 $a^2+b^2=1$ 时，点 (u, v) 到直线 (a, b, c) 的距离则为 $|au+bv+c|$。

得证。

Problem 2

EM 算法可以用于解决数据缺失的参数估计问题。对于共 N 个样本服从的混合了 K 个高斯单模型的高斯混合模型 (π, μ, Σ)，缺失的数据为隐变量 $z_{nk} \in \{0, 1\}$（表示样本 x_n 是否来源于模型 k）。EM 算法则要在只知道混合模型中各个类的分布模型和所有的采样数据的情况下，估计出这些采样数据分别来源于哪一类。

基本原理

EM 算法主要分为 E-step（Expectation-step）和 M-step（Maximization-step）：前者基于上次迭代的混合高斯分布参数来对采样数据做分类划分，即对隐变量 z_{nk} 求期望；后者则在前者的

基础上，计算新的对数似然函数，然后通过最大化对数似然函数来优化混合高斯分布的参数，如此不断迭代循环，直到收敛。

流程

1. 初始化

初始化高斯混合模型的参数 μ_k, Σ_k, π_k，并计算出对数似然函数的初始值。

2. E-step

为了方便 EM 算法估计高斯混合模型的参数，定义 $\gamma(z_{nk}) := p(z_{nk}=1|x_n)$，表示后验概率。根据贝叶斯公式，可得：

$$\gamma(z_{nk}) = p(z_{nk}=1|x_n) = \frac{p(z_{nk}=1)p(x_n|z_{nk}=1)}{\sum_{j=1}^{K} p(z_{nj}=1)p(x_n|z_{nj}=1)} = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$$

E-step 中对隐变量 z_{nk} 求期望即为

$$\begin{aligned} E(z_{nk}|x_n) &= 1 \cdot p(z_{nk}=1|x_n) + 0 \cdot p(z_{nk}=0|x_n) \\ &= p(z_{nk}=1|x_n) \\ &= \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)} \end{aligned}$$

3. M-step

根据要更新的参数，将高斯混合模型改写成 $p(x|\pi,\mu,\Sigma)$，我们要最大化似然函数来进行这些参数的更新。似然函数和对数似然函数为：

$$\begin{aligned} p(X|\pi,\mu,\Sigma) &= \prod_n p(x_n|\pi,\mu,\Sigma) \\ \log p(X|\pi,\mu,\Sigma) &= \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k) \end{aligned}$$

当对数似然函数取得最大值时，即满足对参数的一阶导等于 0，通过该方法可求得模型 μ_k, Σ_k 的估计值：

$$\begin{aligned} \mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \\ \Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n-\mu_k)(x_n-\mu_k)^\top \end{aligned}$$

其中 $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$。

而参数 π_k 的估计，则要考虑限制条件 $\sum_{j=1}^{K} \pi_j = 1$。可通过拉格朗日乘数法来求解，即求 $L = \log p(X|\pi,\mu,\Sigma) + \lambda(\sum_{j=1}^{K} \pi_j - 1)$ 的极大值，满足一阶导等于 0，可解得 $\pi_k = \frac{N_k}{N}$。

因此 M-step 对参数的更新为

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$$

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(x_n-\mu_k)(x_n-\mu_k)^\top$$

$$\pi_k = \frac{N_k}{N}$$

其中 $N_k=\sum_{n=1}^{N}\gamma(z_{nk})$。

4. 检验

计算对数似然函数并检验是否收敛，若未收敛则重复步骤 2 和 3。

Problem 3

关于图像分割的 Mean-shift 算法的伪代码如 Algorithm 1 所示。

Algorithm 1 Mean Shift Algorithm for Segmentation

Input: picture features $\Gamma=(X,Y,L,U,V)$, kernel Φ, bandwidth B, distance threshold D.

Output: picture segmentation $\Lambda=(X,Y,C)$, where $C=(c_1,...,c_n)^\top$, $c_p\in\{1,...,k\}$ refers to which cluster the pixel p is in.

Initialization:
  $i := 0$ /*cluster serial number*/
  $\Lambda := []$
  $M := []$ /*final mean points that each pixel shifts to*/
for each pixel p **do**
  $P \leftarrow \Gamma[p]$
  do
    $N \leftarrow neighbors(P,\Gamma,B)$ /*find neighbor pixels around p within the bandwidth B*/
    $S \leftarrow shiftVector(N,\Phi)$ /*compute the shifting vector to the point of local maximum density among neighbors N weighted by kernel Φ*/
    $P \leftarrow P+S$
  while $\|S\|>D$
  $M[p]\leftarrow P$
for each pixel p **do**
  $\Lambda[p].x,y \leftarrow \Gamma[p].x,y$
  if exists $M[p']$ such that $\|M[p']-M[p]\|\le B\wedge p'<p$ **then**
    $\Lambda[p].c\leftarrow\Lambda[p'].c$
  else
    $i\leftarrow i+1$
    $\Lambda[p].c\leftarrow i$

Mean-shift 算法中需要输入的参数为图像特征 Γ、判断是否属于邻近点的带宽 B、计算偏移向量所需的函数核 Φ、判断是否收敛的阈值 D，因此影响算法性能的主要因素有：

1. 特征子的选取：一般选择使用 5 维的 RGB 信息转化的 5 维的 LUV 信息（LUV 相较 RGB 可保证色彩的均匀性），但也可使用 128 维的 SIFT 特征子来获得更好的效果，当然复杂度也更高。
2. 带宽的大小：影响收敛的效果和速度。
3. 计算偏移向量所需的函数核的选取：影响收敛的效果和速度。
4. 判断是否收敛的阈值的选取：影响收敛的效果和速度。

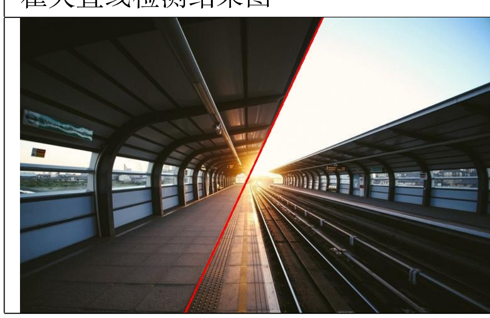
Problem 4

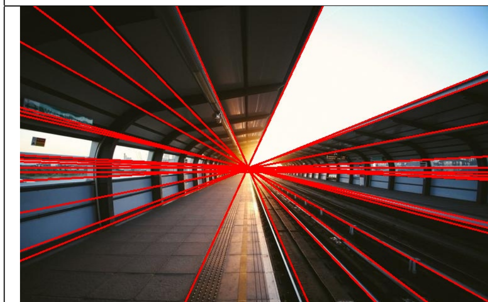
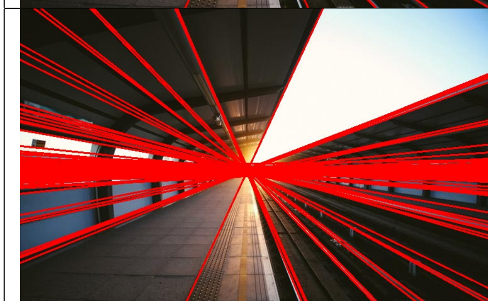
首先将图像转化为灰度图，再通过 Canny 边缘检测算法得到检测出边缘的二值图，如图 1 所示。

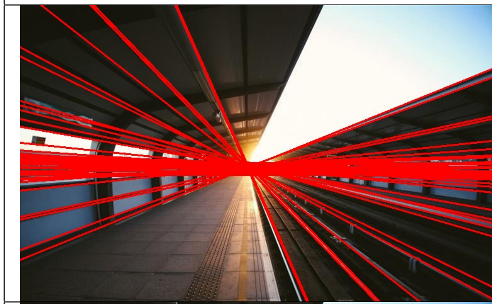
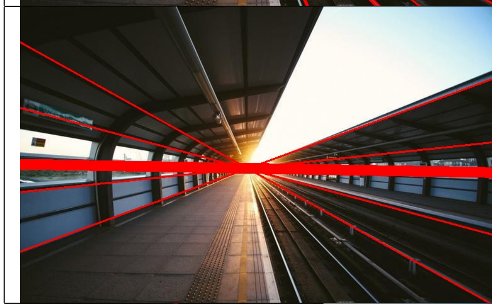
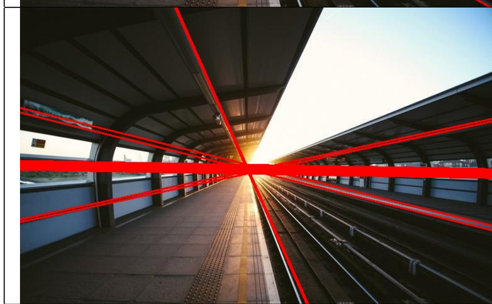
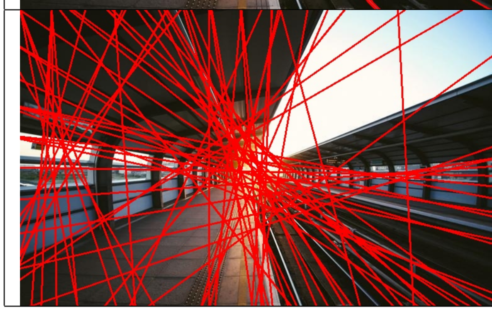
(a) original　　(b) gray　　(c) edges by Canny

图 1: 预处理

然后通过调用 cv2 库中的 HoughLines 函数对二值图进行处理得到检测出来的边缘。HoughLines 函数有三个参数：ρ、θ、$threshold$，对这三个参数尝试不同设置，得到以下结果：

霍夫直线检测结果图	ρ	θ	$threshold$	线条数
	1	1	300	1

霍夫直线检测结果图	ρ	θ	$threshold$	线条数
	1	1	250	17
	1	1	200	62
	1	1	150	227
	1	1	100	1440
	1	1	10	17760

霍夫直线检测结果图	ρ	θ	*threshold*	线条数
	1	2	200	39
	1	7	200	10
	1	13	200	12
	1	17	200	5
	50	1	200	62

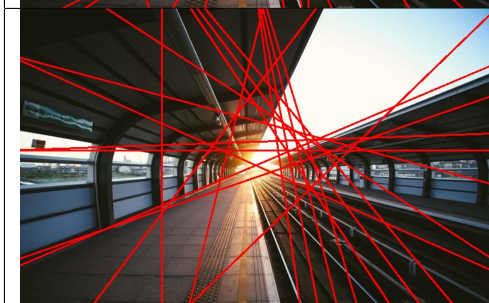
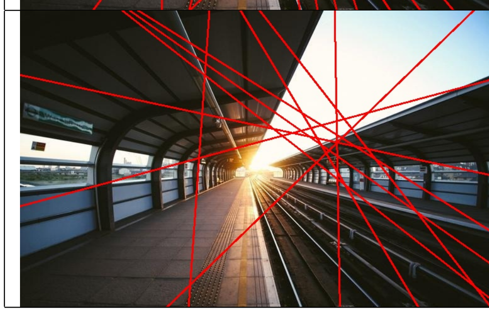
霍夫直线检测结果图	ρ	θ	$threshold$	线条数
	100	1	200	19
	150	1	200	18
	200	1	200	10

表 1: 霍夫直线检测结果

结果比较

1. 阈值 $threshold$ 为一条直线所需最少的曲线交点，$threshold$ 越大，能够检测到的直线越少，该参数过大可能导致检测不到直线。

2. ρ 为检测时的极径步长，ρ 越大则检测精度降低，部分直线可能被跳过，能够检测到的直线越少。

3. θ 为检测时的极角步长，θ 越大则检测精度降低，部分角度的直线可能无法检测出，能够检测到的直线也就越少。

Problem 5

将图片中的文字添加删除线整体流程如下：

1. 对图片进行高斯模糊降噪处理。

2. 对高斯模糊的图片进行（自适应）阈值二值化，突出文字主体。

3. 对二值化的图片进行膨胀处理，将每排文字膨胀为一整个矩形，方便霍夫直线检测；再对膨胀处理后的图片进行腐蚀处理，将每排矩形变细，使得霍夫直线检测出的直线能在更准确的位置。

4. 对腐蚀处理后的图片进行霍夫直线检测，通过二值图判断直线的起始点和结束点，将得到的直线单独放在一张图内。

5. 对这些直线进行膨胀处理，使得每一排文字对应的直线整合在一起；再对整合在一起的直线进行腐蚀处理使其变细，更加符合删除线。

6. 最后将得到的删除线和原图结合在一起得到结果图，如图 2 所示。

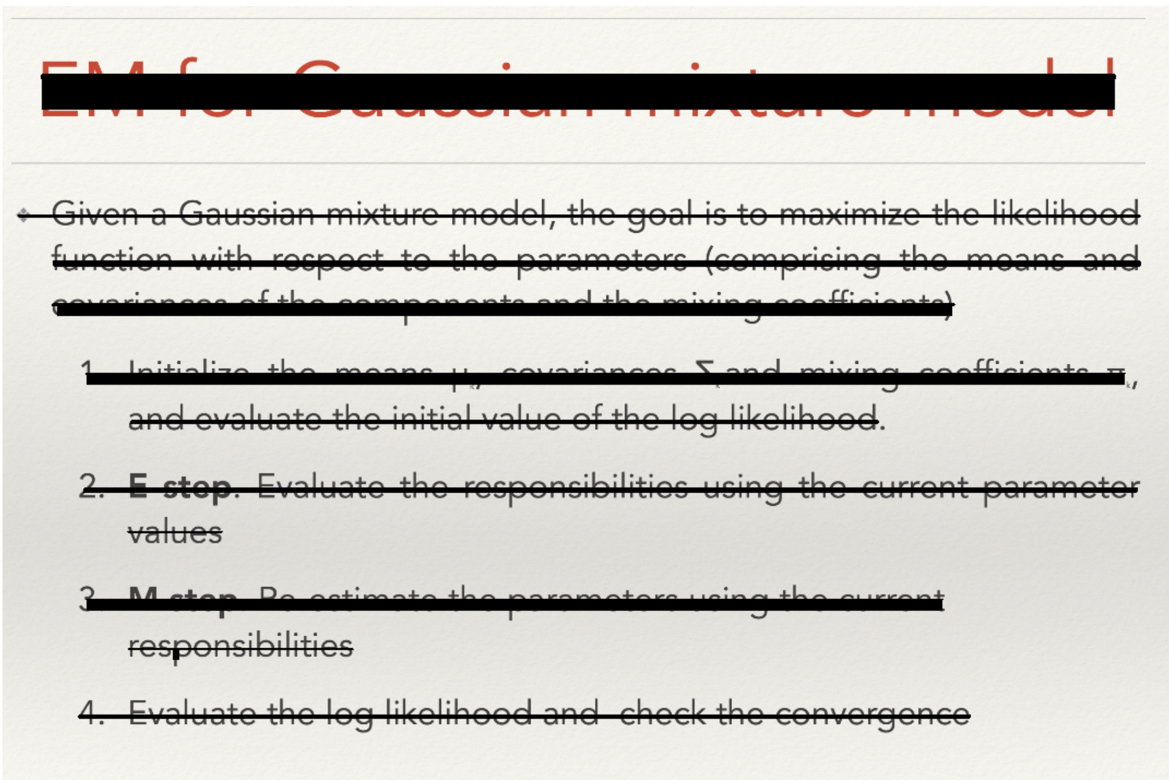


图 2: 对文字添加了删除线的图片