BAT\_Machine\_Learning\_1000

uraboer

2018年1月4日

## 1.请简要介绍下SVM

SVM,全称是support vector Machine,中文名叫支持向量机。SVM是一个面向数据的分类算法,它的目标是为确定一个分类超平面，从而将不同的数据分隔开。

SVM原理、推导 <http://blog.csdn.net/v_july_v/article/details/7624837>

推导视频<http://www.julyedu.com/video/play/18/429>

## 2.请简要介绍下tensorflow的计算图

Tensorflow是一个通过计算图的形式来表述计算的编程系统，计算图也叫数据流图，可以把计算图看作一种有向图，Tensorflow中的每一个计算都是计算图上的一个节点，而节点之间的边描述了计算之间的依赖关系。

* Tensorflow一般可分为两部分

构造部分，包含计算流图

执行部分，通过session来执行图中的计算

* 构造图

创造源节点(source op)

源节点输出传递给其他节点(op)做运算

* TF默认图

TensorFlow Python库有一个默认图(default graph)

节点构造器可以增加节点

## 3.在K-means或KNN，我们常用欧氏距离来计算最近的邻居之间的距离，有时也用曼哈顿距离，请对比下这两种距离的差别。

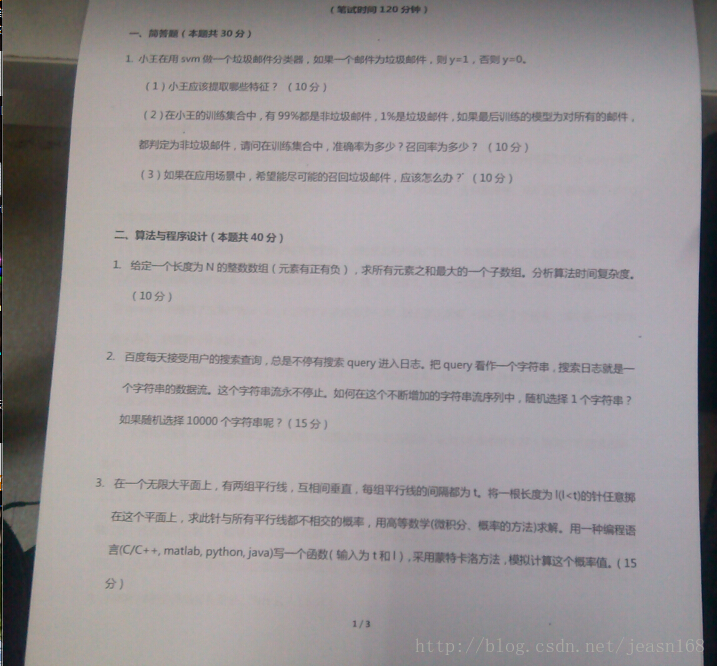
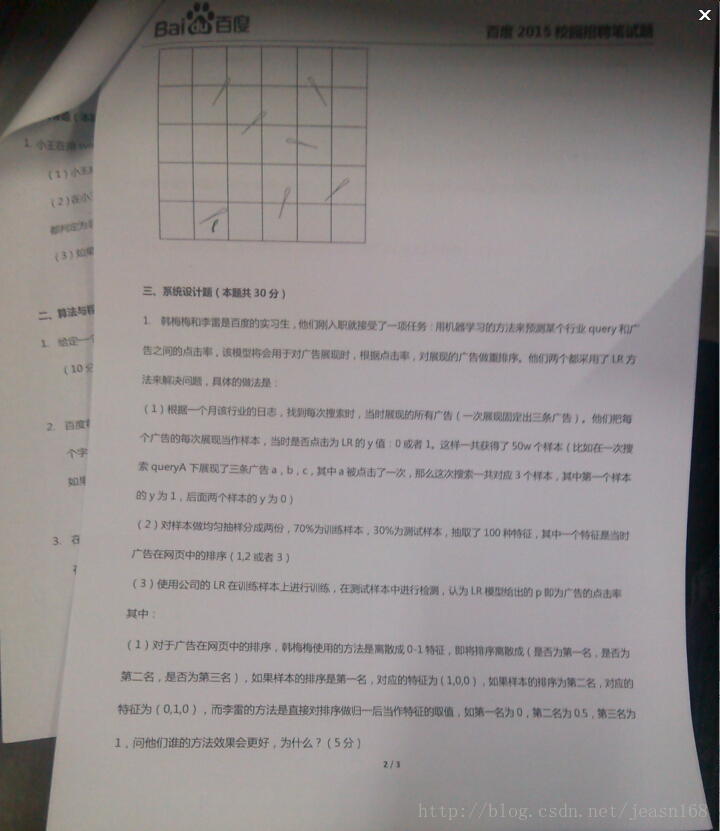
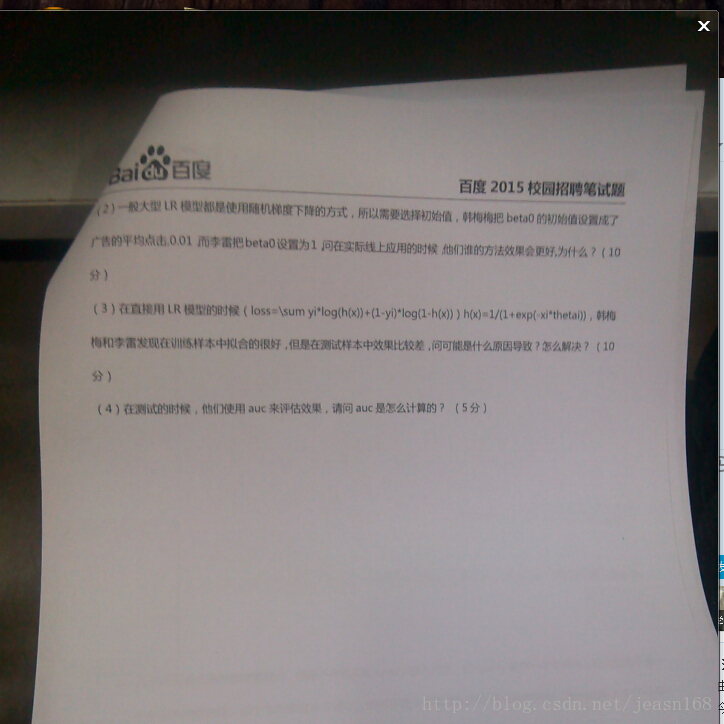
欧式距离，最常见的两点之间或多点之间的距离表示法，又称之为欧几里得度量，它定义于欧几里得空间中，如点和之间的距离为：

曼哈顿距离，我们可以定义曼哈顿距离的正式意义为L1-距离或城市区块距离，也就是在欧几里得空间的固定直角坐标系上两点所形成的线段对轴产生的投影的距离总和。例如在平面上，坐标的点与坐标的点的曼哈顿距离为：。要注意的是，曼哈顿距离依赖坐标系统的转度，而非系统在坐标轴上的平移或映射。

通俗来讲，想象你在曼哈顿要从一个十字路口开车到另一个十字路口，驾驶距离是两点间的直线距离吗？显然不是，除非你能穿越大楼。而实际驾驶距离就是这个“曼哈顿距离”，这也是曼哈顿距离名称的来源，同时，曼哈顿距离也称为城市街区距离(City Block distance)。

各种距离的比较<http://blog.csdn.net/v_july_v/article/details/8203674>

## 4.百度2015校招机器学习笔试题

## 5.关于LR

把LR从头到脚都给讲一遍。建模，现场数学推导，每种解法的原则，正则化，LR和maxent模型啥关系，lr为啥比线性回归好。有不少人会背答案，问逻辑细节就糊涂了。原理都会？那就问工程，并行化怎么做，有几种并行化方式，读过那些开源的实现。还会，那就准备收了吧，顺便逼问LR模型发展历史。

.....

<http://blog.csdn.net/zouxy09/article/details/20319673>

## 6.overfitting怎么解决？

dropout、regularization、batch normalization

## 7.LR和SVM的联系和区别

* 联系：

(1)LR和SVM都可以处理分类问题，且一般都用于处理线性二分类问题

(2)两个方法都可以增加不同的正则化项，如l1、l2等等，所以在很多实验中，两种算法的结果是很接近的

* 区别：

(1)LR是参数模型，SVM是非参数模型

(2)从目标函数来看，却别在于逻辑回归采用的是logistical loss，SVM采用的是hinge loss。这两个损失函数的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。

(3)SVM的处理方法是只考虑support vectors,也就是和分类最相关的少数点，去学习分类器。而逻辑回归通过非线性映射，大大减少了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重

(4)逻辑回归相对来说模型更简单，好理解，特别是大规模线性分类时比较方便。而SVM的理解和优化相对来说复杂一些，SVM转化为对偶问题后，分类只需要计算与少数几个支持向量的距离，这个在进行复杂核函数计算时优势很明显，能够大大简化模型和计算

(5)logic能够做的svm能做，但可能在准确率上有问题，svm能做的logic有的做不了

<http://blog.csdn.net/timcompp/article/details/62237986>

## 8.说说你知道的核函数

通常人们会从一些常用的核函数中选择，例如，显然刚才我们举的例子是这里多项式核的一个特例(R=1,d=2)。虽然比较麻烦，而且没有必要，不过这个核所对应的映射实际上是可以写出来的，该空间的纬度是，其中m是原始空间的维度

高斯核，这个核就是最开始提到过的会将原始空间映射为无穷维空间的那个家伙。不过，如果选得很大的话，高次特征上的权重实际上衰减的非常快，所以实际上相当于一个低维的子空间；反过来，如果选得很小，则可以将任意的数据映射为线性可分——当然，这并不一定是好事。不过，总的来说，通过调控参数，高斯核实际上具有相当高的灵活性，也是使用最广泛的核函数之一。下图所示的例子便是把低维线性不可分的数据通过高斯核函数映射到了高维空间：