## Hands-On Machine Learning

## 机器学习

Most problems can be solved quite well using simpler techniques such as Random Forests and Ensemble methods.

Deep Learning is best suited for complex problems such as image recognition, speech recognition, or natural language processing, provided you have enough data, computing power, and patience.

### The Machine Learning Landscape

• Machine Learning **is about** making machines get better at some task by learning from data, instead of having to explicitly code rules.  
• There are many different **types** of ML systems: supervised or not, batch or online, instance-based or model-based, and so on.  
• In **a ML project** you studied the data, selected a model, trained it on the training data, finally, applied the model to make predictions on new cases  
• The system will not **perform** well if your training set is too small, or if the data is not representative, noisy, or polluted with irrelevant features (garbage in, garbage out). Lastly, your model needs to be neither too simple (in which case it will underfit) nor too complex (in which case it will overfit).

判断：*learning curves*

对于overfitting的五个方法：

To simplify the model by selecting one with fewer parameters, by reducing the number of attributes in the training data or by constraining the model(*regularization:* Ridge Regression, Lasso Regression, Elastic Net, Early Stopping)

To gather more training data

To reduce the noise in the training data

对于underfitting的三个方法：

Selecting a more powerful model, with more parameters  
Feeding better features to the learning algorithm (feature engineering)  
Reducing the constraints on the model

增加错误较多的样本类型

### Machine Learning Project Checklist

1. Frame the problem and look at the big picture.

目标-表现测试、怎么用-模型类型、过往经验

评估

1. Accuracy is not a good measure.

2.

precision：判断“是”的里面，有多少是对的。

recall：在所有为“真”的对象集里面，能判断出多少个“真”

3.

TPR：和R一样。TNR：在所有为“假”的对象集里面，会判断出多少个“真”

PR和ROC都要权衡，看目标。you should prefer the PR curve whenever the positive class is rare or when you care more about the false positives

2. Get the data.

类和量、空间、合法性、创建工作空间、sample测试集(shuffle the training set)

3. Explore the data to gain insights.

专家建议、copy、研究（数据类型、缺失、噪音、分布）、identify target feature

4. Prepare the data to better expose the underlying data patterns to Machine Learning algorithms.

cleaning (outliers, missing values), feature selection, engineering(discretize continuous features, decompose, log/ sqrt etc. transformation, aggregate), scaling

discretize continuous features:

一般不人工选

quantiles

find the intervals which maximize the information gain of discretizing the feature with respect to the target variable？

Kmean

decision tree

5. Explore many different models and short-list the best ones.

每个模型最好的变量、各模型的错误类型（怎么解决）、选出犯不同错误的模型

Error Analysis

conf\_mx = confusion\_matrix(y\_train, y\_train\_pred) 左边为真，右边为判断

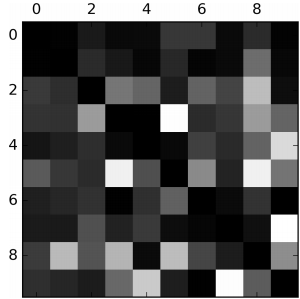
plt.matshow(conf\_mx, cmap=plt.cm.gray)

在对角线上，颜色越深，错误越多

row\_sums = conf\_mx.sum(axis=1, keepdims=True)  
norm\_conf\_mx = conf\_mx / row\_sums

np.fill\_diagonal(norm\_conf\_mx, 0)  
plt.matshow(norm\_conf\_mx, cmap=plt.cm.gray)

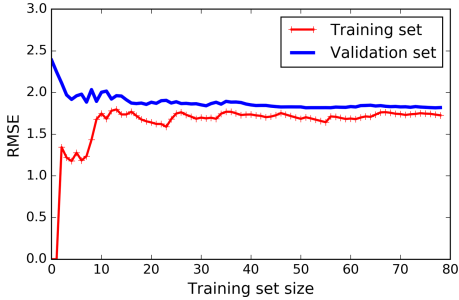
越亮，说明错误多。row为“真”，col为判断；左下角为去真，右上角为存伪；错误不对称。



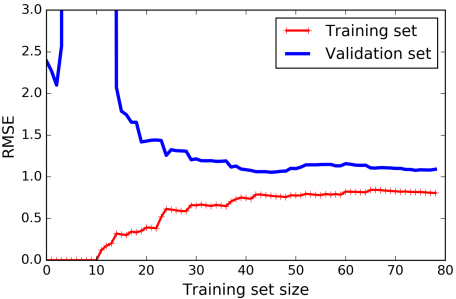
上图可以看出，要改善8s and 9s, as well as fixing the specific 3/5 confusion。比如加大这些数的样本，新features（计算环数），或预处理图像（Scikit-Image, Pillow, or OpenCV）图像模式（如环）更清晰

overfitting or not: cross-validation and *learning curves*

underfitting: Both curves have reached a plateau; they are close and fairly high.



overfitting: There is a gap between the curves, the error on the training data is much lower



model’s generalization error comes from Bias(Increasing complexity), Variance(reducing complexity) and Irreducible error(clean up the data)

6. Fine-tune your models and combine them into a great solution.

missing values的处理视为超参数， 调参（random, grid search, Bayesian optimization）, 组合模型

7. Present your solution.

Document what you have done, assumptions and limitations, Explain why your solution achieves the business objective. what worked and what did not

8. Launch, monitor, and maintain your system.

Launch: plugging the production input data sources into your system and writing tests

Monitor: write monitoring code to check your system’s live performance at regular intervals and trigger alerts when it drops (models tend to “rot” as data evolves over time)

evaluate the system’s input data quality

maintain: train your models on a regular basis using fresh data

If your system is an online learning system, you should make sure you save snapshots of its state at regular intervals so you can easily roll back to a previously working state.

### Classification

评估：PR and ROC

Training a Binary Classifier

**from sklearn.linear\_model import** SGDClassifie

**from sklearn.ensemble import** RandomForestClassifier

Multiclass Classification

Random Forest classifiers

naive Bayes classifiers

OvO Support Vector Machine classifiers

OvO Linear classifiers

Multilabel Classifcation

目标多label

face recognition

**from sklearn.neighbors import** KNeighborsClassifier

Multioutput Classifcation

目标多label，每个label多维

比如用图片作为x和y，然后输入新图片，输出判断图片。

多维数据作为data point和label

##### 补充：

对于binary classifiers有两种策略：

*one-versus-all* (OvA) strategy (also called *one-versus-the-rest*): training few classifiers on large training sets

*one-versus-one* (OvO) strategy: train many classifiers on small training sets

sklearn默认SVM用 OvO，其他binary classification algorithms用OvA

可以用OneVsOneClassifier or OneVsRestClassifier 改

SGDClassifier会根据各个class的评分得出预测

在错误分析中，由于SGDClassifier 计算的是对各个pixel加权后的总和，3和5只有少数pixels不同

对于评估：

1. Accuracy is not a good measure in classification

2.

precision：判断“是”的里面，有多少是对的。

recall：在所有为“真”的对象集里面，能判断出多少个“真”

3.

TPR：和R一样。TNR：在所有为“假”的对象集里面，会判断出多少个“真”

PR和ROC都要权衡，看目标。you should prefer the PR curve whenever the positive class is rare or when you care more about the false positives

### Regression

评估：L1 or L2

##### Linear Regression

The Normal Equation(100000 features)

训练慢，预测快

sklearn 的LinearRegression用的是pseudoinverse. Its time complexity is O(n2) and it works even when m<n or when some features are linear combinations of other features. All the data must fit in memory, it does not require feature scaling and the order of the instances in the training set does not matter

Gradient Descent

if you divide the tolerance ϵ by 10 (to have a more precise solution), then the algorithm will have to run about 10 times more iterations

Stochastic Gradient Descent

huge training sets, out-of-core algorithm

1. jump out of local minima

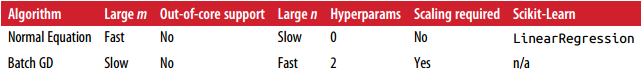
对irregular function有好处

*一般解决方法是simulated annealing（learning rate 由大到小，, learning schedule是决定learning rate变化的函数）)*

2. the cost function will bounce up and down, decreasing only on average

3. If the algorithm shuffle the training set, then go through it instance by instance, then shuffle it again, and so on. This generally converges more slowly than don’t shuffling.

Stochastic GD and Mini-batch GD would also reach the minimum if you used a good learning schedule.



##### Polynomial Regression

在sklearn里，先转化features(degree)，接着scaling，然后应用线性方程

PolynomialFeatures(degree=d) transforms an array containing *n* features into an array containingfeatures

Regularized Linear Models

Note that the regularization term should only be added to the cost function during training. Once the model is trained, you want to evaluate the model’s performance using the unregularized performance measure.

Ridge Regression

Note that the bias term *θ*0 is not regularized

It is important to scale the data before performing Ridge Regression. This is true of most regularized models.

sklearn可以用SGDRegressor或者Ridge

It is quite common for the cost function used during training to be different from the performance measure used for testing.

Lasso Regression

sklearn可以用SGDRegressor或者Lasso

Lasso may behave erratically when the number of features is greater than the number of training instances or when several features are strongly correlated.

Elastic Net

a middle ground between Ridge Regression and Lasso Regression.



sklearn可以用SGDRegressor或者ElasticNet (l1\_ratio corresponds to the mix ratio r)

Early Stopping

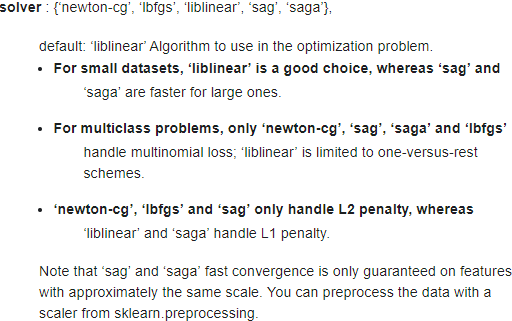
Once the validation error reaches the minimum

With Stochastic and Mini-batch Gradient Descent, the curves are not so smooth, and it may be hard to know whether you have reached the minimum or not. One solution is to stop only after the validation error has been above the minimum for some time (when you are confident that the model will not do any better), then roll back the model parameters to the point where the validation error was at a minimum.

#### Logistic Regression



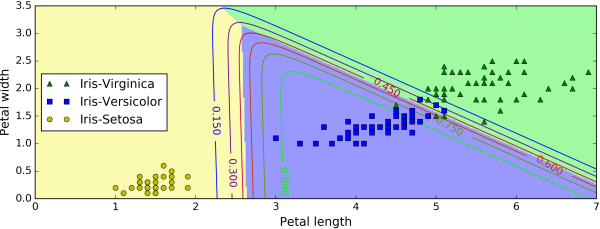


Softmax Regression(*Multinomial Logistic Regression*)

对于一个数据，计算各个class的分数，各个class有相应的w，然后在softmax function里转换成概率，并取最大值的class（实际上就是找最高分的class，但看softmax function可以知道概率）

 *Softmax score*

 *Softmax function*

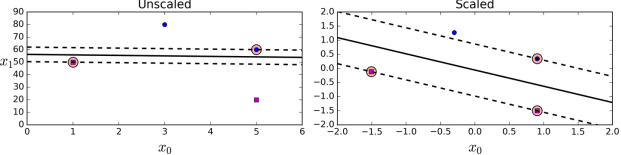


 *Cross entropy cost function（和Logistic Regression 不一样，除非K=2）*

### Support Vector Machines

#### Classification

Linear SVM’s hinge loss using Gradient Descent (SGDClassifier) converges much more slowly than the methods based on QP

SVMs are sensitive to the feature scales 

Soft Margin Classification

*hard margin classification:*  First, it only works if the data is linearly separable, and second it is quite sensitive to outliers.

*soft margin classifcation*

a smaller C value leads to a wider street but more margin violations.

If your SVM model is overfitting, you can try regularizing it by reducing C.

sklearn:

from sklearn.svm import LinearSVC

it converges much more slowly than the methods based on QP.

Nonlinear SVM Classifcation

1. Polynomial Features

2. Polynomial Kernel

3. Adding Similarity Features

 Gaussian *Radial Basis Function* (*RBF*)

a small gamma value makes the bell-shaped curve wider, so instances have a larger range of influence, and the decision boundary ends up smoother. So *γ* acts like a regularization hyperparameter: if your model is overfitting

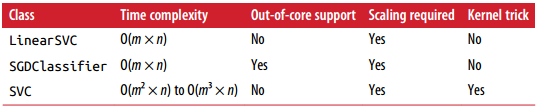
*String kernels* are sometimes used when classifying text documents or DNA sequences (e.g., using the *string subsequence kernel* or kernels based on the *Levenshtein distance*)

实践：You should always try the linear kernel first (LinearSVC), especially if the training set is very large or if it has plenty of features. If the training set is not too large, you should try the Gaussian RBF kernel. Then if you have spare time and computing power, you can also experiment with a few other kernels using cross-validation and grid search, especially if there are kernels specialized for your training set’s data structure.

Computational Complexity

LinearSVC (*liblinear* library and does not support the kernel trick): scales almost linearly with the number of training instances and the number of features. The algorithm takes longer if you require a very high precision. tol in sklearn. In most classification tasks, the default tolerance is fine.

SVC: perfect for complex but small or medium training sets. It scales well with the number of features, especially with *sparse features*



#### Regression

fit as many instances as possible *on* the street while limiting margin violations

SVMs can also be used for outlier detection(see Scikit-Learn’s doc‐umentation for more details - -||)

SVR对应SVC，特性一样

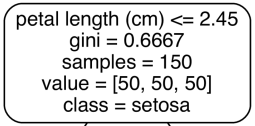
For largescale nonlinear problems, you may want to consider using neural networks instead.

数学补充

The dual problem is faster to solve than the primal when the number of training instances is smaller than the number of features.

### Decision Trees

Decision trees don’t require feature scaling or centering at all

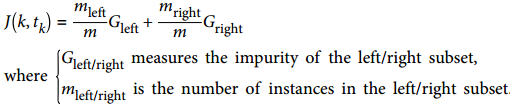


*Gini impurity*

 *pi*,*k* is the ratio of class *k* instances among the training instances in the *i*th node.

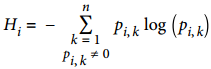
可以提供概率，但在一个node里，概率是不变的，即特征十分明显，但根据该node的纯度，决策树依然不会给一个更高的概率

The CART Training Algorithm



*O*(*log*2(*m*))

If the training algorithm compares all features on all samples at each node. This results in a training complexity of *O*(*n* × *m log*(*m*)). presort=True, but this slows down training con‐  
siderably for larger training sets.

 entropy tends to produce slightly more balanced trees, but Gini impurity is slightly faster.

STOP

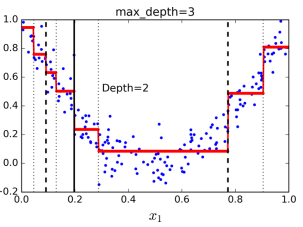
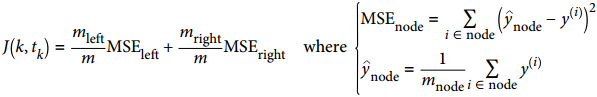
cannot reduce impurity

max\_depth

min\_samples\_split, min\_samples\_leaf, min\_weight\_fraction\_leaf, and max\_leaf\_nodes

Regression

the predicted value for each region is always the average target value of the instances in that region.

Instability

First, as you may have noticed, Decision Trees love orthogonal decision boundaries which makes them sensitive to training set rotation. PCA解决

the main issue with Decision Trees is that they are very sensitive to small variations in the training data.( Scikit-Learn is stochastic6 you may get very different models, randomly selects the set of features to evaluate at each node)

Random Forests can limit this instability by averaging predictions over many trees

### Ensemble Learning and Random Forests

#### voting system

*hard voting* classifier: aggregate the predictions of each classifier and predict the class that gets the most votes.

*soft voting* classifier: predict the class with the highest class probability, averaged over all the individual classifiers.

It often achieves higher performance than hard voting

#### ensemble

Ensemble methods work best when the predictors are as independent from one another as possible.

One way to get diverse classifiers is to train them using very different algorithms.

Another approach is to use the same training algorithm for every predictor, but to train them on different random subsets of the training set. When sampling is performed *with* replacement, this method is called *bagging, otherwise called pasting.* Generally, the net result is that the ensemble has a similar bias but a lower variance than a single predictor trained on the original training set.(意味着test set表现更好) Predictors can all be trained in parallel, via different CPU cores or even different servers. They scale very well.

Bagging ends up with a slightly higher bias than pasting, but this also means that predictors end up being less correlated so the ensemble’s variance is reduced. 通常来说，bagging会更好，但有时间可以用CV决定哪个好。

Out-of-Bag Evaluation

As m grows, this ratio approaches 1 – exp(–1) ≈ 63.212%, training instances are sampled on average for each predictor. The remaining 37% of the training instances that are not sampled are called *out-of-bag* (oob) instances. Note that they are not the same 37%  
for all predictors.

The third approach is to use different feature subsets to train the models. This is particularly useful when you are dealing with high-dimensional inputs (such as images). Sampling both training instances and features is called the *Random Patches* method. Keeping all training instances but sampling features is called the *Random Subspaces* method. Sampling features results in even more predictor diversity, trading a bit more bias for a lower variance.

#### Random Forests

it searches for the best feature among a random subset of features

Extra-Trees

make trees even more random by also using random thresholds for each feature rather than searching for the best possible thresholds

this trades more bias for a lower variance. much faster to train

It is hard to tell in advance whether a RandomForestClassifier will perform better or worse than an ExtraTreesClassifier.

Feature Importance

important features are likely to appear closer to the root of the tree, while unimportant features will often appear closer to the leaves (or not at all)

It is therefore possible to get an estimate of a feature’s importance by computing the average depth at which it appears across all trees in the forest.

Boosting

AdaBoost

One way for a new predictor to correct its predecessor is to pay a bit more attention to the training instances that the predecessor underfitted. This results in new predictors focusing more and more on the hard cases.

SVMs are generally not good base predictors for AdaBoost, because they are slow and tend to be unstable with AdaBoost.

The drawback is the sequential learning technique, it does not scale as well as bagging or pasting

If your AdaBoost ensemble is overfitting the training set, you can try reducing the number of estimators or more strongly regularizing the base estimator.

Scikit-Learn actually uses a multiclass version of AdaBoost called *SAMME.* When there are just two classes, SAMME is equivalent to AdaBoost.

Gradient Boosting

this method tries to fit the new predictor to the *residual errors* made by the previous predictor

If you set it to a low value, such as 0.1, you will need more trees in the ensemble to fit the training set, but the predictions will usually generalize better. In order to find the optimal number of trees, you can use early stopping or staged\_predict() method

Stacking

神经网络？

### Dimensionality Reduction

Reducing dimensionality does lose some information, so even though it will speed up training, it may also make your system perform slightly worse(it all depends on the dataset). So you should first try to train your system with the original data before considering using dimensionality reduction if training is too slow.

The Curse of Dimensionality

the more dimensions the training set has, the greater the risk of overfitting it

In theory, one solution to the curse of dimensionality could be to increase the size of the training set to reach a sufficient density of training instances. 实际上需要的数据量太多

Main Approaches for Dimensionality Reduction

Projection

Many features are almost constant, while others are highly correlated. As a result, all training instances actually lie within (or close to) a much lower-dimensional *subspace* of the high-dimensional space.

However, projection is not always the best approach to dimensionality reduction. In many cases the subspace may twist and turn.

Manifold Learning

a 2D manifold is a 2D shape that can be bent and twisted in a higher-dimensional space.

*manifold hypothesis*: most real-world high-dimensional datasets lie close to a much lower-dimensional manifold.

PCA

preserves the maximum amount of variance or minimizes the mean squared distance

The unit vector that defines the ith axis is called the ith *principal component* (PC)

The direction of the principal components is not stable. 数据稍微变动，PCs的方向就可能不同，但仍在同一根轴。两根PCs构成的平面同理

PCA assumes that the dataset is centered around the origin. Scikit-Learn’s PCA classes已经设置

*reconstruction error:* The mean squared distance between the original data and the reconstructed data (compressed and then decompressed).

Incremental PCA

很慢，但适合online。

Randomized PCA



Kernel PCA

It is often good at preserving clusters of instances after projection, or sometimes even unrolling datasets that lie close to a twisted manifold.

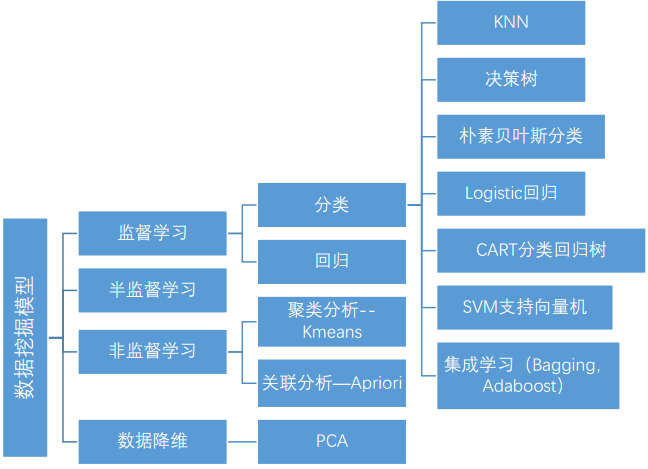
LLE

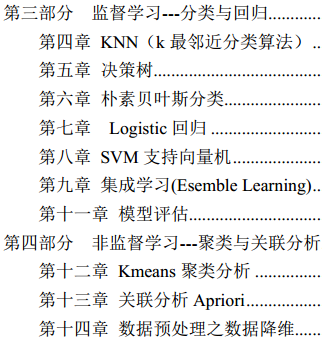
In a nutshell, LLE works by first measuring how each training instance linearly relates to its closest neighbors (c.n.), and then looking for a low-dimensional representation of the training set where these local relationships are best preserved (more details shortly).

This makes it particularly good at unrolling twisted manifolds, especially when there is not too much noise.

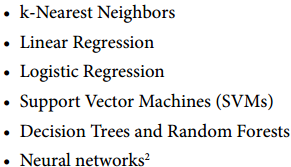
distances are not preserved on a larger scale







### 监督学习



目标：规则化参数的同时最小化误差

#### KNN

1.基本思路：

To classify a new point: find its k nearest neighbour amongst the x(i) using distance function, then return label.

2.提升：

用Cross-validation找估计k；其他distance function； feature selection； faster search algorithm

距离没有加权、变量标准化、样本权重、压缩样本量

3.算法优缺点  
**1)** 优点：无需估计参数，无需训练；样本容量比较大的分类；适合多分类问题，比SVM 好？；适合类域的交叉或重叠较多  
**2)** 缺点：内存开销大，评分慢；可解释性较差；样本量较小，会产生误分；高维度时，欧式距离的区分能力较差

#### Decision Tree

1.基本思路：按某一标准不断划分样本，直到达到一定条件为止。

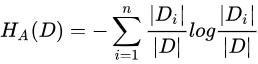
2. 具体过程

特征选择：**ID3**信息增益，**C4.5**信息增益比，**CART** 基尼指数或平方误差最小化

1）**ID3**信息增益（information gain）：

这里条件熵H(D|A)表示某一划分后的熵。A代表某种划分。找出使g(D,A)最大的一种划分。

2）**C4.5**信息增益比

其中 ，新划分下的熵

3）**CART** 基尼指数或平方误差最小化

(1)基尼指数（分类树）



增益最大：Gini(p) - Gini(D,A)

(2)平方误差最小化(回归树)



c1,c2分别为左右子区域输出的均值

决策树生成：递归结构，对应于模型的局部最优

提前停止或最后剪枝

3.算法优缺点  
**1)** 优点：无需估计参数；速度快；便于理解；准确性高；连续和种类字段？；适合高维度  
**2)** 缺点：对于各类别样本数量不一致的数据，信息增益偏向于哪些具有更多数值的特征；易于过拟合；忽略属性之间的相关性

#### Generative Modeling

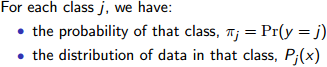
1. 基本思路：

The learning process: Fit a probability distribution to each class, individually

To classify a new point: Pick the label with largest P(x,y)

2.具体过程

Which of these distributions (class j) was a data point(x) most likely to have come from?

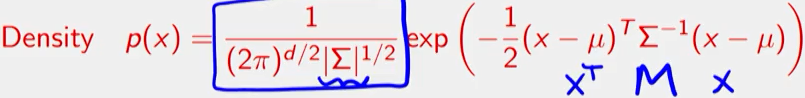
 

实际上只需要找出哪个class能使分子最大，只有它影响大小。

(1)Fitting

pi of j就是样本中某class j出现次数与样本总量n的比

Pj(x)（这里假设个class和各feature服从Gaussian distribution）:



不同class代入不同的期望和协方差（从样本中计算得出）

注意这里的mu是d\*1，covariance matric是d\*d

(2)classifying

 当特征很多的时候，大量小数值的小数乘法会有溢出风险。因此，通常的实现都是将其转换为 log

3.算法优缺点  
**1)** 优点：可以和决策树、神经网络分类算法相媲美，能运用于大型数据库；所需估计的参数少，对于缺失数据不敏感

**2)** 缺点：假设特征独立或者同分布；需要知道先验概率

4. 补充：

1）当feature独立时为朴素贝叶斯分类





由于相当节省空间，所以有时两个变量不是uncorrelated，也当作是。注意任意特征  
xi，P(xi)不能为0。处理方法有：

所有计数 +1加法平滑(additive smoothing，又叫拉普拉斯平滑(Laplace smoothing)

增加一个大于 0 的可调参数 alpha 进行平滑，就叫 **Lidstone** 平滑

2）决策边界

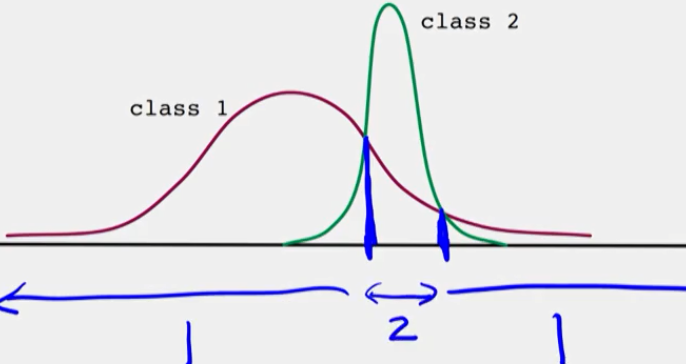
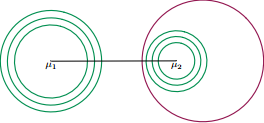
Binary classification with Gaussian generative model



 ,

上面解就是一个quadratic function + linear function，theta是决策边界threshold depending on the various parameters.

所以当两个class的矩阵相等时，决策边界为线性；不等时为椭圆

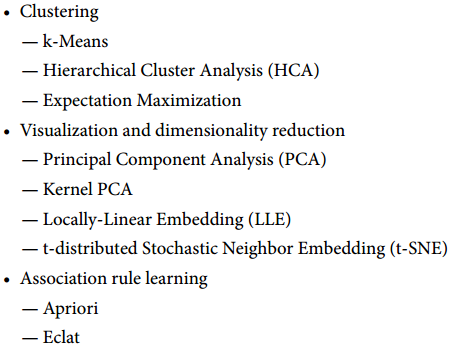


5.提升

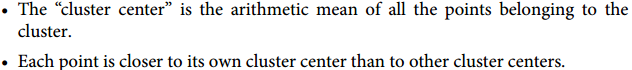
different distribution: GAMMA, BETA, POISSON, CATEGORICAL

Graphical models

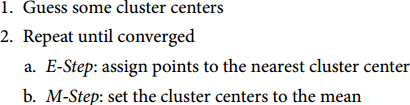
### 非监督学习



#### k-Means



Expectation–Maximization



**Hard Clustering *vs* Soft Clustering**

Rather than arbitrarily choosing the closest cluster for each instance, which is called hard clustering, it might be better measure the distance of each instance to all 5 centroids.

Caveats of expectation–maximization

The globally optimal result may not be achieved

kmeans.inertia\_

K-Means++: Most of the time, this largely compensates for the additional complexity of the initialization process.

*The number of clusters must be selected beforehand*

silhouette analysis

a more complicated clustering algorithm which has a better quantitative measure of the fitness per number of clusters or which can choose a suitable number of clusters (e.g., DBSCAN, mean-shift, or affinity propagation, all available in the sklearn.cluster submodule)

*k-means is limited to linear cluster boundaries*

SpectralClustering

*k-means can be slow for large numbers of samples*

batch-based *k*-means

Its performance is often lower (higher inertia), and it keeps degrading as k increases.

**Gaussian Mixtures**

如果某指标在各class下的分布重叠越少，该指标就能很好地提高准确率

positively correlated

independent means uncorrelated, but uncorrelated does mean independent

Cross-entropy is prefered for **classification**, while mean squared error is one of the best choices for **regression**.

in classification you work with very particular set of possible output values thus MSE is badly defined (as it does not have this kind of knowledge thus penalizes errors in incompatible way)

 logistic regression assumes binomial distribution (or multinomial in generalised case of cross entropy and softmax) of the dependent variable, while linear regression assumes that it is a linear function of the variables plus an IID sampled noise from a 0-mean gaussian noise with fixed variance.

 (insertion, deletions, substitution)is also metric(Hamming distance)

The **Kullback-Leibler divergence** or **relative entropy** between *p; q**is not metric*

### Problem Set 2

1. A generative approach is used for a binary classification problem and it turns out that the resulting classifier predicts + at all points x in the input space.

There are fewer − points than + points in the training set.

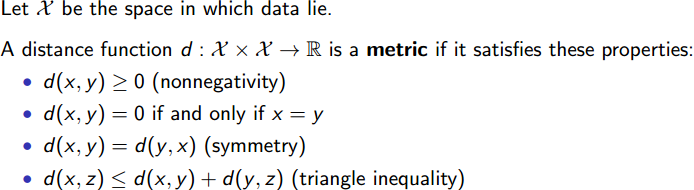
错：The density of + points is greater than the density of − points everywhere in the space.这样+的分布的积分会大于1

2. Roll a fair die. X is 1 if the outcome is even, and 0 otherwise. Y is 1 if the outcome is ≥3, and zero otherwise.是否独立

X和Y的概率相乘，用n//N求和事件并比较

## 数学基础：

1. Metric spaces



如果是metric，那么这个function d可以do **fast** nearest neighbour search

设m=1（特征数量）来看是否metric；d1和d2 metric，d1+d2也metric

all Lp distances are metric

2.

熵(entropy) 表示随机变量不确定性的度量，定义为



其中某事件X可能有n种结果，其对应的概率为

例如当X为Bernoulli分布，则p=0.5时，熵最大，即对两个结果的概率一无所知，两者都取0.5。

条件熵(conditional entropy)



#### 假设检验

(1)选择合适的检验

**两组检验类型**

参数检测： 假定数据遵从某些特定的分布（例如：高斯分布），对总体参数进行估计或检验。  
例如 ***:*** z 检测， *t* 检测， ANOVA， chi-square 等  
非参数检测： 并没有假定数据遵从某种分布。 往往直接对分布的某种特性（如对称性，分位数大小）做检验。  
例如***:*** Kolmogorov-Smirnov 检测， Wilcoxon 检测， Mann-Whitney 检测, Kruskal-Wallis 检测等。

**一个样本和多个样本**单个样本检验： 仅仅基于一个采样样本，通常基于均值、方差和分布的假设  
例如，正态分布检验， z 检验， t 检验  
多个样本检验： 目标是比较多个组别的均值方差是不是相等。  
例如： ANOVA 检验, Kruskal-Wallis 检验, Chi- square 检验等等。

(2)具体检验

***Q-Q plot***（图形检验）: 比较两个概率分布的相似程度。如果这两个分布很相似， Q-Q plot 上的点会近似地位于对角线 y = x 附近

**Kolmogorov-Smirnov**（非参数检验）：以样本数据的累计频数分布与特定理论分布比较，若两者间的差距很小，则推论该样本取自某特定分布。只对连续分布适用。

**Z** 检验



**T** 检验



(3)步骤

1) 根据问题，判定感兴趣的参数  
2) 给定原假设, *H*0  
3) 给定备择假设 *H*1  
4) 选择一个置信水平 α.  
5) 选择合适的假设检验  
6) 推导出拒绝域  
7) 计算需要的统计变量  
8) 决定拒绝或接收原假设 *H*0

**AB Test**



#### 探索性数据分析（ **EDA**）

1.数据类型

结构化数据： 二分、多分、有序类型

非结构化数据：文本音频、视频、图片

2.单变量分析

频数和众数：无序的分类的变量  
百分位数：有序的或连续的变量  
位置度量：均值和中位数  
散布度量：方差、标准差、偏度、峰度、四分位数极差

偏度：关于均值，右偏大于 0

峰度：分布状态的陡缓程度，与正太相比

四分位数极差： ，大于或小于都算异常

3.图

直方图：正态、双峰、平顶、偏态、孤岛。对称、分散、异常、间隙

箱线图

4.正态性检验

直方图、箱线图（若矩形位于中间位置且中位数位于矩形的中间位置）

算法效率：

