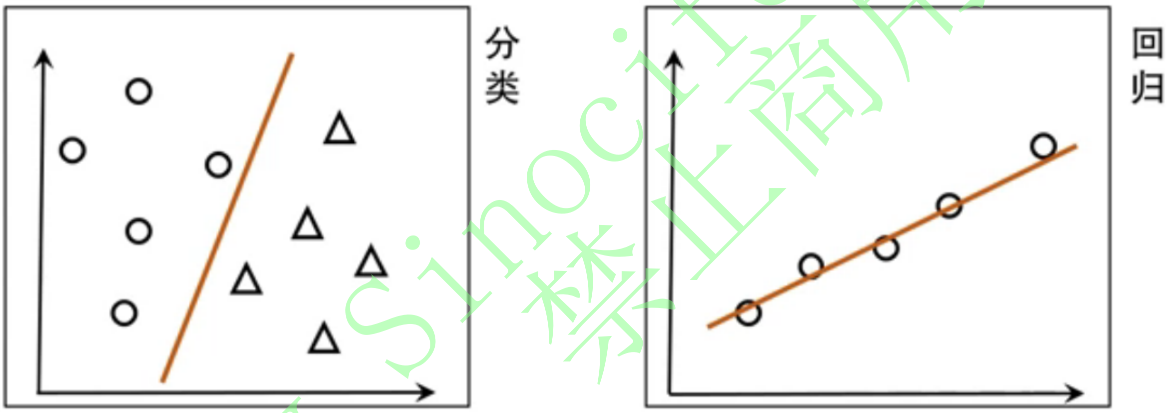
南大周志华《机器学习》课程笔记

> Introduction：最近自学机器学习课程，注意到了南京大学周志华老师的课程。我是在学堂在线平台观看的，注意到b站上也有相应视频，但b站上并未获得授权，随时有消失的可能。
>
> 周志华老师的网络教学视频中，与其西瓜书相比确实少了一些内容。但幸运的是，缺失的内容实际上对于初学者来说并不会产生太大影响。目前这一笔记也遵循视频内容，相比西瓜书中也会有一些缺失，敬请谅解。可能以后如果有机会和时间，我会再阅读周志华老师的书籍将缺失内容补全。
>
> 一切内容敬请关注我的个人Page页面。
>
> 全系列笔记请见：click here
> **About Me:**点击进入我的Personal Page

第三章 线性模型



线性模型(Linear model)试图学到一个通过属性的线性组合来进行预测的函数

$$f(x) = w_1x_1 + w_2x_2 + \ldots + w_dx_d + b$$

向量形式：$f(x) = w^T x + b$

简单、基本、可理解性好

线性回归

线性回归就是在寻找一个方差$f(x_i) = wx_i + b$ 使得 $f(x_i) \simeq y$

> 线性回归非常适合处理数值属性，如 $w_1 * 0.9 + w_2 * 0.8 = y$，但如果是类似 $w_1 * \text{青绿色} + w_2 * \text{声音浑浊} = y$这种该怎么处理？

因此，需要将离散数据转换为连续数据。但在转换过程中，需要考虑数据之间是否存在**"序"**（order）

> 例如将身高分为 高、中、低三类，其看似离散，但实际之间存在"序"关系 （(高=1.0)>(中=0.5)>(低=0)），但如果是西瓜颜色为[青绿色][浅白色][淡黄色]，这三个颜色之间无法找出这种"序"关系。
>
> 此时，我们无法将他们当成0、1来进行处理。换言之**不能看到离散数据就思考将其当作1和0**，因为实际运算时候的数值关系之间说明了一种距离远近的关系，但实际上可能并不存在这样一种关系。如果这么做，就人为错误地引入了这样一种序关系。

如果离散数据之间存在"序"，可以进行0-1处理；如果没有序，将这一属性变成k维的向量，k为属性值的个数。

> 如上面例子，西瓜颜色为[青绿色][浅白色][淡黄色]，我们可以将其分别表示为(1,0,0)、(0,1,0)和(0,0,1)

在机器学习算法中大概可以分为两大类，其中一类适合离散数据，另一类适合处理连续数据。如果要用离散算法处理连续信号，就先要将连续信号离散化；如果要用连续算法处理离散值，就先要将离散值连续化。但这两个过程，目前并未找到一个完美的解决方案。

最小二乘法

对于如何寻找线性回归，我们希望均方误差最小化，有

$$(w^*,b^*)=argmin_{w,b}\sum_{i=1}^{m}(f(x_i)-y_i)^2=argmin_{w,b}\sum_{i=1}^{m}(y_i-wx_i-b)^2$$

对$E_{w,b}=\sum_{i=1}^{m}(y_i-wx_i-b)^2$ 进行最小二乘参数估计【基于均方误差最小化来进行模型求解的方法称为 **最小二乘法** 】

> 最小二乘估计，即分别对w和b求偏导，令偏导数等于0.
>
> **分别对 w 和 b 求导：**
>
> $$\frac{\partial E_{(w,b)}}{\partial w}=2\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)\,x_i\right)$$
>
> $$\frac{\partial E_{(w,b)}}{\partial b}=2\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)$$
>
> **令导数为 0，得到闭式(closed-form)解：**
>
> $$w=\frac{\sum_{i=1}^{m}y_i\,(x_i-\bar{x})}{\sum_{i=1}^{m}x_i^2-\frac{1}{m}\left(\sum_{i=1}^{m}x_i\right)^2}\qquad b=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i)$$
>
> 偏导意味着变化率，当其为0，则希望找到这个值不再发生变化的点，即很有可能到极值点。由于均方误差可以无限大(偏离足够远)，因此这里通过偏导等于0解出来的必然为最小值，也就是均方误差最小，即为我们想要的最优解。

最小二乘法求得的斜率 【小于】最小化数据集到线性模型欧式距离的平方和求得的斜率。这一结论对一般问题也成立，可尝试证明之。

多元(Multi-variate)线性回归

多元，即多变量。$f(x_i)=w\mathbf{x}_i+b$ 使得 $f(\mathbf{x}_i)\simeq y$

$\mathbf{x}_i=(x_{i1},x_{i2},\ldots,x_{id})\quad y_i\in\mathbb{R}$，此时需要考虑的**x**有d个属性

> 思考：结果为$y=w\mathbf{x}_i+b$，可以看成$y=w\mathbf{x}_i+b\times 1$。展开即为$y=w_1x_1+w_2x_2+\ldots+w_dx_d+b\times 1$，可以视$\hat{x}=[x_1,x_2,\ldots,x_d,1],\hat{w}=[w_1,w_2,\ldots,w_d,b]$

因此，将w和b吸收入向量形式$\hat{w}=(w,b)$，数据表示为：

$$X=\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \dots & \dots & \dots & \dots & 1 \\ x_{m1} & x_{m2} & \dots & x_{md} & 1 \end{pmatrix}=\begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \dots & \dots \\ x_m^T & 1 \end{pmatrix} \qquad y=(y_1,y_2,\dots,y_m)$$

同样采用最小二乘法求解，有

$$\hat{\boldsymbol{w}}^*=\underset{\hat{\boldsymbol{w}}}{\arg\min}\,(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})$$

令 $E_{\hat{\boldsymbol{w}}}=(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})$，对$\hat{\boldsymbol{w}}$求导：

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}}=2\mathbf{X}^{\mathrm{T}}(\mathbf{X}\hat{\boldsymbol{w}}-\boldsymbol{y})$$ 令其为零可得 $\hat{\boldsymbol{w}}$

然而，麻烦来了：涉及矩阵求逆!

- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 满秩或正定，则$\hat{\boldsymbol{w}}^*=\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不满秩，则可解出多个$\hat{\boldsymbol{w}}$

此时需求助于归纳偏好，或引入 正则化 (regularization)

> 例如：$y_1=Ax_1+Bx_2+Cx_3+d$和 $y_2=A'x_1+B'x_2+C'x_3+d$。变成矩阵后发现其必然不满秩，存在多组参数值。此时的归纳偏好，如：要求参数C越小越好，或C越大越好。这样就可以得到一组确定的解。

广义线性模型

对于样例$(x,y),y\in\mathbb{R}$，希望线性模型的预测值逼近真实标记，则得到线性回归模型$y=w^Tx+b$。

可否令预测值不直接逼近y，而是逼近y的衍生物？

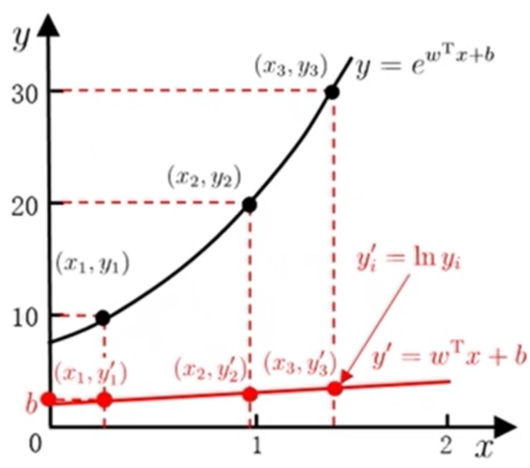
令预测值逼近 y 的衍生物?

若令 $\ln y=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$

则得到对数线性回归

(log-linear regression)

实际是在用 $e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}$ 逼近 y



一般形式：$y = g^{-1}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b\right)$

单调可微的 联系函数 (link function)

令 $g(\cdot) = \ln(\cdot)$ 则得到 对数线性回归

$$\ln y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

… …

对率回归（用回归做分类）

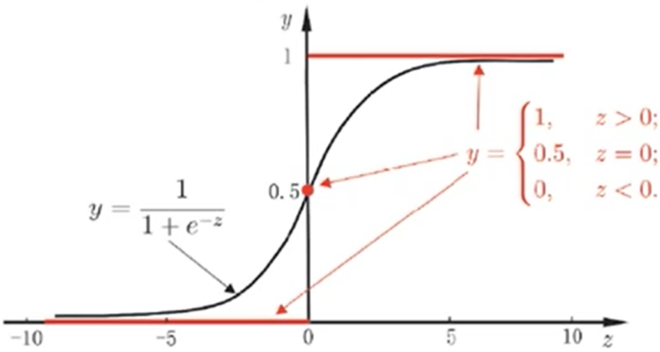
线性回归模型产生的实值输出 $z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

期望输出 $y \in \{0, 1\}$

找 z 和 y 的联系函数

理想的“单位阶跃函数”
(unit-step function)

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$



性质不好，
需找“替代函数”
(surrogate function)

常用
单调可微、任意阶可导

$$y = \frac{1}{1 + e^{-z}}$$

对数几率函数
(logistic function)
简称“对率函数”

> 注意：logistic与“逻辑”没有任何关系
>
> 1. logistic来自于Logit 而不是Logic
> 2. 实数值，并非“非0即1”的逻辑值

以对率函数为联系函数：

$$y=\frac{1}{1+e^{-z}} \quad 变为 \quad y=\frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$$

即：

$$\ln\frac{y}{1-y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$$

几率(odds)，反映了 $\boldsymbol{x}$ 作为正例的相对可能性

“对数几率”
(log odds，亦称 logit)

“对数几率回归”(logistic regression)
简称“对率回归”

> $\frac{y}{1-y}$即为$\frac{P(x=正例)}{P(x=负例)}$。

好处：

- 无需事先假设数据的分布
- 可以得到“类别”的近似概率预测
- 可以直接应用现有数据优化算法求取最优解

对率回归求解

若将 y 看作类后验概率估计 $p(y=1\mid\boldsymbol{x})$ ，则

$$\ln\frac{y}{1-y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \quad 可写为 \quad \ln\frac{p(y=1\mid\boldsymbol{x})}{p(y=0\mid\boldsymbol{x})}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$$

于是，可使用“极大似然法” → 第7章
(maximum likelihood method)

目前的理解：
如果要看起来正例就让P(y=1|x)变大
如果要看起来负例就让P(y=0|x)变大

给定数据集 $\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{m}$

最大化“对数似然”(log-likelihood)函数

$$\ell(\boldsymbol{w},b)=\sum_{i=1}^{m}\ln p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w},b)$$

?我们知道，目标函数为$f(x)=\frac{1}{1+e^{-(w^Tx+b)}}$，为什么不用线性回归那样采用均方误差进行求解呢，直接进行均方误差最小化，采用最小二乘法进行求解？

我们之前做的就是让$(f(x)-y)^2$最小，那么可以将目标改写为：求$(\frac{1}{1+e^{-(w^Tx+b)}}-y)^2$最小，即对w求导，令其等于0，得到导数为0的点，这种思路可行吗？

答案是**不行！**。什么时候可以通过求梯度为0的点来获取极值？（极值点梯度为0，但梯度为0未必为极值点）当函数本身为凸函数时候才能存在极值点。为什么线性回归可以使用均方误差最小化？因为一条线只存在两个极值点——极大(可以无穷大)和极小(极值点必然为极小)。而考虑上面的函数$(\frac{1}{1+e^{-(w^Tx+b)}}-y)^2$，其为非凸函数，因此无法使用均方误差进行求解。所以这里采用了极大似然法来进行求解。

💡**极大似然法**的基本思想(详细内容之后再仔细讲解)

极大似然法的思想，即`MAX{P(确实为正类)P(预测为正类)+P(确实为负类)P(预测为负类)}`，使预测成功的部分尽可能最大。通常要对其进行对数运算，原因是概率通常比较小，当把概率进行连乘可能会出现浮点数的下溢。取对数后，乘法就可以变成加法，可以缓解下溢问题。

> 令 $\boldsymbol{\beta}=(\boldsymbol{w};b)$，$\hat{\boldsymbol{x}}=(\boldsymbol{x};1)$，则 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 可简写为 $\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}$
>
> 再令 $p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=p(y=1\mid\hat{\boldsymbol{x}};\boldsymbol{\beta})=\dfrac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$
>
> $p_0(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=p(y=0\mid\hat{\boldsymbol{x}};\boldsymbol{\beta})=1-p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=\dfrac{1}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$
>
> 则似然项可重写为 $p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w}_i,b)=y_ip_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})+(1-y_i)p_0(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})$
>
> 于是，最大化似然函数 $\ell(\boldsymbol{w},b)=\sum_{i=1}^{m}\ln p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w},b)$
>
> 等价为最小化 $\ell(\boldsymbol{\beta})=\sum_{i=1}^{m}\left(-y_i\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i+\ln\left(1+e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i}\right)\right)$
>
> 高阶可导连续凸函数，可用经典的数值优化方法
> 如梯度下降法/牛顿法 [Boyd and Vandenberghe, 2004]
>
> P(确实为正类)就是y，P(预测为正类)为$p_1=\frac{e^{\beta^Tx}}{1+e^{\beta^Tx}}$；
>
> P(确实为负类)就是$1-y$，P(预测为负类)为$p_0=1-p_1=\frac{1}{1+e^{\beta^Tx}}$；

也就是说，极大似然法是要下式最大化：

$$y*\frac{e^{\beta^Tx}}{1+e^{\beta^Tx}}+(1-y)*\frac{1}{1+e^{\beta^Tx}}$$

对其化简，得到

$$\frac{ye^{\beta^Tx}+1-y}{1+e^{\beta^Tx}}$$

当然要对其进行实际应用，就需要取对数，并取最大值，即下式：

$$\begin{aligned}
&\max\left\{\ln\frac{ye^{\beta^Tx}+1-y}{1+e^{\beta^Tx}}\right\}\\
&=\max\left\{\ln\left(ye^{\beta^Tx}+1-y\right)-\ln\left(1+e^{\beta^Tx}\right)\right\}\\
&=\max\begin{cases}\beta^Tx-\ln\left(1+e^{\beta^Tx}\right),y=1\\0-\ln\left(1+e^{\beta^Tx}\right),y=0\end{cases}
\end{aligned}$$

可以将结果视为所要求的某个函数的两个极值点(y=1和y=0时)。如果现在要将极值点嵌入到一个函数中去，则可以得到下式（注意y=1时和y=0时，下式满足上面的分布函数）：

$$\max\left\{y*\beta^T x-\ln\left(1+e^{\beta^T x}\right)\right\}$$

等价于：

$$\min\left\{\ln\left(1+e^{\beta^T x}\right)-y*\beta^T x\right\}$$

这样，便得到了上面图片中最后的公式。为什么取了 i 从1至m，是因为取了m个样本，在每个样本上都要进行这样的运算。如此，我们便得到了我们的目标函数。

> 该目标函数的好处：
>
> 对该目标函数进行变形，可以得到
>
> $$\min\left\{\ln\left(1+e^{\beta^T x}\right)-\ln e^{(y*\beta^T x)}\right\}$$
> $$=\min\left\{\ln\frac{1+e^{\beta^T x}}{e^{(y*\beta^T x)}}\right\}$$
>
> $\beta^T x$ 实际上就是原本的线性回归$f(x)=w^T x+b$，那么上面的变形实际上就是下面的式子：
>
> $$\min\left\{\ln\frac{1+e^{f(x)}}{e^{yf(x)}}\right\}$$
>
> 而这一个函数性质非常好，其高阶连续可导。因此，很多求解技术就都可以在它上面使用。

得到目标后，最简单的方式是采用**梯度下降**，如下：

$$\frac{\partial \ln\frac{1+e^{f(x)}}{e^{yf(x)}}}{\partial w}$$

每次对w加上Δw 再往下走。由于函数高阶可微，也可以采用二阶法(即牛顿法)进行求解。

二阶的方法实际上是下降的方向再往下的冲量走的更快的方向，这样可能下降的更快。并非所有都可以用牛顿法，但梯度下降肯定是可以的。

❓ 我们在这里得到的最小化函数(记为$g(x)$)现在是凸函数了，那么是否可以使用这一函数采用$min\{g(x)-y\}^2$来做？

答案是通常是不可以的。在线性回归$y=\beta^T x$中，y和x均为标量或至少x的维数较低才行。更一般地，我们最后得到的w为$w=(X^T X)^{-1}X^T y$ <注：在多元线性回归中有介绍，其中y也是矢量>。当x并非一维而是很多维时，这样求解w时$(X^T X)^{-1}$ 求解是有问题的。现实生活中通常这个逆很难求甚至不存在。

因此，通过求解$min\{g(x)-y\}^2$，即使得其导数等于0，就等价于求解$w=(X^T X)^{-1}X^T y$。但对于线性回归来说，解决这一问题通常是无法通过解方程可以解决的。所以，即使是凸函数，现实生活中也很少通过对其求导使得导数等于0来直接进行求解。直接求导，令导数等于0来获得最优解是非常少的模型才能做的，这是非常理想化的条件了。

因此，即使是线性回归，我们可能都要通过梯度下降来求解。梯度下降的方法基本上是放之四海而皆准的，因为函数的最终解一定是梯度为0的点（但梯度为0的点不一定是最终解），这是**原因一**。

而有时候，即使可以进行求解，实际中还是喜欢使用梯度下降的方法。这是由于梯度下降过程中$w_n=w_{n-1}+\Delta w$，即本轮的w值是上一轮做出了一点微小的变化。可以看到，这是一种迭代的解法，而迭代的解法容易并行化，比较适合计算机进行求解。因此就算$w=(X^T X)^{-1}X^T y$这一方差可以求解，但是由于方程很大，为了更好利用计算机的优势，往往解出来会更快，这是**原因二**。

类别不平衡(class-imbalance)

不同类别的样本之间比例相差很大，“小类”往往更加重要，例如检测信用卡检测，更加关注异常数据。

基本思想：若几率$\frac{y}{1-y} > 1$，即$y > 1-y$, 等价于$y > \frac{1}{2}$，则预测为正类。然而这是正类、负类同等重要，且双方数量相差不大时。但假如正类更加重要，且正类样本数比较少呢?

对此，我们直觉上认为对几率的范围应该做出限制，即$\frac{y}{1-y} > \frac{m^+}{m^-}$，则预测为正类($m^+, m^-$分别为样本中正类、负类数量)。

对此，提出我们的基本策略：

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

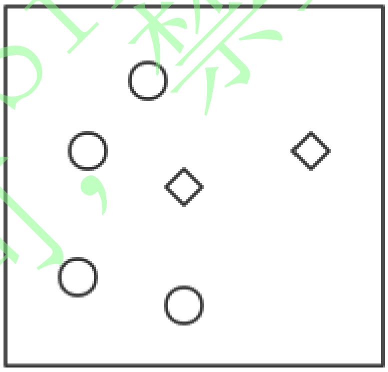
但是，精确估计m^-/m^+是困难的，因为如果我们在训练集中这么或者该比值，就引入了一个潜在的假设——拿到的训练集是整个数据集中的一个无偏采样。

🎖常见类别不平衡学习方法：

- 过采样(oversamping)：如SMOTE——小类增加，增加到和大类一样多，按原本方法做
- 欠采样(undersamping)：如EasyEnsemble——大类变小，让其与小类一样小
- 阈值移动(thresold-moving)：——很少的算法支持上述的移动阈值方法，如SVM

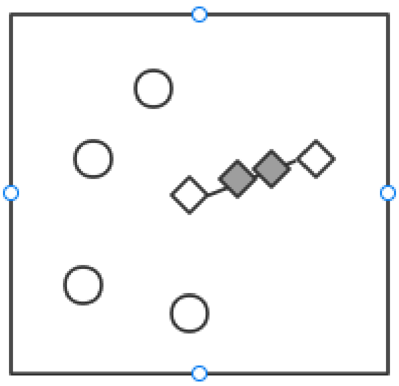
关于过采样

假如数据空间中，两类数据分布如下，假如要使用过采样方式应该如何处理?



> 最简单的想法是copy，将小类数据copy多份，从而保障两类样本数据量相同。然而，简单的copy会带来很大的问题，见到copy会使得overfiting大幅度增加。因为如果小样本中存在噪声，copy的方式会使得噪声的影响加倍。

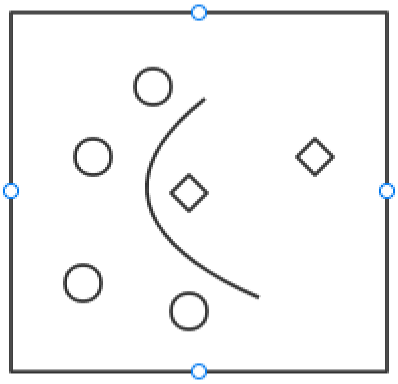
对此，最著名的一种方法是nitech chawla提出的SMOTE方法。其思想为任意的pair之间插入值，如下图中灰色数据就是在原有数据之间插入的值。这样就可以得到很好的结果，这也是被广泛认可的一种方法。

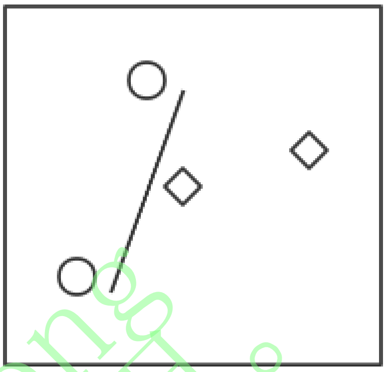


关于欠采样

欠采样是要丢弃一些大类样本。那么可以随机丢弃一些样本吗？实际上早期有人就是这么做的，然而实际上会带来很大问题，因为不知道是否会丢弃一些关键样本。

例如下图，左侧的分类边界为原本的结果，丢弃关键样本后的分类边界就变成右图了，可见产生了非常显著的变化。





所以，随机丢弃数据是不太好的，因为不知道是否丢弃了关键的大类样本。有一个很重要的做法为周志华团队提出的EasyEnsemble，该方法为集成学习模型。

以上面数据为例，可知小类数据有2个，大类数据4个。每次从大类中找2个数据做一个模型，共C_4^2次，每一个子模型都是均衡的。但通过做很多个模型后，最后采用投票等方式将其结合起来，其中每一个都是均衡的，从而避免丢弃大类中关键数据。此外，该方法还利用了集成学习的好处，从而使得模型精度更高。