支持向量机通俗导论（理解 SVM 的三层境界）

作者：July、pluskid ；致谢：白石、JerryLead
出处：结构之法算法之道 blog。

前言

动笔写这个支持向量机 (support vector machine) 是费了不少劲和困难的，原因很简单，一者这个东西本身就并不好懂，要深入学习和研究下去需花费不少时间和精力，二者这个东西也不好讲清楚，尽管网上已经有朋友写得不错了 (见文末参考链接)，但在描述数学公式的时候还是显得不够。得益于同学白石的数学证明，我还是想尝试写一下，希望本文在兼顾通俗易懂的基础上，真真正正能足以成为一篇完整概括和介绍支持向量机的导论性的文章。

本文在写的过程中，参考了不少资料，包括《支持向量机导论》、《统计学习方法》及网友 pluskid 的支持向量机系列等等，于此，还是一篇学习笔记，只是加入了自己的理解和总结，有任何不妥之处，还望海涵。全文宏观上整体认识支持向量机的概念和用处，微观上深究部分定理的来龙去脉，证明及原理细节，力保逻辑清晰 & 通俗易懂。

同时，阅读本文时建议大家尽量使用 chrome 等浏览器，如此公式才能更好的显示，再者，阅读时可拿张纸和笔出来，把本文所有定理. 公式都亲自推导一遍或者直接打印下来（可直接打印网页版或本文文末附的 PDF，享受随时随地思考、演算的极致快感），在文稿上演算。

Ok，还是那句原话，有任何问题，欢迎任何人随时不吝指正 & 赐教，感谢。

1 第一层、了解 SVM

1.1 什么是支持向量机 SVM

要明白什么是 SVM，便得从分类说起。

分类作为数据挖掘领域中一项非常重要的任务，它的目的是学会一个分类函数或分类模型 (或者叫做分类器)，而支持向量机本身便是一种监督式学习的方法 (至于具体什么是监督学习与非监督学习，请参见此系列 Machine L&Data Mining 第一篇)，它广泛的应用于统计分类以及回归分析中。

支持向量机（SVM）是 90 年代中期发展起来的基于统计学习理论的一种机器学习方法，通过寻求结构化风险最小来提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。

通俗来讲，它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

对于不想深究 SVM 原理的同学或比如就只想看看 SVM 是干嘛的，那么，了解到这里便足够了，不需上层。而对于那些喜欢深入研究一个东西的同学，甚至究其本质的，咱们则还有很长的一段路要走，万里长征，咱们开始迈第一步吧，相信你能走完。

1.2 线性分类

OK，在讲 SVM 之前，咱们必须先弄清楚一个概念：线性分类器 (也可以叫做感知机，这里的机表示的是一种算法，本文第三部分、证明 SVM 中会详细阐述)。

1.2.1 分类标准

这里我们考虑的是一个两类的分类问题，数据点用 x 来表示，这是一个 n 维向量，w^T 中的 T 代表转置，而类别用 y 来表示，可以取 1 或者 -1 ，分别代表两个不同的类。一个线性分类器的学习目标就是要在 n 维的数据空间中找到一个分类超平面，其方程

可以表示为：

$$w^T x + b = 0 \tag{1}$$

上面给出了线性分类的定义描述，但或许读者没有想过：为何用 y 取 1 或者 -1 来表示两个不同的类别呢？其实，这个 1 或 -1 的分类标准起源于 logistic 回归，为了完整和过渡的自然性，咱们就再来看看这个 logistic 回归。

1.2.2 1 或 -1 分类标准的起源：logistic 回归

Logistic 回归目的是从特征学习出一个 0/1 分类模型，而这个模型是将特性的线性组合作为自变量，由于自变量的取值范围是负无穷到正无穷。因此，使用 logistic 函数（或称作 sigmoid 函数）将自变量映射到 $(0,1)$ 上，映射后的值被认为是属于 y=1 的概率。

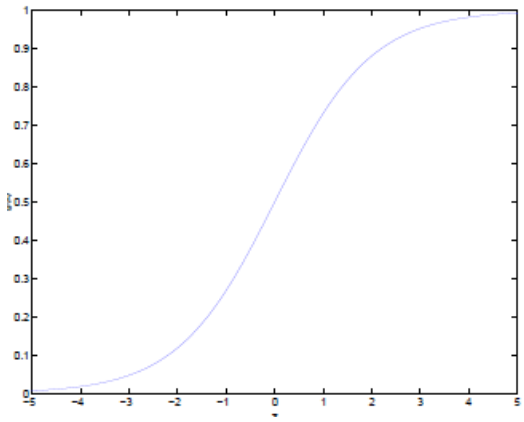
形式化表示就是

假设函数

$$h_0(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \tag{2}$$

其中 x 是 n 维特征向量，函数 g 就是 logistic 函数。

而 $g(z) = \frac{1}{1+e^{-z}}$ 的图像是



可以看到，将无穷映射到了 $(0,1)$。

而假设函数就是特征属于 $y=1$ 的概率。

$$P(y=1|x;\theta) = h_\theta(x) P(y=0|x;\theta) = 1 - h_\theta(x) \tag{3}$$

当我们要判别一个新来的特征属于哪个类时，只需求，若大于 0.5 就是 $y=1$ 的类，反之属于 $y=0$ 类。

再审视一下 $h_\theta(x)$，发现 $h_\theta(x)$ 只和 θ^T 有关，$\theta^T x > 0$，那么 $h_\theta(x) > 0.5$，$g(z)$ 只不过是用来映射，真实的类别决定权还在 $\theta^T x$。还有当时 $\theta^T x >> 0$，$h_\theta(x) = 1$，反之 $h_\theta(x) = 0$。如果我们只从 $\theta^T x$ 出发，希望模型达到的目标无非就是让训练数据中 $y=1$ 的特征

$$\theta^T x >> 0$$

，而是 $y=0$ 的特征 $\theta^T x << 0$。Logistic 回归就是要学习得到 θ，使得正例的特征远大于 0，负例的特征远小于 0，强调在全部训练实例上达到这个目标。

1.2.3 形式化标示

我们这次使用的结果标签是 $y=-1,y=1$，替换在 logistic 回归中使用的 $y=0$ 和 $y=1$。同时将 θ 替换成 w 和 b。以前的 $\theta^T x=\theta_0+\theta_1 x_1+\theta_2 x_2+\cdots+\theta_n x_n$，其中认为 $x_0=1$。现在我们替换为 b，后面替换 $\theta_1 x_1+\theta_2 x_2+\cdots+\theta_n x_n$ 为 $w_1 x_1+w_2 x_2+\cdots+w_n x_n$（即 $x^T x$）。这样，我们让 $\theta^T x=w^T x+b$，进一步 $h_\theta(x)=g(\theta^T x)=g(w^T x+b)$。也就是说除了 y 由 $y=0$ 变为 $y=-1$，只是标记不同外，与 logistic 回归的形式化表示没区别。

再明确下假设函数

$$h_{w,b}=g(w^T x+b) \tag{4}$$

上面提到过我们只需考虑 $\theta^T x$ 的正负问题，而不用关心 $g(z)$，因此我们这里将 $g(z)$ 做一个简化，将其简单映射到 $y=-1$ 和 $y=1$ 上。映射关系如下：

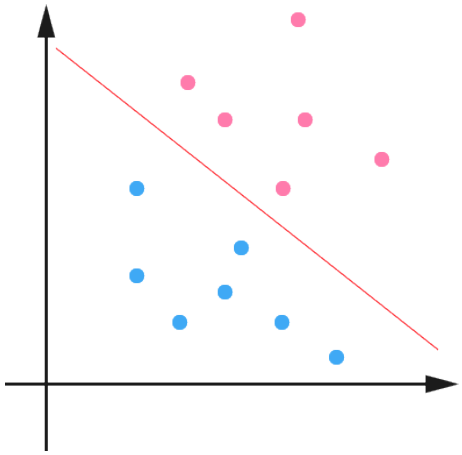
$$g(z)=\begin{cases}1, & z\geqslant 0\\ -1, & z<0\end{cases} \tag{5}$$

于此，想必已经解释明白了为何线性分类的标准一般用 1 或者 -1 来标示。

注：上小节来自 jerrylead 所作的斯坦福机器学习课程的笔记。

1.3 线性分类的一个例子

下面举个简单的例子，一个二维平面 (一个超平面，在二维空间中的例子就是一条直线)，如下图所示，平面上有两种不同的点，分别用两种不同的颜色表示，一种为红颜色的点，另一种则为蓝颜色的点，红颜色的线表示一个可行的超平面。

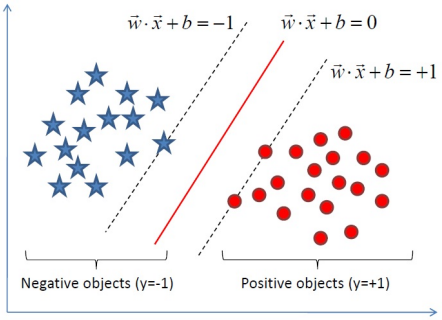


从上图中我们可以看出，这条红颜色的线把红颜色的点和蓝颜色的点分开来了。而这条红颜色的线就是我们上面所说的超平面，也就是说，这个所谓的超平面的的确确便把这两种不同颜色的数据点分隔开来，在超平面一边的数据点所对应的 y 全是 -1，而在另一边全是 1。

接着，我们可以令分类函数（提醒：下文很大篇幅都在讨论着这个分类函数）：

$$f(x)=w^T x+b \tag{6}$$

显然，如果 $f(x)=0$，那么 x 是位于超平面上的点。我们不妨要求对于所有满足 $f(x)<0$ 的点，其对应的 y 等于 -1，而 $f(x)>0$ 则对应 $y=1$ 的数据点。



注：上图中，定义特征到结果的输出函数 $u=\vec{w}\cdot\vec{x}-b$，与我们之前定义的 $f(x)=w^Tx+b$ 实质是一样的。为什么？因为无论是 $u=\vec{w}\cdot\vec{x}-b$，还是 $f(x)=w^Tx+b$，不影响最终优化结果。下文你将看到，当我们转化到优化 $\max\frac{1}{\|w\|}, s.t., y_i(w^Tx_i+b)\geq 1, i=1,\cdots,n$ 的时候，为了求解方便，会把 $yf(x)$ 令为 1，即 $yf(x)$ 是 $y(w^x+b)$，还是 $y(w^x-b)$，对我们要优化的式子 $\max\frac{1}{\|w\|}$ 已无影响。

（有一朋友飞狗来自 Mare Desiderii，看了上面的定义之后，问道：请教一下 SVM functional margin 为 $\widehat{\gamma}=y(wTx+b)=yf(x)$ 中的 γ 是只取 1 和 -1 吗？y 的唯一作用就是确保 functional margin 的非负性？真是这样的么？当然不是，详情请见本文评论下第 43 楼）

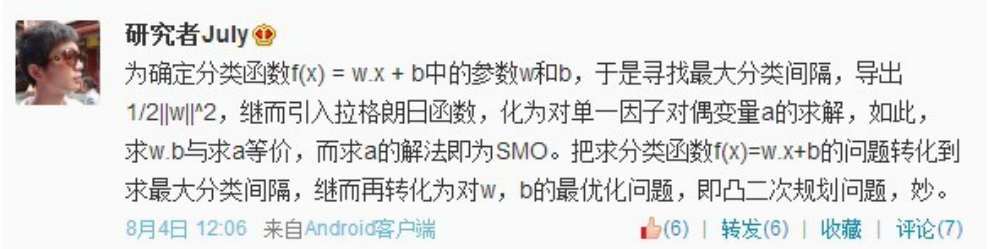
当然，有些时候，或者说大部分时候数据并不是线性可分的，这个时候满足这样条件的超平面就根本不存在 (不过关于如何处理这样的问题我们后面会讲)，这里先从最简单的情形开始推导，就假设数据都是线性可分的，亦即这样的超平面是存在的。

更进一步，我们在进行分类的时候，将数据点 x 代入 $f(x)$ 中，如果得到的结果小于 0，则赋予其类别 -1，如果大于 0 则赋予类别 1。如果 $f(x)=0$，则很难办了，分到哪一类都不是。

请读者注意，下面的篇幅将按下述 3 点走：

1. 咱们就要确定上述分类函数 $f(x)=w\cdot x+b$（$w\cdot x$ 表示 w 与 x 的内积）中的两个参数 w 和 b，通俗理解的话 w 是法向量，b 是截距（再次说明：定义特征到结果的输出函数 $u=|w\cdot|x-b$，与我们最开始定义的 $f(x)=w^Tx+b$ 实质是一样的）；
2. 那如何确定 w 和 b 呢？答案是寻找两条边界端或极端划分直线中间的最大间隔（之所以要寻最大间隔是为了能更好的划分不同类的点，下文你将看到：为寻最大间隔，导出 $1/2||w||^2$!!!!!!!!!!!!!!!!!!!!，继而引入拉格朗日函数和对偶变量 a，化为对单一因数对偶变量 a 的求解，当然，这是后话），从而确定最终的最大间隔分类超平面 hyper plane 和分类函数；
3. 进而把寻求分类函数 $f(x)=w\cdot x+b$ 的问题转化为对 w，b 的最优化问题，最终化为对偶因子的求解。

总结成一句话即是：从最大间隔出发（目的本就是为了确定法向量 w），转化为求对变量 w 和 b 的凸二次规划问题。亦或如下图所示（有点需要注意，如读者 @ 酱爆小八爪所说：从最大分类间隔开始，就一直是凸优化问题）：



1.4 函数间隔 Functional margin 与几何间隔 Geometrical margin

一般而言，一个点距离超平面的远近可以表示为分类预测的确信或准确程度。

- 在超平面 $w*x+b=0$ 确定的情况下，$|w*x+b|$ 能够相对的表示点 x 到距离超平面的远近，而 $w*x+b$ 的符号与类标记 y 的符号是否一致表示分类是否正确，所以，可以用量 $y*(w*x+b)$ 的正负性来判定或表示分类的正确性和确信度。

于此，我们便引出了定义样本到分类间隔距离的函数间隔 functional margin 的概念。

1.4.1 函数间隔 Functional margin

我们定义函数间隔 functional margin 为：

$$\widehat{\gamma} = y(w^T x + b) = yf(x) \tag{7}$$

接着，我们定义超平面 (w,b) 关于训练数据集 T 的函数间隔为超平面 (w,b) 关于 T 中所有样本点 (x_i, y_i) 的函数间隔最小值，其中，x 是特征，y 是结果标签，i 表示第 i 个样本，有：

$$\widehat{\gamma} = \min \widehat{\gamma}_i (i = 1, \cdots, n) \tag{8}$$

然与此同时，问题就出来了。上述定义的函数间隔虽然可以表示分类预测的正确性和确信度，但在选择分类超平面时，只有函数间隔还远远不够，因为如果成比例的改变 w 和 b，如将他们改变为 $2w$ 和 $2b$，虽然此时超平面没有改变，但函数间隔的值 $f(x)$ 却变成了原来的 2 倍。

其实，我们可以对法向量 w 加些约束条件，使其表面上看起来规范化，如此，我们很快又将引出真正定义点到超平面的距离 --几何间隔 geometrical margin 的概念（很快你将看到，几何间隔就是函数间隔除以个 $\|w\|$，即 $yf(x)/\|w\|$）。

数字与普通运算符号可直接由键盘上键入。下列符号可以直接由键盘键入：

$$+ \ - \ = \ < \ > \ / \ : \ ! \ | \ [\ ] \ (\ )$$

要注意的是，左右大括号 { } 在 LaTeX 中有特殊用途。欲排版左大括号，指令为 \{ ，右大括号之指令为 \} 。排版展示数式有以下四种方法可以达到目的:

\begin{equation} ... \end{equation}
\begin{displaymath} ... \end{displaymath}
\[... \]
$$ ... $$

除第一种方式外，其余将不对数学式子进行编号。数式内若要排版文字时，必须置于 \mbox 指令内，否则将被视为数学符号，譬如，

$$f(x) = x^2 - 3x + 1, \text{where} -2 \leq x \leq 2$$

2 常见的数学式

本节列举一些常见的数学式作为练习与未来使用的参考，每个函数都有其特别之处，请仔细观察研究。读者可以依此为基础，在往后的写作过程中，逐渐累积更多有特殊型态的或符号的数学式，只要这里出现过的，参照原使档一定写得出来。

2.1 函数

CODE 1: Binomial

```
$f(x)={n\choose x}p^x(1-p)^{1-x}, \;\; x=0,1,2,\cdots,n$
```

$f(x) = \binom{n}{x} p^x (1-p)^{1-x}, \;\; x = 0, 1, 2, \cdots, n$

CODE 2: Poisson

```
$f(x)=\frac{e^{-\lambda}\lambda^x}{x!}, \;\; x=0,1,2,\cdots$
```

$f(x) = \frac{e^{-\lambda}\lambda^x}{x!}, \;\; x = 0, 1, 2, \cdots$

CODE 3: Gamma

```
$f(x)=\frac{1}{\Gamma(\alpha)\beta^\alpha}x^{\alpha-1}
    e^{-\frac{x}{\beta}}, \;\; x\geq 0$
```

$f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}}, \;\; x \geq 0$

CODE 4: Normal

```
$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}},
    \;\;  -\infty < x < \infty $
```

$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \;\; -\infty < x < \infty$

CODE 5: 积分式与方程式编号

```
\begin{equation}\label{gamma}%.....后的名称自订，代表该方程式label
\int^\infty_0 x^{\alpha-1}e^{-\lambda x} dx =
    \frac{\Gamma(\alpha)}{\lambda^\alpha}
\end{equation}
```

$$\int_0^\infty x^{\alpha-1} e^{-\lambda x} dx = \frac{\Gamma(\alpha)}{\lambda^\alpha} \tag{9}$$

方程式 (9) 是广义 Γ 积分。[1]

CODE 6: 开根号

```
$$f(x)=\sqrt[3]{\frac {\displaystyle 4-x^{3}}{\displaystyle 1+x^{2}}}$$
```

$$f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$$

CODE 7: 微分与极限（注意大刮号的使用）

```
$$f'(x)=\frac{df(x)}{dx}=\lim_{h\rightarrow 0} \left(
    \frac{f(x+h)-f(x)}{h} \right)$$
```

[1]这里利用方程式标签（label）来引用方程式，编号将自动更新。

$$f'(x) = \frac{df(x)}{dx} = \lim_{h \to 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$

CODE 8: 上下限的使用

```
$$\int_a^b f(x) dx \approx \lim_{n\rightarrow \infty }\sum_{k=1}^n f(x_k)\triangle x_k$$
```

$$\int_a^b f(x)dx \approx \lim_{n\rightarrow \infty }\sum_{k=1}^n f(x_k)\triangle x_k$$

CODE 9: 最佳化问题

```
$$\max_{\mathbf{u} ,\mathbf{u}^T\mathbf{u}=1} \mathbf{u}^T\Sigma_X\mathbf{u}$$
```

$$\max_{\mathrm{u},\mathrm{u}^T\mathrm{u}=1} \mathrm{u}^T\Sigma_X\mathrm{u}$$

CODE 10: 几个符号

```
$$\mathbf{e}=\mathbf{x} -\mathbf{x}_q=(I -P)\mathbf{x} \in V^{\perp}, \mbox{where}\; V\oplus V^{\perp}=R^p $$
```

$$\mathrm{e} = \mathrm{x} - \mathrm{x}_q = (I - P)\mathrm{x} \in V^{\perp}, \text{where } V \oplus V^{\perp} = R^p$$

2.2 矩阵与行列式

矩阵或有规则排列的数学式或组合很常见，以下列举几种模式，请特别注意其使用的标签及一些需要注意的小地方。譬如，

1. 矩阵的左右括号需各别加上。
2. 横行各项之间是以 & 区隔。
3. 除最后一行外，每行之末则加上换行指令 \\。
4. 使用 array 指令时, 须加上选项以控制每一直栏内各数字或符号要居中排列、靠左或靠右。

范例与注意事项：

1. 左右方框刮号的使用及各直栏的对齐方式：

$$A = \left[\begin{array}{clr} a+b & mnop & xy \\ a+b & pn & yz \\ b+c & mp & xyz \end{array} \right]$$

```
$$ A = \left[
\begin{array}{clr}
    a+b & mnop  & xy  \\
    a+b & pn    & yz  \\
    b+c & mp    & xyz
\end{array} \right] $$
```

2. 左右圆框刮号的使用及各式点状：

$$A=\left(\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array}\right)$$

```
$$ A=\left(
\begin{array}{cccc}
    a_{11} & a_{12} & \cdots & a_{1n}\\
    a_{21} & a_{22} & \cdots & a_{2n}\\
    \vdots & \vdots & \ddots & \vdots\\
    a_{n1} & a_{n2} & \cdots & a_{nn}
\end{array} \right) $$
```

3. 排列整齐的符号：

$$\begin{array}{clr} a+b+c & m+n & xy \\ a+b & p+n & yz \\ b+c & m-n & xz \end{array}$$

```
$$ \begin{array}{clr}\\
    a+b+c    & m+n & xy  \\
    a+b      & p+n & yz  \\
    b+c      & m-n & xz
\end{array} $$
```

4. 等号对齐的函数组合（不编号）

$$\begin{aligned} b_1 &= d_1+c_1 \\ a_2 &= c_2+e_2 \end{aligned}$$

```
\begin{eqnarray*}
  b_1 &=& d_1+c_1  \\
  a_2 &=& c_2+e_2
\end{eqnarray*}
```

5. 等号对齐的函数组合（编号在最后一行）

$$\begin{aligned} b_1 &= d_1+c_1 \\ a_2 &= c_2+e_2 \end{aligned} \tag{10}$$

```
\begin{eqnarray}
    \nonumber b_1 &=& d_1+c_1  \\
    a_2 &=& c_2+e_2
\end{eqnarray}
```

6. 使用巨集 amsmath 的指令 align（控制编号在第一行）

$$\begin{aligned} b_1 &= d_1 + c_1 \\ a_2 &= c_2 + e_2 \end{aligned} \tag{11}$$

```
\begin{align}
    b_1 &= d_1+c_1\\
    a_2 &= c_2+e_2 \notag
\end{align}
```

7. 两组数学式分别对齐

$$\alpha_1 = \beta_1 + \gamma_1 + \delta_1, \qquad a_1 = b_1 + c_1 \tag{12}$$

$$\alpha_2 = \beta_2 + \gamma_2 + \delta_2, \qquad a_2 = b_2 + c_2 \tag{13}$$

```
\begin{align}
    \alpha_1 &= \beta_1+\gamma_1+\delta_1, &a_1 &=
       b_1+c_1\\
    \alpha_2 &= \beta_2+\gamma_2+\delta_2, &a_2 &=
       b_2+c_2
\end{align}
```

8. 编号在中间（split 指令环境）

$$\begin{aligned} \alpha_1 &= \beta_1 + \gamma_1 \\ \alpha_2 &= \beta_2 + \gamma_2 \end{aligned} \tag{14}$$

```
\begin{equation}
    \begin{split}
        \alpha_1 &= \beta_1+\gamma_1\\
        \alpha_2 &= \beta_2+\gamma_2
    \end{split}
\end{equation}
```

9. 只是居中对齐的数学式组（gather 指令环境）

$$\alpha_1 + \beta_1$$

$$\alpha_2 + \beta_2 + \gamma_2$$

```
\begin{gather}
    \alpha_1 + \beta_1\notag\\
    \alpha_2 + \beta_2 + \gamma_2\notag
\end{gather}
```

10. 长数学式的表达（注意第二行加号的位置）

$$\begin{aligned} y &= x_1 + x_2 + x_3 \\ &\quad + x_4 + x_5 \end{aligned} \tag{15}$$

```
\begin{align}
    y    &= x_1 + x_2 + x_3 \notag\\
         &\quad + x_4 + x_5
\end{align}
```

2.3 其他

$$X_n \xrightarrow{d} X$$

```
$$X_{n} \stackrel{d}{\longrightarrow} X$$
```

$$\overbrace{X_1 + \ldots + \underbrace{X_{15} + \ldots + X_{30}}}$$

```
$$\overbrace{X_{1} + \ldots + \underbrace{X_{15} + \ldots +
  X_{30}}}$$
```

$$G = \begin{cases} CLASS\#1 \text{ if } \hat{\beta}^T \mathrm{x} \leq 0 \\ CLASS\#2 \text{ if } \hat{\beta}^T \mathrm{x} > 0 \end{cases}$$

```
\begin{equation*}
  G = \left\{\begin{array}{l}
        CLASS\#1 \;\;\mbox{if} \;\; \hat{\beta}^T\bf{x} \leq
           0 \\
        CLASS\#2 \;\;\mbox{if} \;\; \hat{\beta}^T\bf{x} > 0
      \end{array}\right.
\end{equation*}
```

以 equation 或 align 排版时，数学式会自动编上号码。文稿其他地方若要引述某数学式，可先以 \label 指令加上标签，再使用 \ref 指令引述。如此一来若排版文稿须反覆修改，使用 \label 与 \ref 指令可以「自动对焦」不会出错。