机器学习（西瓜书）

注　解

（第 11 章　特征选择与稀疏学习）

https://blog.csdn.net/jbb0523

前言

经常听人说南大周老师所著的《机器学习》（以下统称为西瓜书）是一本入门教材，是一本科普性质的教科书。在该书第十次印刷之际，周老师在“如何使用本书”中也提到“这是一本入门级教科书”。然而，本人读起来却感觉该书远不止“科普”“入门”那么简单，书中的很多公式需要思考良久方能推导，很多概念需要反复咀嚼才能消化。边读边想着是不是应该将自己学习时遇到的一些知识难点的解析分享出来，以帮助更多的人入门。自己的确也随手做过一些笔记，但由于怀疑这仅是自己的个别现象，毕竟读书期间，思考更多的是如何使用单片机、DSP、ARM、FGPA 等，而这些基本是不需要推导任何公式的，因此作罢。偶然间在周老师的新浪微博看到如下对话：

此时方知，可能“读不懂”并不是个别现象。因此决定写一本“西瓜书注解”或者称为“西瓜书读书笔记”，对自己研读西瓜书时遇到的“台阶”进行解释和推导，以帮助更多的人能够更快地进入到这个领域。另外，近期越来越强地意识到，扎扎实实地推导一些基础算法的公式，无论是对于理解算法本身机理还是进行学术研究，都是非常有必要的。

自己会根据个人学习进度和研究需要按章发布，不知道能不能坚持写完，加油！

毕竟自己也是一名初学者，所以可能一些概念解释并不完整、一些公式推导并不优美，甚至会存在错误，这是不可避免的，不接受漫骂，但欢迎将问题反馈给我，共同学习进步！

（网盘链接：https://pan.baidu.com/s/1QtEiNnk8jMzmbs0KPBN-_w）

第 11 章目录

第 11 章 特征选择与稀疏学习

11.1 子集搜索与评价

本节内容简单易懂，几乎不需要什么注解。

开篇给出了“特征选择”的概念，并谈到特征选择与第 10 章的降维有相似的动机。特征选择与降维的区别在于特征选择是从所有特征中简单地选出相关特征，选择出来的特征就是原来的特征；降维则对原来的特征进行了映射变换，降维后的特征均不再是原来的特征。

本节涉及“子集评价”的式(14.1)和式(14.2)与第 4 章的式(4.2)和式(4.1)相同，这是因为“决策树算法在构建树的同时也可看作进行了特征选择”（参见“11.7 阅读材料”）。

接下来在 11.2 节、11.3 节、11.4 节分别介绍的三类特征选择方法：过滤式(filter)、包裹式(wrapper)和嵌入式(embedding)。

11.2 过滤式选择

“过滤式方法先对数据集进行特征选择，然后再训练学习器，特征选择过程与后续学习器无关。这相当于先用特征选择过程对初始特征进行‘过滤’，再用过滤后的特征来训练模型。”，这是本节开篇第一段原话，之所以重写于此，是因为这段话里包含了“**过滤**”的概念，该概念并非仅针对特征选择，那些所有先对数据集进行某些预处理，然后基于预处理结果再训练学习器的方法（**预处理过程独立于训练学习器过程**）均可以称之为“过滤式算法”。特别地，本节介绍的 Relief 方法只是过滤式特征选择方法的其中一种而已。

本节内容简单易懂，几乎不需要什么注解。

从式(11.3)可以看出，Relief 方法本质上基于“空间上相近的样本具有相近的类别标记”假设。Relief 基于样本与同类和异类的最近邻之间的距离来计算相关统计量δ^j，越是满足前提假设，原则上样本与同类最近邻之间的距离$\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2$会越小，样本与异类最近邻之间的距离$\mathrm{diff}(x_i^j, x_{i,\mathrm{nm}}^j)^2$会越大，因此相关统计量$\delta^j$越大，对应属性的分类能力就越强。

对于能处理多分类问题的扩展变量 Relief-F，由于有多个异类，因此对所有异类最近邻进行加权平均，各异类的权重为其在数据集中所占的比例。

11.3 包裹式选择

“与过滤式特征选择不考虑后续学习器不同，包裹式特征选择直接把最终将要使用的学习器的性能作为特征子集的评价准则。换言之，包裹式特征选择的目的就是为给定学习器选择最有利于其性能、‘量身定做’的特征子集。”，这是本节开篇第一段原话，之所以重写于此，是因为这段话里包含了“**包裹**”的概念，该概念并非仅针对特征选择，那些所有基于学习器的性能作为评价准则对数据集进行预处理的方法（**预处理过程依赖训练所得学习器的测试性能**）均可以称之为“包裹式算法”。特别地，本节介绍的 LVW 方法只是包裹式特征选择方法的其中一种而已。

本节内容简单易懂，几乎不需要什么注解。

图 11.1 中，第 1 行$E=\infty$表示初始化学习器误差为无穷大，以至于第 1 轮迭代第 9 行的条件就一定为真；第 2 行$d=|A|$中的$|A|$表示特征集A的包含的特征个数；第 9 行$E'<E$表示学习器$\mathfrak{L}$在特征子集A'上的误差比当前特征子集A上的误差更小，$(E'=E)\vee(d'<d)$表示学习器$\mathfrak{L}$在特征子集A'上的误差与当前特征子集A上的误差相当但A'中包含的特征数更小；$\wedge$表示“逻辑与”，$\vee$表示“逻辑或”。注意到，第 5 行至第 17 行的while循环中t并非一直增加，当第 9 行条件满足时t会被清零。

本节最后提到，“若有运行时间限制，则有可能给不出解”，对这一点我很好奇的是，为什么不在到了规定时间时，直接将第 13 行的中间结果输出呢？

最后，本节 LVW 算法基于拉斯维加斯方法框架，可以仔细琢磨体会拉斯维加斯方法和蒙特卡罗方法的区别。网上较为流行的解释如下（原文出处不详）：

蒙特卡罗算法：采样越多，越近似最优解；

拉斯维加斯算法：采样越多，越有机会找到最优解；

举个例子，假如筐里有 100 个苹果，让我每次闭眼拿 1 个，挑出最大的。于是我随机拿 1 个，再随机拿 1 个跟它比，留下大的，再随机拿 1 个……我每拿一次，留下的苹果都至少不比上次的小。拿的次数越多，挑出的苹果就越大，但我除非拿 100 次，否则无法肯定挑出了最大的。这个挑苹果的算法，就属于蒙特卡罗算法——尽量找好的，但不保证是最好的。

而拉斯维加斯算法，则是另一种情况。假如有一把锁，给我 100 把钥匙，只有 1 把是对的。于是我每次随机拿 1 把钥匙去试，打不开就再换 1 把。我试的次数越多，打开（最优解）的机会就越大，但在打开之前，那些错的钥匙都是没有用的。这个试钥匙的算法，就是拉斯维加斯的——尽量找最好的，但不保证能找到。

11.4 嵌入式选择与 L_1 正则化

“嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成，即在学习器训练过程中自动地进行了特征选择。”，具体可以对比本节式(11.7)的例子与前两节方法的本质区别，细细体会本节第一段的这句有关“嵌入式”的概念描述。

1、式(11.5)的解释

该式与第 3 章式(3.4)基本一致，只需将第三章的$f(x_i)$替换为$\boldsymbol{w}^{\top}\boldsymbol{x}_i$。当然，式(3.4)描述的是输入属性只有一个，而式(11.5)描述的是输入属性有多个，但这并不影响该式的形式。

2、式(11.6)的解释

实际上，若学习过 Coursera 上吴恩达的机器学习课程，本式就是带有正则化项的线性回归优化目标函数。

对应到第 133 页 6.4 节最后一段话，式(11.6)中的$\|\boldsymbol{w}\|_2^2$就是式(6.42)中的正则化项$\Omega(f)$，即采用L_2范数的正则化框架。

3、式(11.7)的解释

在 6.4 节最后一段话提到，“L_2范数$\|\boldsymbol{w}\|_2$倾向于$\boldsymbol{w}$的分量取值尽量均衡，即非零分量个数尽量稠密，而L_0范数$\|\boldsymbol{w}\|_0$和L_1范数$\|\boldsymbol{w}\|_1$则倾向于$\boldsymbol{w}$的分量取值尽量稀疏，即非零分量个数尽量少”。

首先，介绍一下L_0范数和L_1范数的定义：L_0范数$\|\boldsymbol{w}\|_0$等于向量$\boldsymbol{w}$非零元素的个数，L_1范数$\|\boldsymbol{w}\|_1$等于向量$\boldsymbol{w}$所有元素绝对值（模值）之和。

其次，书中第 252 页左下角注释中也提到，“事实上，对$\boldsymbol{w}$施加‘稀疏约束’最自然的是使用L_0范数，但L_0范数不连续，难以优化求解，因此常使用L_1范数来近似”。

最后，式(11.7)称为 LASSO，这并不是一个具体的算法，而是一个优化问题，因此个人认为部分文献中称“LASSO 算法”是不严谨的。

4、图 11.2 的解释

书中对该图的解释已经很清楚了，而且通俗易懂。这里再重复强调地是，所谓“稀疏”，即$\boldsymbol{w}$非零分量个数尽量少；本例中$\boldsymbol{w}=(w_1,w_2)$为二维向量，当交点在坐标轴时，则必有其一为 0，即$\boldsymbol{w}$为稀疏的（注意：两个分量不能都等于 0，否则还预测啥呢？）。

因此，“基于L_1正则化的学习方法就是一种嵌入式特征选择方法，其特征选择过程与学习器训练过程（即求解$\boldsymbol{w}$，笔者注）融为一体，同时完成”。

5、式(11.8)的解释

从本式开始至本节结束，都在介绍近端梯度下降求解L_1正则化问题。若将本式对应到式(11.7)，则本式中$f(\boldsymbol{w})=\sum_{i=1}^{m}(y^i-\boldsymbol{w}^\top\boldsymbol{x}_i)^2$，注意变量为$\boldsymbol{w}$（若感觉不习惯就将其用$\boldsymbol{x}$替换好了）。最终推导结果仅含$f(\boldsymbol{w})$的一阶导数$\nabla f(\boldsymbol{w})=-\sum_{i=1}^{m}2(y^i-\boldsymbol{w}^\top\boldsymbol{x}_i)\boldsymbol{x}_i$ 。

6、式(11.9)的解释

该式即为L-Lipschitz(利普希茨)条件的定义。简单来说，该条件约束函数的变化不能太快。将式(11.9)变形则更为直观（**注：式中应该是 2 范数，而非 2 范数平方**）：

$$\frac{\|\nabla f(\boldsymbol{x}')-\nabla f(\boldsymbol{x})\|_2}{\|\boldsymbol{x}'-\boldsymbol{x}\|_2}\leqslant L,\quad(\forall\boldsymbol{x},\boldsymbol{x}')$$

进一步地，若$\boldsymbol{x}'\to\boldsymbol{x}$，即

$$\lim_{\boldsymbol{x}'\to\boldsymbol{x}}\frac{\|\nabla f(\boldsymbol{x}')-\nabla f(\boldsymbol{x})\|_2}{\|\boldsymbol{x}'-\boldsymbol{x}\|_2}$$

这明显是在求解函数$\nabla f(\boldsymbol{x})$的导数绝对值（模值）。

因此，式(11.9)即要求$f(\boldsymbol{x})$的二阶导数不大于L，其中L称为Lipschitz常数。

“Lipschitz连续”很常见，知乎有一个问答(https://www.zhihu.com/question/51809602)对Lipschitz连续的解释很形象：以陆地为例，Lipschitz连续就是说这块地上没有特别陡的坡；其中最陡的地方有多陡呢？这就是所谓的Lipschitz常数。

7、式(11.10)的推导

对于函数$f(x)$，由定积分的定义，则

$$f(x_2)-f(x_1)=\int_{x_1}^{x_2}f'(x)\mathrm{d}x$$

其中$f'(x)$表示函数$f(x)$的导数。

设$f(\boldsymbol{x})$二次连续可微，$\boldsymbol{x}_k\in\mathbb{R}^n$，令$g(t)=f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)$，则

$$g(0)=f\left(\boldsymbol{x}_k+0\cdot(\boldsymbol{x}-\boldsymbol{x}_k)\right)=f(\boldsymbol{x}_k)$$
$$g(1)=f\left(\boldsymbol{x}_k+1\cdot(\boldsymbol{x}-\boldsymbol{x}_k)\right)=f(\boldsymbol{x})$$

即$f(\boldsymbol{x})-f(\boldsymbol{x}_k)=g(1)-g(0)$。由定积分的定义，则

$$f(\boldsymbol{x})-f(\boldsymbol{x}_k)=g(1)-g(0)=\int_0^1 g'(t)\mathrm{d}t$$

而对于$g'(t)$：（注意求导时变量为t）

$$\begin{aligned}g'(t)&=\frac{\partial f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)}{\partial t}\\&=\nabla f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\end{aligned}$$

代入以上定积分表达式

$$\begin{aligned}f(\boldsymbol{x})-f(\boldsymbol{x}_k)&=\int_0^1 \nabla f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}t\\&=\int_0^1 \left[\nabla f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)-\nabla f\left(\boldsymbol{x}_k\right)+\nabla f\left(\boldsymbol{x}_k\right)\right]^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}t\\&=\int_0^1 \nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}t\\&+\int_0^1 \left[\nabla f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)-\nabla f\left(\boldsymbol{x}_k\right)\right]^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}t\\&\leqslant\int_0^1 \nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}t\\&+\int_0^1 \left\|\nabla f\left(\boldsymbol{x}_k+t(\boldsymbol{x}-\boldsymbol{x}_k)\right)-\nabla f\left(\boldsymbol{x}_k\right)\right\|_2\cdot\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2\mathrm{d}t\\&\leqslant\int_0^1 \nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}t+\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2\cdot\int_0^1 L\left\|t(\boldsymbol{x}-\boldsymbol{x}_k)\right\|_2\mathrm{d}t\\&=\nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)\int_0^1 1\mathrm{d}t+L\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2\cdot\int_0^1 t\mathrm{d}t\\&=\nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)+\frac{L}{2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2\end{aligned}$$

上式推导过程中，第 1 次“$\leqslant$”缩放使用了内积性质：

$$\boldsymbol{x}^{\top}\boldsymbol{y}=\|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\|\cos(\theta)\leqslant\|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\|$$

其中θ为向量$\boldsymbol{x},\boldsymbol{y}$的夹角；第 2 次“$\leqslant$”缩放使用式(11.9)；而且注意积分变量为$t$，与积分变量无关的项（如$\nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)$和$\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2$等）可以作为常量拿到积分号外面。对上式缩放结果移项，得

$$f(\boldsymbol{x})\leqslant f(\boldsymbol{x}_k)+\nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)+\frac{L}{2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2$$

不等式左侧即为式(11.10)的第 1 行表达式（上式第 2 项为内积，可表示为$\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle$，此时与式(11.10)完全相同），令

$$\hat{f}(\boldsymbol{x})=f(\boldsymbol{x}_k)+\nabla f\left(\boldsymbol{x}_k\right)^{\top}(\boldsymbol{x}-\boldsymbol{x}_k)+\frac{L}{2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2$$

则有$f(\boldsymbol{x})\leqslant\hat{f}(\boldsymbol{x})$，即该式不小于$f(\boldsymbol{x})$；容易验证，当$\boldsymbol{x}=\boldsymbol{x}_k$时不等式的等号成立。该不等式在文献中常被称为“descent lemma”。

注意到$\hat{f}(\boldsymbol{x})$表达式与$f(\boldsymbol{x})$在$\boldsymbol{x}_k$附近用二次泰勒展开相似（符号“$\sim$”表示“约等于”）：

$$f(\boldsymbol{x}) \simeq f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^\top \nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k)$$

其中$\nabla^2 f(\boldsymbol{x}_k)$为 Hesse 矩阵，并且正定（此处仅顺便提一下而已，别无它图）。

接下来，对式(11.10)的第 2 行进行变形：

$$\begin{aligned}
&\frac{L}{2}\left\|\boldsymbol{x} - \left(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)\right\|_2^2 = \frac{L}{2}\left\|(\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right\|_2^2 \\
=&\frac{L}{2}\left((\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)^\top \left((\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right) \\
=&\frac{L}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{L}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^\top \frac{1}{L}\nabla f(\boldsymbol{x}_k) \\
&+ \frac{L}{2}\frac{1}{L}\nabla f(\boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{L}{2}\frac{1}{L}\nabla f(\boldsymbol{x}_k)^\top \frac{1}{L}\nabla f(\boldsymbol{x}_k) \\
=&\frac{L}{2}\|(\boldsymbol{x} - \boldsymbol{x}_k)\|_2^2 + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k\rangle + \frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|_2^2
\end{aligned}$$

与式(11.10)第 1 行对比发现，与变量$\boldsymbol{x}$有关的项$\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k\rangle$和$\frac{L}{2}\|(\boldsymbol{x} - \boldsymbol{x}_k)\|_2^2$完全相同，差别仅在于与变量$\boldsymbol{x}$无关的项，而这些项又不会对求$f(\boldsymbol{x})$的最小值位置产生影响，因此可统一用第 2 行的const表示。

实际上，从式(11.10)第 1 行出发，亦可推导出相同的结果：

$$\begin{aligned}
&f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{L}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) \\
=&\frac{L}{2}\left((\boldsymbol{x} - \boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + 2\cdot\frac{1}{L}\nabla f(\boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{L}\nabla f(\boldsymbol{x}_k)^\top \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right) \\
&- \frac{L}{2}\frac{1}{L}\nabla f(\boldsymbol{x}_k)^\top \frac{1}{L}\nabla f(\boldsymbol{x}_k) + f(\boldsymbol{x}_k) \\
=&\frac{L}{2}\left((\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)^\top \left((\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right) + \text{const} \\
=&\frac{L}{2}\left\|\boldsymbol{x} - \left(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)\right\|_2^2 + \text{const}
\end{aligned}$$

其中$\text{const} = -\frac{L}{2}\frac{1}{L}\nabla f(\boldsymbol{x}_k)^\top \frac{1}{L}\nabla f(\boldsymbol{x}_k) + f(\boldsymbol{x}_k)$，为与变量$\boldsymbol{x}$无关的项。

8、式(11.11)的解释

由于const为与变量$\boldsymbol{x}$无关的项，即变量$\boldsymbol{x}$无论取何值均不发生变化，因此$f(\boldsymbol{x})$的极值点的位置仅由第 1 项决定。显然第 1 项非负，并且当$\boldsymbol{x} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$时其值等于 0，即最小值。

9、式(11.12)的解释

注意$\hat{f}(\boldsymbol{x})$在式(11.11)处取得最小值，因此，以下不等式肯定成立：

$$\hat{f}(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)) \leqslant \hat{f}(\boldsymbol{x}_k)$$

在式(11.10)推导中有$f(\boldsymbol{x}) \leqslant \hat{f}(\boldsymbol{x})$恒成立，因此，以下不等式肯定成立：

$$f(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)) \leqslant \hat{f}(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k))$$

在式(11.10)推导中还知道$f(\boldsymbol{x}_k) = \hat{f}(\boldsymbol{x}_k)$，因此

$$f(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)) \leqslant \hat{f}(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)) \leqslant \hat{f}(\boldsymbol{x}_k) = f(\boldsymbol{x}_k)$$

也就是说通过迭代$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$可以使$f(\boldsymbol{x})$的函数值逐步下降。

同理，对于函数$g(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1$，可以通过最小化$\hat{g}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1$逐步求解。式(11.12)就是在最小化$\hat{g}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1$。

以上优化方法称为 Majorization-Minimization，此处不再细述，具体可参见 CSDN 博客《Majorization-Minimization 优化框架》（https://blog.csdn.net/jbb0523/article/details/52134630）。

10、式(11.14)的推导

为什么要将式(11.8)转化为式(11.12)的迭代呢？这是因为式(11.12)的优化问题容易求解。所有形如式(11.13)的优化问题均有闭式解，即式(11.14)，推导过程如下：（注意，在推导过程中，应用了前提条件：Lipschitz常数$L > 0$和正则化参数$\lambda > 0$）

令式(11.13)的目标函数为$g(\boldsymbol{x})$，即

$$\begin{aligned} g(\boldsymbol{x}) =& \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \\ =& \frac{L}{2}\sum_{i=1}^{d}\left\|x^i - z^i\right\|_2^2 + \lambda\sum_{i=1}^{d}\left\|x^i\right\|_1 \\ =& \sum_{i=1}^{d}\left(\frac{L}{2}\left(x^i - z^i\right)^2 + \lambda\left|x^i\right|\right) \end{aligned}$$

其中x^i与z^i分别是$\boldsymbol{x} \in \mathbb{R}^d$与$\boldsymbol{z} \in \mathbb{R}^d$的第$i$个分量；由上式可见，$g(\boldsymbol{x})$可以拆成$d$个独立的函数，求解式(11.13)可以分别求解$d$个独立的目标函数。

针对目标函数$g(x^i)$，

$$g(x^i) = \frac{L}{2}\left(x^i - z^i\right)^2 + \lambda\left|x^i\right|$$

欲求函数的极值，可先对该函数求导，得

$$\frac{\mathrm{d}g(x^i)}{\mathrm{d}x^i} = L\left(x^i - z^i\right) + \lambda\mathrm{sgn}(x^i)$$

其中$\mathrm{sgn}(x^i) = \begin{cases} 1, & x^i > 0 \\ -1, & x^i < 0 \end{cases}$，这是因为$|x^i| = \begin{cases} x^i, & x^i > 0 \\ -x^i, & x^i < 0 \end{cases}$，暂不考虑$x^i = 0$时的导数（单独讨论）。为求极值点，令导数等于 0，得

$$x^i = z^i - \frac{\lambda}{L}\mathrm{sgn}(x^i)$$

上式即为方程导数等于 0 的根，但两边均含x^i，以下分三种情况讨论：

(a)第一种情况：当$z^i > \frac{\lambda}{L}$时：

①假设此时的根$x^i < 0$，则$\text{sgn}(x^i) = -1$，所以$x^i = z^i + \frac{\lambda}{L} > 0$，与假设矛盾；

②假设此时的根$x^i > 0$，则$\text{sgn}(x^i) = 1$，所以$x^i = z^i - \frac{\lambda}{L} > 0$，成立；

即，此时$g(x^i)$在$x^i = z^i - \frac{\lambda}{L} > 0$处取得极小值。是不是最小值呢？

$$\begin{aligned} g(x^i)|_{x^i=z^i-\frac{\lambda}{L}} &= \frac{L}{2}\left(z^i - \frac{\lambda}{L} - z^i\right)^2 + \lambda\left(z^i - \frac{\lambda}{L}\right) \\ &= \frac{L}{2}\left(-\frac{\lambda^2}{L^2} + 2\frac{\lambda}{L}z^i - (z^i)^2 + (z^i)^2\right) \\ &= -\frac{L}{2}\left(\frac{\lambda}{L} - z^i\right)^2 + \frac{L}{2}(z^i)^2 \\ &\leqslant \frac{L}{2}(z^i)^2 = g(x^i)|_{x^i=0} \end{aligned}$$

也就是说$x^i > 0$区间的极小值小于$g(x^i)|_{x^i=0}$；而对于$x^i < 0$区间，结合条件$z^i > \frac{\lambda}{L}$：

$$\begin{aligned} \frac{\mathrm{d}g(x^i)}{\mathrm{d}x^i} &= L\left(x^i - z^i\right) + \lambda\text{sgn}(x^i) \\ &= L\left(x^i - z^i\right) - \lambda \\ &\leqslant L\left(x^i - \frac{\lambda}{L}\right) - \lambda \\ &= Lx^i - 2\lambda < 0 \end{aligned}$$

导数小于零，说明函数在$x^i < 0$区间是单调递减函数；虽然$g(x^i)$在$x^i = 0$处不可导，但在此处连续，因此对于任意$x^i < 0$，$g(x^i) \geqslant g(x^i)|_{x^i=0}$。

综上所述，当$z^i > \frac{\lambda}{L}$时，$g(x^i)$在$x^i = z^i - \frac{\lambda}{L} > 0$处取得极小值，而该极小值小于等于$g(x^i)|_{x^i=0}$，并且又有对于任意$x^i < 0$，$g(x^i) \geqslant g(x^i)|_{x^i=0}$，因此该极小值是最小值。

(b)第二种情况：当$z^i < -\frac{\lambda}{L}$时：

①假设此时的根$x^i > 0$，则$\text{sgn}(x^i) = 1$，所以$x^i = z^i - \frac{\lambda}{L} < 0$，与假设矛盾；

②假设此时的根$x^i < 0$，则$\text{sgn}(x^i) = -1$，所以$x^i = z^i + \frac{\lambda}{L} < 0$，成立；

即，此时$g(x^i)$在$x^i = z^i + \frac{\lambda}{L} < 0$处取得极小值。是不是最小值呢？

$$\begin{aligned} g(x^i)|_{x^i=z^i+\frac{\lambda}{L}} &= \frac{L}{2}\left(z^i+\frac{\lambda}{L}-z^i\right)^2-\lambda\left(z^i+\frac{\lambda}{L}\right) \\ &= \frac{L}{2}\left(-\frac{\lambda^2}{L^2}-2\frac{\lambda}{L}z^i-(z^i)^2+(z^i)^2\right) \\ &= -\frac{L}{2}\left(\frac{\lambda}{L}+z^i\right)^2+\frac{L}{2}(z^i)^2 \\ &\leqslant \frac{L}{2}(z^i)^2 = g(x^i)|_{x^i=0} \end{aligned}$$

也就是说$x^i<0$区间的极小值小于$g(x^i)|_{x^i=0}$；而对于$x^i>0$区间，结合条件$z^i<-\frac{\lambda}{L}$：

$$\begin{aligned} \frac{\mathrm{d}g(x^i)}{\mathrm{d}x^i} &= L\left(x^i-z^i\right)+\lambda\mathrm{sgn}(x^i) \\ &= L\left(x^i-z^i\right)+\lambda \\ &\geqslant L\left(x^i+\frac{\lambda}{L}\right)+\lambda \\ &= Lx^i+2\lambda>0 \end{aligned}$$

导数大于零，说明函数在$x^i>0$区间是单调递增函数；虽然$g(x^i)$在$x^i=0$处不可导，但在此处连续，因此对于任意$x^i>0$，$g(x^i)\geqslant g(x^i)|_{x^i=0}$。

综上所述，当$z^i<-\frac{\lambda}{L}$时，$g(x^i)$在$x^i=z^i+\frac{\lambda}{L}<0$处取得极小值，而该极小值小于等于$g(x^i)|_{x^i=0}$，并且又有对于任意$x^i>0$，$g(x^i)\geqslant g(x^i)|_{x^i=0}$，因此该极小值是最小值。

(c)第三种情况：当$-\frac{\lambda}{L}<z^i<\frac{\lambda}{L}$时(即$|z^i|<\frac{\lambda}{L}$)：

①假设此时的根$x^i>0$，则$\mathrm{sgn}(x^i)=1$，所以$x^i=z^i-\frac{\lambda}{L}<0$，与假设矛盾；

②假设此时的根$x^i<0$，则$\mathrm{sgn}(x^i)=-1$，所以$x^i=z^i+\frac{\lambda}{L}>0$，与假设矛盾；

即无论$x^i>0$还是$x^i<0$函数$g(x^i)$导数均不等于 0，即$g(x^i)$没有极值点，那么$x^i=0$是否为函数$g(x^i)$的极值点呢？

对于$\Delta x\neq 0$，

$$\begin{aligned} g(x^i)|_{x^i=\Delta x} &= \frac{L}{2}\left(\Delta x-z^i\right)^2+\lambda\left|\Delta x\right| \\ &= \frac{L}{2}\left((\Delta x)^2-2\Delta x\cdot z^i+\frac{2\lambda}{L}\left|\Delta x\right|\right)+\frac{L}{2}(z^i)^2 \\ &= \frac{L}{2}\left((\Delta x)^2-2\Delta x\cdot z^i+\frac{2\lambda}{L}\left|\Delta x\right|\right)+g(x^i)|_{x^i=0} \end{aligned}$$

当$\Delta x>0$时，结合条件$z^i<\frac{\lambda}{L}$，有$\Delta x\cdot z^i<\Delta x\cdot\frac{\lambda}{L}$，因此

$$
\begin{aligned}
g(x^i)|_{x^i=\Delta x} &= \frac{L}{2}\left((\Delta x)^2 - 2\Delta x \cdot z^i + \frac{2\lambda}{L}|\Delta x|\right) + g(x^i)|_{x^i=0} \\
&> \frac{L}{2}\left((\Delta x)^2 - 2\Delta x \cdot \frac{\lambda}{L} + \frac{2\lambda}{L}\Delta x\right) + g(x^i)|_{x^i=0} \\
&= \frac{L}{2}(\Delta x)^2 + g(x^i)|_{x^i=0} \\
&> g(x^i)|_{x^i=0}
\end{aligned}
$$

当$\Delta x < 0$时，结合条件$z^i > -\frac{\lambda}{L}$，有$-\Delta x \cdot z^i > \Delta x \cdot \frac{\lambda}{L}$（注意：$-\Delta x > 0$，不等式两边同乘以$-\Delta x$，不等式方向保持改变），因此

$$
\begin{aligned}
g(x^i)|_{x^i=\Delta x} &= \frac{L}{2}\left((\Delta x)^2 - 2\Delta x \cdot z^i + \frac{2\lambda}{L}|\Delta x|\right) + g(x^i)|_{x^i=0} \\
&> \frac{L}{2}\left((\Delta x)^2 + 2\Delta x \cdot \frac{\lambda}{L} - \frac{2\lambda}{L}\Delta x\right) + g(x^i)|_{x^i=0} \\
&= \frac{L}{2}(\Delta x)^2 + g(x^i)|_{x^i=0} \\
&> g(x^i)|_{x^i=0}
\end{aligned}
$$

因此，函数在$x^i = 0$处取得极小值，也是最小值。

综合以上三种情况，$g(x^i)$的最小值在以下位置取得：

$$
\arg\min\ g(x^i) = \begin{cases} z^i - \frac{\lambda}{L}, & z^i > \frac{\lambda}{L} \\ 0, & |z^i| < \frac{\lambda}{L} \\ z^i + \frac{\lambda}{L}, & z^i < -\frac{\lambda}{L} \end{cases}
$$

即式(11.14)。

该式称为软阈值(Soft Thresholding)函数，很常见，建议掌握。另外，常见的变形是式(11.13)中的$L = 1$或$L = 2$时的形式，其解直接代入式(11.14)即可。与软阈值函数相对的是硬阈值函数，是将式(11.13)中的 1 范数替换为 0 范数的优化问题的闭式解。

11.5 稀疏表示与字典学习

稀疏表示与字典学习实际上是信号处理领域的概念。本节内容核心就是 K-SVD 算法。

1、式(11.15)的解释

字典矩阵$\mathbf{B} \in \mathbb{R}^{d\times k}$，样本$\boldsymbol{x}_i \in \mathbb{R}^d$，稀疏表示系数$\boldsymbol{\alpha}_i \in \mathbb{R}^k$，稀疏表示希望

$$
\boldsymbol{x}_i \simeq \mathbf{B}\boldsymbol{\alpha}_i
$$

即$\boldsymbol{x}_i$可近似由$\mathbf{B}$所有列的线性组合表示（矩阵$\mathbf{B}$右乘列向量$\boldsymbol{\alpha}_i$等于矩阵$\mathbf{B}$所有列的线性组合）；若$\boldsymbol{\alpha}_i$的k个元素中非零元素个数为$n \ll k$，即$\boldsymbol{\alpha}_i$是稀疏的，则$\boldsymbol{x}_i$可由字典$\mathbf{B}$稀疏表示。

以上描述中有两个关键点：$\boldsymbol{x}_i$与$\mathbf{B}\boldsymbol{\alpha}_i$尽可能相等、$\boldsymbol{\alpha}_i$非零元素个数尽可能少，这两点在式(11.15) 分别体现为第 1 项和第 2 项。

该式就是 K-SVD 算法要解决的问题。

2、式(11.16)的解释

该式目标函数就是式(11.15)中的第i项，是 K-SVD 算法第 1 步：Sparse Coding Stage。

注意，该式目标函数中存在矩阵$\mathbf{B}$，因此与式(11.13)不同。但是，可参照 LASSO 的解法，将$\|\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i\|_2^2$视为式(11.8)中的$f(\boldsymbol{x})$，则可使用近端梯度下降求解。所有推导过程均不变，只需要求出$\|\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i\|_2^2$对$\boldsymbol{\alpha}_i$的偏导数，代入式(11.12)即可；当然，$\|\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i\|_2^2$需要满足式(11.9)的$L$-Lipschitz条件。

实际上，该式就是在字典矩阵$\mathbf{B}$上的稀疏表示问题，所有稀疏表示的方法都可以直接应用，如匹配追踪[Mallat & Zhang, 1993]、基追踪降噪[Chen et al., 1998]等。

3、式(11.17)的推导

该式及式(11.18)都属于 K-SVD 算法第 2 步：Codebook Update Stage。

该式固定$\boldsymbol{\alpha}_i$，因此式(11.15)中的第 2 项相对于矩阵$\mathbf{B}$为常量，不影响极值点的位置，故在目标函数中略去不写。注意到$\mathbf{B}\boldsymbol{\alpha}_i\in\mathbb{R}^d$，$\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i$为列向量，$\|\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i\|_2^2$为列向量的 2 范数平方，即向量中各元素的平方和。而$\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i$实际是矩阵$\mathbf{X}-\mathbf{B}\mathbf{A}$的第$i$列，$\sum_{i=1}^m\|\boldsymbol{x}_i-\mathbf{B}\boldsymbol{\alpha}_i\|_2^2$则是矩阵$\mathbf{X}-\mathbf{B}\mathbf{A}$所有列 2 范数平方之和，即矩阵$\mathbf{X}-\mathbf{B}\mathbf{A}$的有元素的平方和，即 F 范数的平方$\|\mathbf{X}-\mathbf{B}\mathbf{A}\|_F^2$。这是矩阵的性质，即矩阵的 F 范数平方等于矩阵各列（行）的 2 范数平方之和。

4、式(11.18)的推导

这里只推导一下$\mathbf{B}\mathbf{A}=\sum_{j=1}^k\boldsymbol{b}_j\boldsymbol{\alpha}^j$，其中$\boldsymbol{b}_j$表示$\mathbf{B}$的第$j$列，$\boldsymbol{\alpha}^j$表示$\mathbf{A}$的第$j$行。

接下来，我们用b_j^i,α_j^i分别表示$\mathbf{B}$和$\mathbf{A}$的第i行第j列的元素。首先计算$\boldsymbol{b}_j\boldsymbol{\alpha}^j$：

$$\boldsymbol{b}_j\boldsymbol{\alpha}^j=\begin{bmatrix}b_j^1\\b_j^2\\\vdots\\b_j^d\end{bmatrix}\begin{bmatrix}\alpha_1^j&\alpha_2^j&\cdots&\alpha_m^j\end{bmatrix}=\begin{bmatrix}b_j^1\alpha_1^j&b_j^1\alpha_2^j&\cdots&b_j^1\alpha_m^j\\b_j^2\alpha_1^j&b_j^2\alpha_2^j&\cdots&b_j^2\alpha_m^j\\\vdots&\vdots&\ddots&\vdots\\b_j^d\alpha_1^j&b_j^d\alpha_2^j&\cdots&b_j^d\alpha_m^j\end{bmatrix}_{d\times m}$$

然后求和，得

$$\begin{aligned}\sum_{j=1}^k\boldsymbol{b}_j\boldsymbol{\alpha}^j&=\sum_{j=1}^k\left(\begin{bmatrix}b_j^1\\b_j^2\\\vdots\\b_j^d\end{bmatrix}\begin{bmatrix}\alpha_1^j&\alpha_2^j&\cdots&\alpha_m^j\end{bmatrix}\right)\\&=\begin{bmatrix}\sum_{j=1}^k b_j^1\alpha_1^j&\sum_{j=1}^k b_j^1\alpha_2^j&\cdots&\sum_{j=1}^k b_j^1\alpha_m^j\\\sum_{j=1}^k b_j^2\alpha_1^j&\sum_{j=1}^k b_j^2\alpha_2^j&\cdots&\sum_{j=1}^k b_j^2\alpha_m^j\\\vdots&\vdots&\ddots&\vdots\\\sum_{j=1}^k b_j^d\alpha_1^j&\sum_{j=1}^k b_j^d\alpha_2^j&\cdots&\sum_{j=1}^k b_j^d\alpha_m^j\end{bmatrix}_{d\times m}\end{aligned}$$

再次计算$\mathbf{BA}$：

$$\mathbf{BA} = \begin{bmatrix} b_1^1 & b_2^1 & \cdots & b_k^1 \\ b_1^2 & b_2^2 & \cdots & b_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ b_1^d & b_2^d & \cdots & b_k^d \end{bmatrix}_{d\times k} \cdot \begin{bmatrix} \alpha_1^1 & \alpha_2^1 & \cdots & \alpha_m^1 \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^k & \alpha_2^k & \cdots & \alpha_m^k \end{bmatrix}_{k\times m}$$

$$= \begin{bmatrix} \sum_{j=1}^k b_j^1\alpha_1^j & \sum_{j=1}^k b_j^1\alpha_2^j & \cdots & \sum_{j=1}^k b_j^1\alpha_m^j \\ \sum_{j=1}^k b_j^2\alpha_1^j & \sum_{j=1}^k b_j^2\alpha_2^j & \cdots & \sum_{j=1}^k b_j^2\alpha_m^j \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^k b_j^d\alpha_1^j & \sum_{j=1}^k b_j^d\alpha_2^j & \cdots & \sum_{j=1}^k b_j^d\alpha_m^j \end{bmatrix}_{d\times m}$$

对比可知$\mathbf{BA} = \sum_{j=1}^k \boldsymbol{b}_j\boldsymbol{\alpha}^j$。其实这就是矩阵运算的基本性质，记住就好~

5、K-SVD 算法

Task: Find the best dictionary to represent the data samples $\{\mathbf{y}_i\}_{i=1}^N$ as sparse compositions, by solving

$$\min_{\mathbf{D},\mathbf{X}} \left\{\|\mathbf{Y}-\mathbf{DX}\|_F^2\right\} \quad \text{subject to} \quad \forall i,\ \|\mathbf{x}_i\|_0 \le T_0.$$

Initialization : Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbb{R}^{n\times K}$ with ℓ^2 normalized columns. Set $J = 1$.

Repeat until convergence (stopping rule):

- *Sparse Coding Stage*: Use any pursuit algorithm to compute the representation vectors $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximating the solution of

$$i = 1, 2, \ldots, N, \quad \min_{\mathbf{x}_i} \left\{\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2\right\} \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \le T_0.$$

- *Codebook Update Stage*: For each column $k = 1, 2, \ldots, K$ in $\mathbf{D}^{(J-1)}$, update it by
  - Define the group of examples that use this atom, $\omega_k = \{i \mid 1 \le i \le N,\ \mathbf{x}_T^k(i) \ne 0\}$.
  - Compute the overall representation error matrix, $\mathbf{E}_k$, by

$$\mathbf{E}_k = \mathbf{Y} - \sum_{j\ne k} \mathbf{d}_j\mathbf{x}_T^j.$$

  - Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to ω_k, and obtain $\mathbf{E}_k^R$.
  - Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T$. Choose the updated dictionary column $\tilde{\mathbf{d}}_k$ to be the first column of $\mathbf{U}$. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of $\mathbf{V}$ multiplied by $\boldsymbol{\Delta}(1,1)$.
- Set $J = J + 1$.

Fig. 2. The K-SVD algorithm.

本节前半部分铺垫概念，后半部分核心就是 K-SVD。作为字典学习的最经典的算法，K-SVD[Aharon et al., 2006]自 2006 年发表以来引用量已超七千次（谷歌学术）。理解 K-SVD 的基础是 SVD，即奇异值分解，参见西瓜书附录 A.3。

对于任意实矩阵$\mathbf{A} \in \mathbb{R}^{m\times n}$，都可分解为$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$，其中$\mathbf{U} \in \mathbb{R}^{m\times m}, \mathbf{V} \in \mathbb{R}^{n\times n}$，分别为$m$阶和$n$阶正交矩阵（复数域时称为酉矩阵），即$\mathbf{U}^\top\mathbf{U} = \mathbf{I}, \mathbf{V}^\top\mathbf{V} = \mathbf{I}$（逆矩阵等于自身的转置），$\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$，且除$(\mathbf{\Sigma})_{ii} = \sigma_i$之外其它位置的元素均为零，$\sigma_i$称为奇异值，可以证明，矩阵$\mathbf{A}$的秩等于非零奇异值的个数。

正如西瓜书附录 A.3 所述，K-SVD 分解中主要使用 SVD 解决低秩矩阵近似问题。之所以称为 K-SVD，原文献中专门有说明：

> We shall call this algorithm "K-SVD" to parallel the name K-means. While K-means applies K computations of means to update the codebook, K-SVD obtains the updated dictionary by K SVD computations, each determining one column. A full description of the algorithm is given in Fig. 2.

具体来说，就是原文献中的字典共有 K 个原子（列），因此需要迭代 K 次；这类似于 K 均值算法欲将数据聚成 K 个簇，需要计算 K 次均值。

K-SVD 算法伪代码详见原文献 Fig.2（如图所示），符号与本节符号有差异。具体来说，原文献中字典矩阵用$\mathbf{D}$表示（书中用$\mathbf{B}$），稀疏系数用$\mathbf{x}_i$表示（书中用$\boldsymbol{\alpha}_i$），数据集用$\mathbf{Y}$表示（书中用$\mathbf{X}$）。

在初始化字典矩阵$\mathbf{D}$以后，K-SVD 算法迭代过程分两步：第 1 步 Sparse Coding Stage 就是普通的已知字典矩阵$\mathbf{D}$的稀疏表示问题，可以使用很多现成的算法完成此步，不再详述；K-SVD 的核心创新点在第 2 步 Codebook Update Stage，在该步骤中分 K 次分别更新字典矩阵$\mathbf{D}$中每一列，更新第k列$\mathbf{d}_k$时其它各列都是固定的，如原文献式(21)所示：

$$\begin{aligned}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 &= \left\|\mathbf{Y}-\sum_{j=1}^{K}\mathbf{d}_j\mathbf{x}_T^j\right\|_F^2 \\ &= \left\|\left(\mathbf{Y}-\sum_{j\neq k}\mathbf{d}_j\mathbf{x}_T^j\right)-\mathbf{d}_k\mathbf{x}_T^k\right\|_F^2 \\ &= \left\|\mathbf{E}_k-\mathbf{d}_k\mathbf{x}_T^k\right\|_F^2.\end{aligned}$$

注意，矩阵$\mathbf{d}_k\mathbf{x}_T^k$的秩为 1（其中，$\mathbf{x}_T^k$表示稀疏系数矩阵$\mathbf{X}$的第$k$行，以区别于其第$k$列$\mathbf{x}_k$），对比西瓜书附录 A.3 中的式(A.34)，这就是一个低秩矩阵近似问题，即对于给定矩阵$\mathbf{E}_k$，求其最优 1 秩近似矩阵$\mathbf{d}_k\mathbf{x}_T^k$；此时可对$\mathbf{E}_k$进行 SVD 分解，类似于西瓜书附录式(A.35)，仅保留最大的 1 个奇异值；具体来说$\mathbf{E}_k = \mathbf{U}\mathbf{\Delta}\mathbf{V}^\top$，仅保留$\mathbf{\Delta}$中最大的奇异值$\mathbf{\Delta}(1,1)$，则$\mathbf{d}_k\mathbf{x}_T^k = \mathbf{U}_1\mathbf{\Delta}(1,1)\mathbf{V}_1^\top$，其中$\mathbf{U}_1, \mathbf{V}_1$分别为$\mathbf{U}, \mathbf{V}$的第 1 列，此时$\mathbf{d}_k = \mathbf{U}_1$，$\mathbf{x}_T^k = \mathbf{\Delta}(1,1)\mathbf{V}_1^\top$。但这样更新会破坏第 1 步中得到的稀疏系数的稀疏性！

为了保证第 1 步中得到的稀疏系数的稀疏性，K-SVD 并不直接对$\mathbf{E}_k$进行 SVD 分解，而是根据$\mathbf{x}_T^k$仅取出与$\mathbf{x}_T^k$非零元素对应的部分列，例如行向量$\mathbf{x}_T^k$只有第 1、3、5、8、9 个元素非零，则仅取出$\mathbf{E}_k$的第 1、3、5、8、9 列组成矩阵进行 SVD 分解$\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^\top$，则

$$\tilde{\mathbf{d}}_k = \mathbf{U}_1,\ \tilde{\mathbf{x}}_T^k = \mathbf{\Delta}(1,1)\mathbf{V}_1^\top$$

即得到更新后的$\tilde{\mathbf{d}}_k$和$\tilde{\mathbf{x}}_T^k$（注意，此时的行向量$\tilde{\mathbf{x}}_T^k$长度仅为原$\mathbf{x}_T^k$非零元素个数，需要按原$\mathbf{x}_T^k$对其余位置填 0）。如此迭代 K 次即得更新后的字典矩阵$\tilde{\mathbf{D}}$，以供下一轮 Sparse Coding 使用。K-SVD 原文献中特意提到，在 K 次迭代中要使用最新的稀疏系数$\tilde{\mathbf{x}}_T^k$，但并没有说是否要用最新的$\tilde{\mathbf{d}}_k$（个人猜测应该也要用最新的$\tilde{\mathbf{d}}_k$）：

In the K-SVD algorithm, we sweep through the columns and use always the most updated coefficients as they emerge from preceding SVD steps. Parallel versions of this algorithm can also be considered, where all updates of the previous dictionary are done based on the same $\mathbf{X}$. Experiments show that while this version also converges, it yields an inferior solution and typically requires more than four times the number of iterations.

有关奇异值分解，可阅读博客园中的以下两篇博客：(1)奇异值分解(SVD)原理与在降维中的应用(https://www.cnblogs.com/pinard/p/6251584.html)；(2)[机器学习笔记]奇异值分解SVD 简介及其在推荐系统中的简单应用(https://www.cnblogs.com/lzllovesyl/p/5243370.html)。

11.6 压缩感知

个人认为，虽然压缩感知与稀疏表示关系密切，但它是彻彻底底的信号处理领域的概念。西瓜书作者在本章有几个专业术语翻译与信号处理领域人士的习惯翻译略不一样（虽然如何翻译并不关键）：第 258 页的 Restricted Isometry Property (RIP)作者翻译为“限定等距性”，信号处理领域一般翻译为“有限等距性质”；第 259 页的 Basis Pursuit De-Noising、第 261 页的 Basis Pursuit 和 Matching Pursuit 中的“Pursuit”作者翻译为“寻踪”，信号处理领域一般翻译为“追踪”，即基追踪降噪、基追踪、匹配追踪。

压缩感知的相关理论比较复杂，感兴趣的话可参见本人 CSDN 博客中“压缩感知(Compressive Sensing)”分类下的文章（https://blog.csdn.net/jbb0523/article/category/2637103）。

1、式(11.21)的解释

将式(11.21)进行变形

$$(1-\delta_k) \leqslant \frac{\|\mathbf{A}_k \boldsymbol{s}\|_2^2}{\|\boldsymbol{s}\|_2^2} \leqslant (1+\delta_k)$$

注意不等式中间，若$\boldsymbol{s}$为输入信号，则分母$\|\boldsymbol{s}\|_2^2$为输入信号的能量，分子$\|\mathbf{A}_k \boldsymbol{s}\|_2^2$为对应的观测信号的能量，即 RIP 要求观测信号与输入信号的能量之比在一定的范围之内；例如当δ_k等于 0 时，观测信号与输入信号的能量相等，即实现了等距变换（参见：杨孝春. 欧氏空间中的等距变换与等距映射[J]. 四川工业学院学报, 1999(1):72-76.）；RIP 放松了对矩阵$\mathbf{A}$的约束（而且$\mathbf{A}$并非方阵），因此称为“有限”等距性质。

2、式(11.24)的解释

该式实际是在矩阵$\mathbf{X}$低秩的假设条件下，恢复$\mathbf{X}$中的未知项。

矩阵低秩已成为研究热点，低秩的确是线性代数中所讲的“低秩”，但应该理解其实际内含，具体可参见知乎问答“矩阵低秩的意义?”(https://www.zhihu.com/question/28630628)，特别点赞的是答主“Rebecca Lyu”有关“为什么叫秩”的回答和答主“柒生”有关“图像所包含的信息的丰富程度”的回答。

3、式(11.25)的解释

该式即为核范数定义：矩阵的核范数（迹范数）为矩阵的奇异值之和。

有关“凸包”的概念，引用百度百科里的两句原话：在二维欧几里得空间中，凸包可想象为一条刚好包著所有点的橡皮圈；用不严谨的话来讲，给定二维平面上的点集，凸包就是

将最外层的点连接起来构成的凸多边形，它能包含点集中所有的点。

个人理解，将$\mathrm{rank}(\mathbf{X})$的“凸包”是$\mathbf{X}$的核范数$\|\mathbf{X}\|_*$这件事简单理解为$\|\mathbf{X}\|_*$是$\mathrm{rank}(\mathbf{X})$的上限即可，即$\|\mathbf{X}\|_*$恒大于$\mathrm{rank}(\mathbf{X})$，类似于式(11.10)中的式子恒大于$f(\boldsymbol{x})$。

11.7 本章小节

其实本章后三节的内容更多属于信号处理领域，前四节虽然在介绍机器学习中的特征选择问题，但也是一个很笼统的介绍，包括“过滤式”算法、“包裹式”算法、“嵌入式”算法也是通用的概念；因此本章与其说是介绍机器学习特征选择的相关知识，倒不如说是介绍机器学习与信号处理交叉领域的相关内容。

西瓜书作者通过 11.4 节巧妙的将特征选择与稀疏学习联系在一起，不得不说这是一个很好的安排，毕竟稀疏学习也是“学习”嘛~

基于矩阵的低秩假设对矩阵缺失值进行填充也是当今的研究热点。不知道作者本章安排的用意是否是为了在介绍特征选择的同时，将当今的热点“低秩”“稀疏”“压缩感知”等统统囊括进来（即使仅是蜻蜓点水），以扩展西瓜书知识面的覆盖广度。

特别地，注意到鲁棒主成分分析（RPCA）的原始文献（Candès, E. J., Li, X., **Ma, Y.**, & Wright, J. (2011). Robust principal component analysis?. Journal of the ACM (JACM), 58(3), Article 11.）的第三作者 Ma Y.，他就是马毅老师（微博 毅马当闲，个人主页，百度百科），ACM Fellow，IEEE Fellow，2017 年当选 ACM Fellow 的理由是“对计算机视觉和模式识别的低维模型的理论和应用作出贡献”。