

高效稀疏编码算法

张晋

2018 年 2 月 4 日

1 稀疏编码

1.1 介绍

稀疏编码是一种无监督学习方法，它通过寻找一组**过完备 (overcomplete)** 的基向量来更高效地表示样本数据。

具体来说，若输入向量为 $\vec{\xi}^{(1)}, \vec{\xi}^{(2)}, \dots, \vec{\xi}^{(m)}$ ，该算法要找到一组基向量 $\vec{b}_1, \dots, \vec{b}_n \in \mathbb{R}^k$ ($n > k$)，使得我们能将输入向量 $\vec{\xi} \in \mathbb{R}^k$ 表示为这些基向量的线性组合：

$$\xi = \sum_{j=1}^n \vec{b}_j s_j \quad (1)$$

其中， $\vec{s} \in \mathbb{R}^n$ 是对应的稀疏系数向量。

虽然一组**完备**的基向量就可以将输入向量全部表示出来，并且很快很方便，例如主成分分析法 (PCA)，但我们依旧选择**过完备基 (overcomplete basis)**，因为它能更有效地找出隐含在输入数据内部的结构与模式。

但在选择过完备基的同时，基向量的系数 \vec{s} 却不是唯一确定的了，为了使学到的特征更明显，我们需要给系数 \vec{s} 加上一个**稀疏性**的限制，那么我们可以定义代价函数为：

$$\sum_{i=1}^m \left[\frac{1}{2\sigma^2} \|\vec{\xi}^{(i)} - \sum_{j=1}^n \vec{b}_j s_j^{(i)}\|^2 + \beta \sum_{j=1}^n \phi(s_j^{(i)}) \right] \quad (2)$$

其中 β 是常系数， $\phi(\cdot)$ 是稀疏代价函数，可以取以下任何一种：

$$\phi(s_j) = \begin{cases} \|s_j\|_1 & (L_1 \text{ 罚函数}) \\ \sqrt{s_j^2 + \epsilon} & (\text{epsilon}L_1 \text{ 罚函数}) \\ \log(1 + s_j^2) & (\log \text{ 罚函数}). \end{cases}$$

本文中采用的是 L_1 罚函数, 因为 L_1 正则化能产生稀疏系数, 并且对不相关的特征具有鲁棒性。

此外, 我们需要给 \vec{b}_j 加上一些约束条件, 因为若 b 无约束, 那么 b 就能在 s 线性减小的同时线性增大, 并使得 $\sum_{j=1}^n \vec{b}_j s_j^{(i)}$ 不变, 最后 $s \rightarrow 0, b \rightarrow \infty$, 这显然不是我们想要的结果。因此, 我们需要限制 b 的大小: $\|\vec{b}_j\|^2 \leq c, \forall j = 1, \dots, n$. 包含了限制条件的稀疏编码代价函数的完整形式如下:

$$\begin{aligned} \text{minimize}_{\{\vec{b}_j\}, \{\vec{s}^{(i)}\}} \quad & \sum_{i=1}^m \left[\frac{1}{2\sigma^2} \|\vec{\xi}^{(i)} - \sum_{j=1}^n \vec{b}_j s_j^{(i)}\|^2 + \beta \sum_{j=1}^n \phi(s_j^{(i)}) \right] \\ \text{subject to} \quad & \|\vec{b}_j\|^2 \leq c, \forall j = 1, \dots, n. \end{aligned} \quad (3)$$

可以将问题 (3) 表示成更简洁的矩阵形式, 记 $X \in \mathbb{R}^{k \times m}$ 为输入向量 $\vec{\xi}^{(1)}, \vec{\xi}^{(2)}, \dots, \vec{\xi}^{(m)}$ 排成的矩阵; 记 $B \in \mathbb{R}^{k \times n}$ 为基向量矩阵, 由基向量 $\vec{b}_1, \dots, \vec{b}_n$ 组成; 记 $S \in \mathbb{R}^{n \times m}$ 为系数矩阵, 由系数向量 $\vec{s}^{(1)}, \vec{s}^{(2)}, \dots, \vec{s}^{(m)}$ 组成。那么原优化问题 (3) 可以写成以下形式:

$$\begin{aligned} \text{minimize}_{B, S} \quad & \frac{1}{2\sigma^2} \|X - BS\|_F^2 + \beta \sum_{i,j} \phi(s_{i,j}) \\ \text{subject to} \quad & \sum_i B_{i,j}^2 \leq c, \forall j = 1, \dots, n. \end{aligned} \quad (4)$$

1.2 概率解释

Olshausen 和 Field 在 1997 年提出了这个基于超完备基的稀疏编码算法, 并从概率的角度解释了稀疏编码算法的**生成模型 (generative model)**。

首先假定输入 ξ 是 n 个特征 \vec{b}_j 的线性组合, 并加上高斯噪声 ν :

$$\xi = \sum_{j=1}^n \vec{b}_j s_j + \nu(\xi) \quad (5)$$

其中噪声 ν 服从均值为 0、协方差为 $\sigma^2 I$ 的高斯分布, 那么有:

$$P(X | B, S) = \frac{1}{2\pi\sigma} \exp \left(-\frac{1}{2\sigma^2} (X - BS)^\top (X - BS) \right) \quad (6)$$

然后给定系数向量 s 一个先验分布 $P(s)$ ，由于我们期望的 s 是稀疏的，所以先验分布 $P(s)$ 通常取一个峰值很尖锐且接近 0 的分布。常见的有 Laplace、Cauchy、Student-t 分布。在此我们选择 Laplace 分布：

$$P(s_i) = \text{Laplace}(s_i; 0, \frac{1}{\lambda}) = \frac{\lambda}{2} e^{-\lambda|s_i|} \quad (7)$$

同时每个 s_i 之间都是独立的，因此：

$$P(S) = \prod_{i=1}^n P(s_i) \quad (8)$$

对于基 B 可能的分布一无所知，所以根据**同等无知原则**，先验地认为基服从均匀分布 $P(B) = 1/\theta$ ，那么基和系数的最大后验估计为：

$$\begin{aligned} [\hat{B}, \hat{S}]_{MAP}(X) &= \arg \max_{B, S} \frac{P(X | B, S) P(B) P(S)}{\iint_{\Theta} P(X | B', S') P(B') P(S') dB' dS'} \\ &= \arg \max_{B, S} P(X | B, S) P(S) \\ &= \arg \max_{B, S} \log P(X | B, S) + \log P(S) \\ &= \arg \max_{B, S} -\frac{1}{2\sigma^2} (X - BS)^T (X - BS) - \lambda \sum_{j=1}^n \|s_j\|_1 \\ &= \arg \min_{B, S} \frac{1}{2\sigma^2} \|X - BS\|_F^2 + \lambda \sum_{j=1}^n \|s_j\|_1 \end{aligned}$$

不难看出，如果先验分布 $P(s)$ 取柯西分布 $P(s_j) = \beta/(1 + s_j^2)$ 的话，那么得到的正则化项为 $\sum_{i=1}^m \sum_{j=1}^n \log(1 + s_{i,j}^2)$ ，对应稀疏代价函数 $\phi(\cdot)$ 取 \log 罚函数的情况。

同时，如果我们没有对 B, S 假定一个先验分布，而是直接求基与系数的极大似然估计 $[\hat{B}, \hat{S}]_{ML}(X) = \arg \max_{B, S} P(X | B, S)$ 的话，那么得到的就是没有罚函数项的重构误差代价函数。

2 问题求解

观察式 (4)，可以看出当固定住 S 时，这是个关于 B 的凸优化问题，而固定住 B 时，这也是个关于 S 的凸优化问题，但是组合到一起时却是非凸的。因此考虑在固定住其中一个的情况下，对另一个进行迭代优化，并且采用交替着进行迭代优化的方法。

当固定住 S 时, 问题 (4) 就成了一个带着二次约束的最小二乘问题, 此时可以采用 Lagrange 对偶法将其化成无约束二次优化。

当固定住 B 时, 问题 (4) 成了一个标准的 LASSO (least absolute shrinkage and selection operator) 问题, 由于 L_1 正则化项不可微, 所以许多直接基于梯度的优化方法都不可用, 要解决这种问题, 可以使用通用 QP 求解器 (例如 CVX)、内点法、修正最小角回归法 (LARS)、嫁接法 (grafting) 等, 但本文介绍一种新的方法——特征符号搜索法 (feature-sign search algorithm) 可以更高效地解决这个问题。

2.1 系数的学习——特征符号搜索法

2.1.1 Motivation

虽然 L_1 项在 0 处不可微, 但可以先对 x_i 的符号进行假设, 在此基础上消去绝对值, 然后就可以求得子问题的最优解 x_{new} , 并在线段 x_c 到 x_{new} 之间进行线搜索, 为了减小计算量和保证稀疏性, 只需要搜索在线段与坐标平面的交点即可, 然后在里面找出原问题的最优点, 由于子问题也是原问题的众多种情况之一, 因此在子问题中通往最优解的这条线段上, 也能找到点在原问题中是下降的。得到新解后, 根据已知信息对符号进行合理的假设, 以保障下一次的迭代也会下降。最后由于 x 符号的状态有限, 加上每一次都是严格下降的, 所以最终一定能收敛到最优解。

2.1.2 子问题

对于原问题:

$$\text{minimize}_x \quad f(x) \equiv \|y - Ax\|^2 + \gamma \|x\|_1 \quad (9)$$

特征符号搜索法通过维持一个非零系数的积极集并记录其系数的符号来使问题得到简化: 用 *active set* 记录非零系数的标号, 用 $\theta_i \in \{-1, 0, 1\}$ 记录 x_i 的符号, 那么 $\gamma \|x\|_1$ 就可以表示为 $\gamma \theta^\top x$, 用 \hat{x} 表示 x 中非零的子向量, $\hat{\theta}$ 是对应的符号向量, 那么 $\gamma \theta^\top x$ 可以进一步简化为 $\gamma \hat{\theta}^\top \hat{x}$, 再用 \hat{A} 表示 A 中对应的子矩阵, 原优化问题就转化成了以下优化问题:

$$\text{minimize}_{\hat{x}} \quad f(x) \equiv \|y - \hat{A}\hat{x}\|^2 + \gamma \hat{\theta}^\top \hat{x} \quad (10)$$

$$\nabla f = -2\hat{A}^\top(y - \hat{A}\hat{x}) + \gamma\hat{\theta}$$

令 $\nabla f = 0$ 解得¹

$$\hat{x} = (\hat{A}^\top \hat{A})^{-1}(\hat{A}^\top y - \gamma\hat{\theta}/2)$$

2.1.3 具体算法

Algorithm 1 特征符号搜索法

- 1: 初始化 $x := \vec{0}$, $\theta := \vec{0}$, 积极集 $active\ set := \{\}$.
 - 2: 从值为 0 的 $x_i \in x$ 中, 选出 $i = \arg \max_i \left| \frac{\partial \|y - Ax\|^2}{\partial x_i} \right|$.
 如果 x_i 的变动能使目标值减小, 那就把 i 添加到积极集中去:
 若 $\frac{\partial \|y - Ax\|^2}{\partial x_i} > \gamma$, 令 $\theta_i := -1$, $active\ set := \{i\} \cup active\ set$.
 若 $\frac{\partial \|y - Ax\|^2}{\partial x_i} < -\gamma$, 令 $\theta_i := 1$, $active\ set := \{i\} \cup active\ set$.
 - 3: 特征符号步
 求解子问题 (10), 得到解 $\hat{x}_{new} = (\hat{A}^\top \hat{A})^{-1}(\hat{A}^\top y - \frac{1}{2}\gamma\hat{\theta})$
 在 \hat{x} 到 \hat{x}_{new} 的闭线段上进行离散的线搜索:
 找到 \hat{x}_{new} 和所有改变符号处的点, 即 zero-crossing 点:

$$\hat{x} - \frac{\hat{x}_j}{(\hat{x}_{new})_j - \hat{x}_j}(\hat{x}_{new} - \hat{x}), \quad j \in active\ set$$

 计算在这些点处的目标函数值, 并取最小值点作为更新
 从 $active\ set$ 中删去新解 \hat{x} 中值为 0 的指标, 并更新 $\theta = \text{sign}(x)$
 - 4: 检查最优性条件
 (a) 对非零系数: $\frac{\partial \|y - Ax\|^2}{\partial x_j} + \gamma \text{sign}(x_j) = 0, \quad \forall x_j \neq 0$
 如果不符合条件 (a), 转到第 3 步; 否则检查条件 (b)
 (b) 对零系数: $\left| \frac{\partial \|y - Ax\|^2}{\partial x_i} \right| \leq \gamma, \quad \forall x_j = 0$
 如果不符合条件 (b), 转到第 2 步; 否则返回 x 作为最优解。
-

¹前提是 \hat{A} 列满秩。若 $\hat{A}^\top \hat{A}$ 是奇异的, 首先检查是否 $\hat{A}^\top y - \gamma\hat{\theta}/2 \in \mathcal{R}(\hat{A}^\top \hat{A})$, 若成立, 则说明有解, 只是解不唯一, 此时可以用伪逆来代替; 若不成立, 说明无解, 此时朝任一满足条件 $z \in \mathcal{N}(\hat{A}^\top \hat{A})$ 的方向 z 移动 \hat{x} , 直到其与坐标平面相交, 然后更新 \hat{x} 。这两种情况对于目标函数都是严格下降的, 因此算法的收敛性不变。

2.1.4 收敛性证明

引理 2.1. 将在第 3 步开始时的解记为 x_c , 若其符号与 θ 保持一致且不是问题 (10) 的最优解, 算法保证第 3 步更新后的解对于目标值来说是严格下降的。

证明. 记 \hat{x}_c 是 x_c 的非零子向量, 考虑一个关于 \hat{x} 的光滑二次函数 $\tilde{f}(\hat{x}) = \|y - \hat{A}\hat{x}\|^2 + \gamma\hat{\theta}^\top \hat{x}$. 由于 \hat{x}_c 不是 \tilde{f} 的最优点, 故有 $\tilde{f}(\hat{x}_{new}) < \tilde{f}(\hat{x}_c)$. 现在有两种情况:

1. 若 \hat{x}_{new} 跟原积极集及符号向量一致, 显然 \hat{x}_{new} 更优, 更新 $\hat{x}_c = \hat{x}_{new}$
2. \hat{x}_{new} 跟原积极集及符号向量不一致, 这说明 \hat{x}_c 到 \hat{x}_{new} 的闭线段必然穿过了坐标平面, 取 \hat{x}_d 为该线段与坐标平面第一个相交的点 (也就是说 \hat{x}_d 中只有一个维度的值变成了 0, 其它维度的符号与原符号向量保持一致), 显然 $\hat{x}_c \neq \hat{x}_d$, 由 \tilde{f} 的凸性可知 $\tilde{f}(\hat{x}_d) < \tilde{f}(\hat{x}_c)$ ², 由于 \hat{x}_d 没有变号, 故有 $f(\hat{x}_d) = \tilde{f}(\hat{x}_d) < \tilde{f}(\hat{x}_c) = f(\hat{x}_c)$

□

引理 2.2. 将在第 2 步开始时的解记为 x_c , 若其是问题 (10) 的最优解并且不是问题 (9) 的最优解, 算法保证第 3 步更新后的解对于目标值来说是严格下降的

证明. 由于 x_c 是问题 (10) 的最优解, 所以其必然满足最优性条件 (a), 但其又不是问题 (9) 的最优解, 所以必然不满足最优性条件 (b); 因此必然存在一个或多个 i 使得 $\left| \frac{\partial \|y - Ax\|^2}{\partial x_i} \right| > \gamma$, 不妨记第 2 步中添加到积极集 *active set* 中的指标为 i 。

在第 3 步中, 记 $\tilde{f}(\hat{x}) = \|y - \hat{A}\hat{x}\|^2 + \gamma\hat{\theta}^\top \hat{x}$, 那么有:

1. 将 \tilde{f} 在 \hat{x}_c 处展开, 其一阶项只有 x_i (由最优性条件 (a), 其他 x_j 的一阶偏导都为 0), 又由第 2 步可知: 若 $\nabla \tilde{f} > 0$, 则 $\theta_i = -1$, 若 $\nabla \tilde{f} < 0$, 则 $\theta_i = 1$, 记 $\vec{a} = (0, \dots, 0, \theta_i, 0, \dots, 0)^\top$, 则有 $-\nabla \tilde{f} // \vec{a}$, 即 \tilde{f} 的下降方向 \vec{p} 与 \vec{a} 夹锐角, 那么任一沿 \vec{p} 方向上的点 $\hat{x} = \hat{x}_c + \alpha \vec{p}$ 中 x_i 项的系数为 αp_i , 因为 $\alpha p_i \theta_i > 0$, 故沿 \tilde{f} 下降方向更新会使得 \hat{x}_i 的符号与 θ 保持一致。
2. 由于 \hat{x}_c 不是 \tilde{f} 的最优点, 由 \tilde{f} 的凸性, 在 \hat{x}_c 点处沿着 $\hat{x}_{new} - \hat{x}$ 方向必然使 \tilde{f} 严格下降。

² \hat{x}_{new} 是凸函数 \tilde{f} 的极小点, 而 $\tilde{f}(\hat{x}_d)$ 又在 \hat{x}_c 到 \hat{x}_{new} 的线段上, 因此 \hat{x}_d 处的目标函数值严格比 \hat{x}_c 处的小

综上：在线段 \hat{x}_c 到 \hat{x}_{new} 上的点的符号必然与 θ 保持一致，由符号一致可以得出 $\tilde{f} = f$ ，那么使 \tilde{f} 严格下降的点也必然会使 f 严格下降。 \square

引理 2.3. 本算法能保证在有限步之内收敛到问题 (9) 的全局极小点

证明. 对第 2 步开始时的解 x_c ，无论 x_c 是由第 1 步初始化得来还是由第 4 步 (b) 转来， x_c 都是满足最优性条件 (a) 的，那么由引理 2.2 可知第 3 步后得到的解必然严格下降。

对第 3 步开始时的解 x_c ，若它是由第 2 步得来，那么上面论证过在第 3 步结束后的解其必然能严格下降；若它是由第 4 步 (a) 转来，那么它的符号必然与 θ 保持一致，由引理 2.2 可知第 3 步结束后得到的解必然严格下降

由于积极集 *active set* 和符号向量 θ 的组合的状态数目是有限的，再加上每一次迭代都使得目标值严格下降，因此迭代不会存在循环，必然在有限步之内得到最优解停止迭代。 \square

此外， x 的初始值不一定非得为 $\vec{0}$ ，任意初始值都可以，但是在根据 x 初始化完 θ 和 *active set* 后，需要跳过第 2 步，直接从第 3 步开始。

2.2 基的学习——Lagrange 对偶法

当固定住 S 时，问题 (4) 可以简化为以下问题：

$$\begin{aligned} & \text{minimize}_B \quad \|X - BS\|_F^2 \\ & \text{subject to} \quad \sum_{i=1}^k B_{i,j}^2 \leq c, \forall j = 1, \dots, n. \end{aligned} \quad (11)$$

记 $\Lambda = \text{diag}(\vec{\lambda})$ ，可以使用 Lagrange 对偶法来解决该优化问题：³

$$\begin{aligned} \mathcal{L}(B, \vec{\lambda}) &= \text{tr} \left((X - BS)^\top (X - BS) \right) + \sum_{j=1}^n \lambda_j (\sum_{i=1}^k B_{i,j}^2 - c) \\ &= \text{tr} \left((X - BS)^\top (X - BS) \right) + \text{tr}(\Lambda B^\top B - c\Lambda) \end{aligned} \quad (12)$$

³ 在下面的运算中要用到迹 (**trace**) 的以下性质：

1. $df = \sum_{i,j} \frac{\partial f}{\partial X_{ij}} dX_{ij} = \text{tr} \left(\frac{\partial f}{\partial X}^\top dX \right)$
2. $d(\text{tr}(X)) = \text{tr}(dX)$
3. $\text{tr}(A^\top) = \text{tr}(A)$
4. 对尺寸相同的矩阵 A, B 有： $\text{tr}(A^\top B) = \sum_{i,j} A_{ij} B_{ij} = \text{tr}(BA^\top)$
5. 对尺寸相同的方阵 A, B, C 有： $\text{tr}(ABC) = \text{tr}(ACB) = \text{tr}(BAC)$

$$\begin{aligned}
d\mathcal{L} &= \text{tr} \left((-dBS)^\top (X - BS) + (X - BS)^\top (-dBS) + \Lambda dB^\top B + \Lambda B^\top dB \right) \\
&= \text{tr} \left(-2S(X - BS)^\top dB + 2\Lambda B^\top dB \right) \\
&= \text{tr} \left((-2(X - BS)S^\top + 2B\Lambda^\top)^\top dB \right)
\end{aligned} \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial B} = -2(X - BS)S^\top + 2B\Lambda^\top = 0 \tag{14}$$

$$\Rightarrow B = XS^\top(\Lambda + SS^\top)^{-1} \tag{15}$$

将 $B = XS^\top(\Lambda + SS^\top)^{-1}$ 代入式 (12) 中得：

$$\begin{aligned}
\mathcal{L} &= \text{tr} \left(X^\top X - X^\top BS - S^\top B^\top X + S^\top B^\top BS + \Lambda B^\top B - c\Lambda \right) \\
&= \text{tr} \left(X^\top X - 2BSX^\top + B^\top BSS^\top + B^\top B\Lambda - c\Lambda \right) \\
&= \text{tr} \left(X^\top X - 2BSX^\top + B^\top XS^\top - c\Lambda \right) \\
&= \text{tr} \left(X^\top X - BSX^\top - c\Lambda \right) \\
&= \text{tr} \left(X^\top X - XS^\top(SS^\top + \Lambda)^{-1}(XS^\top)^\top - c\Lambda \right)
\end{aligned}$$

$$\mathcal{D}(\vec{\lambda}) = \min_B \mathcal{L}(B, \vec{\lambda}) = \text{tr} \left(X^\top X - XS^\top(SS^\top + \Lambda)^{-1}(XS^\top)^\top - c\Lambda \right) \tag{16}$$

$$d\mathcal{D}(\vec{\lambda}) = \text{tr} \left((XS^\top(SS^\top + \Lambda)^{-1}(XS^\top)^\top - c)d\Lambda \right) \tag{17}$$

$$\Rightarrow \frac{\partial \mathcal{D}(\vec{\lambda})}{\partial \Lambda} = \|XS^\top(SS^\top + \Lambda)^{-1}\|_F^2 - c \tag{18}$$

$$\begin{aligned}
\frac{\partial \mathcal{D}(\vec{\lambda})}{\partial \lambda_i} &= \left(\|XS^\top(SS^\top + \Lambda)^{-1}\|_F^2 \right)_{i,i} - c \\
&= \|XS^\top(SS^\top + \Lambda)^{-1}e_i\|_2^2 - c
\end{aligned} \tag{19}$$

$$\begin{aligned}
d\left(\frac{\partial \mathcal{D}(\vec{\lambda})}{\partial \lambda_i}\right) &= d\left(e_i^\top (SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i\right) \\
&= -\text{tr}\left(e_i^\top (SS^\top + \Lambda)^{-2} d\Lambda (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i + \right. \\
&\quad \left. e_i^\top (SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-2} d\Lambda e_i\right) \\
&= -2\text{tr}\left((XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i e_i^\top (SS^\top + \Lambda)^{-2} d\Lambda\right) \\
\frac{\partial \Lambda}{\partial \lambda_j} &= e_j e_j^\top
\end{aligned} \tag{20}$$

$$\begin{aligned}
\frac{\partial^2 \mathcal{D}(\vec{\lambda})}{\partial \lambda_i \partial \lambda_j} &= -2\text{tr}\left((XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i e_i^\top (SS^\top + \Lambda)^{-2} e_j e_j^\top\right) \\
&= -2\text{tr}\left((SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i e_i^\top (SS^\top + \Lambda)^{-1} e_j e_j^\top\right) \\
&= -2\text{tr}\left(e_j^\top (SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i e_i^\top (SS^\top + \Lambda)^{-1} e_j\right) \\
&= -2e_j^\top (SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_i e_i^\top (SS^\top + \Lambda)^{-1} e_j \\
&= -2\left(e_i^\top (SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1} e_j\right) \left(e_i^\top (SS^\top + \Lambda)^{-1} e_j\right) \\
&= -2((SS^\top + \Lambda)^{-1} (XS^\top)^\top (XS^\top) (SS^\top + \Lambda)^{-1})_{i,j} ((SS^\top + \Lambda)^{-1})_{i,j}
\end{aligned} \tag{21}$$

Lagrange 对偶法需要极大化式 (16) 中 $\mathcal{D}(\vec{\lambda})$ ，而在上面的式 (19) 和式 (21) 中我们分别计算出了式 (16) 中 $\mathcal{D}(\vec{\lambda})$ 的梯度与 Hessian 阵，因此可以采用一些梯度方法如牛顿法或共轭梯度法来进行优化。

当确定了 $\mathcal{D}(\vec{\lambda})$ 的极大值点 $\lambda_1, \dots, \lambda_n$ 后，可以求得最优基 B 为：

$$B = XS^\top (\Lambda + SS^\top)^{-1} \tag{22}$$

由于这一类优化问题的稀疏性： $n > k$ ，因此转化为对偶问题后需要优化的变量（Lagrange 乘子）的个数大大减少。

此外，由于基的学习不涉及稀疏惩罚项，因此稀疏函数 $\phi(\cdot)$ 的选择不会影响本问题的求解。

3 实验

实验简述

本实验使用稀疏编码来对图像进行编码的学习，这需要解决两个一般性的凸优化问题：

- **L1 正则化最小二乘问题**——使用特征符号搜索法解决
- **L2 约束最小二乘问题**——使用 Lagrange 对偶转化为无约束 QP 后使用共轭梯度法求解。

实验代码来源于 Honglak Lee 的[高效稀疏编码算法论文主页](#)，实验数据来源于 Olshausen 为他 1996 年的那篇开创性论文编写的[稀疏编码仿真程序主页](#)。

文件说明

- `demo_fast_sc.m` 整个实验的主程序框架
- `IMAGES.mat` 实验数据集
- `getdata_imagearray.m` 将图片转换为输入参数
- `sparse_coding.m` 进行稀疏编码的学习
- `l1ls_featuresign.m` 使用特征符号搜索法学习稀疏系数 S
- `cgf_fitS_sc2.m` 使用共轭梯度法学习稀疏系数 S （不可用）
- `l2ls_learn_basis_dual.m` 使用 Lagrange 对偶学习基 B
- `getObjective2.m` 计算目标函数值（重构误差 + 稀疏惩罚）
- `display_figures.m` 输出图像
- `save_figures.m` 保存实验结果

代码改动说明

由于代码是作者 2007 年编写的，而着期间 MATLAB 版本改动较大，因此需要作出以下调整：

1. 将 demo_fast_sc.m 中第 6 行的

```
opt_choice = 1;
```

改为

```
opt_choice = 2;
```

2. 将 demo_fast_sc.m 中第 10 行的

```
load ../data/IMAGES.mat
```

改为

```
load ../data/IMAGES.mat
```

3. 将 sparse_coding.m 中第 34 行的

```
options = optimset('GradObj','on', 'Hessian','on');
```

改为

```
options =  
↪ optimset('Algorithm','trust-region-reflective'  
↪ , 'GradObj','on', 'Hessian','on');
```

4. 此外，文件夹fast_sc\code\sc2中的 cgf_sc2.dll 是由 32 位 MATLAB 编译的，因此在 64 位的 MATLAB 中无法运行。而这个 dll 将在cgf_fitS_sc2.m 中被调用来进行共轭梯度法的计算，因此在 64 位的 MATLAB 中调用cgf_fitS_sc2.m 时会报错。

实验步骤

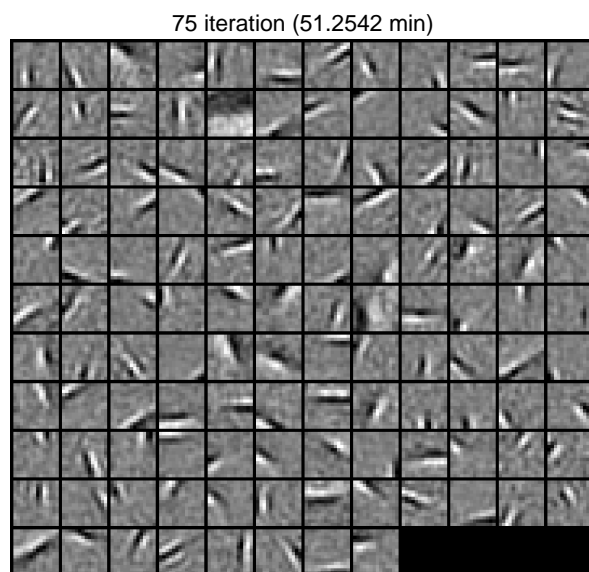
1. 使用`cd` 命令将路径更改到文件当前目录处
2. 在 MATLAB 命令行窗口输入`demo_fast_sc.m(2)`，这意味着你使用特征搜索法调用程序 `l1ls_featuresign.m` 来进行系数 S 的学习。
如果输入 `demo_fast_sc.m(1)`的话，意味着使用共轭梯度法调用程序 `cgf_fitS_sc2.m` 来进行系数 S 的学习（会报错）。
也可以直接运行`demo_fast_sc.m` 程序，它将默认选择特征搜索法。
3. 程序内部每迭代一次会在命令行窗口输出当前信息，并弹出图片展示迭代情况，当 100 次迭代结束或达到终止条件后，程序将自动停止，并保存信息在文件夹`fast_sc\results` 中。

实验结果展示

在文件夹`fast_sc\results` 中可以找到以`.mat` 为后缀的文件，将其拖入 MATLAB 命令行窗口，就能自动载入参数信息。然后在命令行输入：

```
display_figures(pars, stat, B, B, t)
```

就能将迭代情况用图像展示出来：



该图展示了算法学到的 128 个基（每个基都是由 14×14 个像素点组成），迭代了 75 次，花费 51 分钟。

通过这 128 个基的线性组合，基本上可以将原图片给还原出来，可以看出算法学到的基都是一些边缘图像，这与人类大脑类似，神经生理研究表明：在初级视觉皮层 (Primary Visual Cortex) 下细胞的感受野具有显著的方向敏感性，单个神经元仅对处于其感受野中的刺激做出反应，即单个神经元仅对某一频段的信息呈现较强的反映，如特定方向的边缘、线段、条纹等图像特征。

此外，尽管我们选取作为训练集的照片不同，但最后提取出来的特征都是类似的，也说明了这种边缘特征的存在是具有普遍性的，在稀疏性的限制下更有利于其他图像结构的表达。

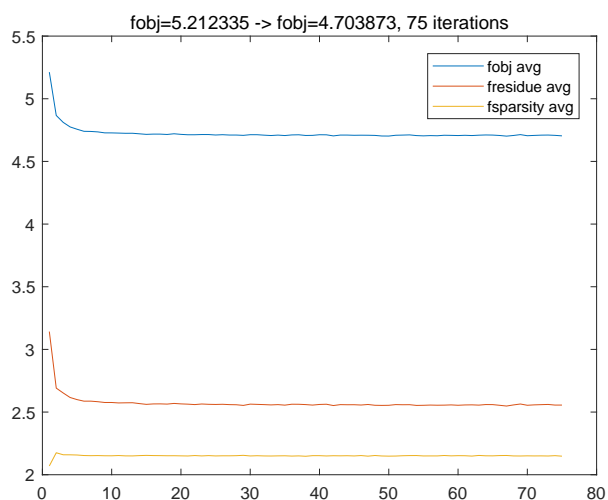
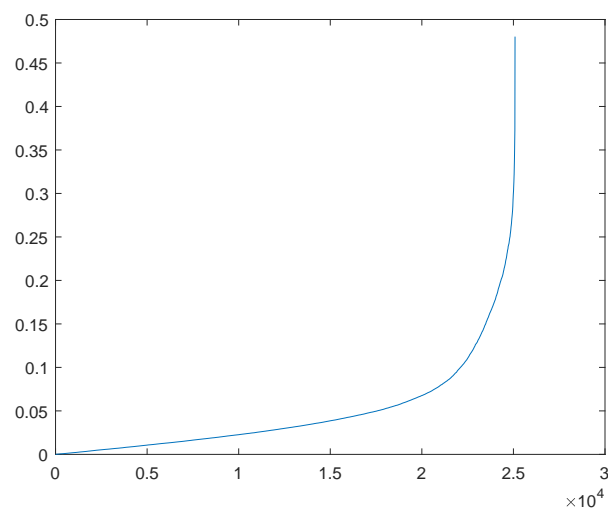
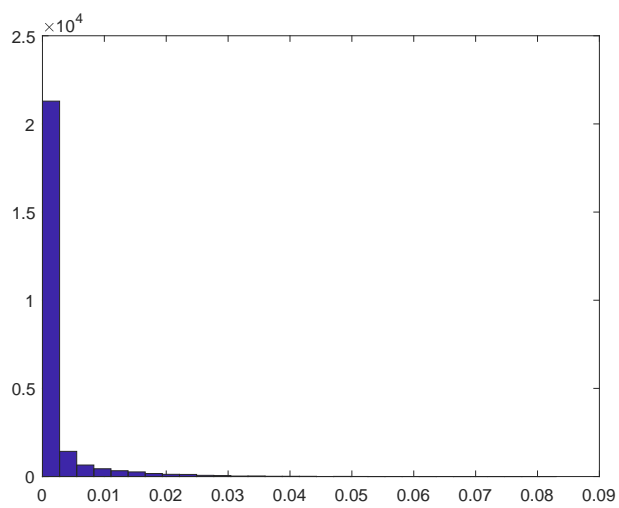


图 1: 目标函数、重构误差、稀疏惩罚值的下降情况

图 2: $S_{i,j}$ 排序后输出的值图 3: $S_{i,j}$ 的 HuberLoss 损失函数直方图

从图2可以看出来，系数 S 的分布有 4/5 都在 0.06 以下，图3更是说明大部分系数都是集中在 0 附近，满足了稀疏性的要求。

A HuberLoss 函数

HuberLoss 是一个用于回归问题的带参数的损失函数。

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{if } |y - f(x)| \leq \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & \text{if } |y - f(x)| > \delta. \end{cases}$$

δ 是 HuberLoss 的参数, y 是真实值, $f(x)$ 是模型的预测值。当预测偏差小于 δ 时, 它采用平方误差, 当预测偏差大于 δ 的时候, 采用的线性误差。相比于最小二乘的线性回归, HuberLoss 降低了对 outlier 的惩罚程度, 所以 HuberLoss 是一种常用的 robust regression 的损失函数。

```

1 % 绘制 HuberLoss 函数的代码
2 diff = linspace(-4, 4, 1000);
3 hold on
4 plot(diff, 0.5.*diff.^2)
5 plot(diff,
    ↪ (abs(diff)<1).*0.5.*diff.^2+(abs(diff)>1).*(abs(diff)-0.5))
6 legend('0.5x^2', 'Huber loss')

```

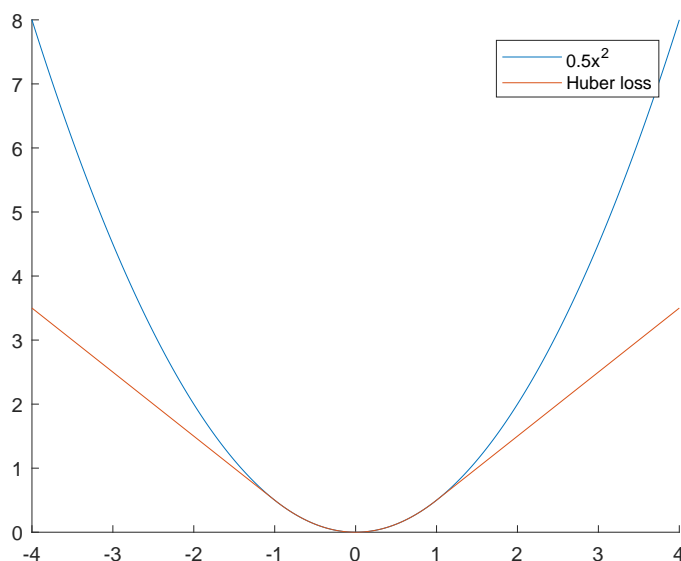


图 4: HuberLoss 损失函数

B Andrew Ng 对于稀疏编码的说明



V1 区是大脑视觉处理的第一阶段，V1 中的神经元起到的**边缘检测器作用 (edge detectors)**，当瞳孔发现了眼前的物体的边缘，而且这个边缘指向某个方向时，对应的 V1 神经元细胞就会开始活跃。

同时，更复杂的结构也能很好地被这些边缘特征线性表达，而将复杂结构组合起来就能表达更复杂的结构，也就是说高层的特征是低层特征的组合。

例如，当人眼看到一个气球时，视网膜得到了最基本的信息——像素点（由瞳孔摄入），接着像素点组合成了边缘和方向（被 V1 区细胞感知），然后大脑判定眼前的物体的形状是圆形的（由 V2 区细胞感知形状），最后进一步判定该物体是只气球（由 V3、V4 等区域感知更高级更抽象的特征）。

First stage of visual processing: V1

V1 is the first stage of visual processing in the brain.
Neurons in V1 typically modeled as edge detectors:



Neuron #1 of visual cortex
(model)

Neuron #2 of visual cortex
(model)

Andrew Ng

