

# 基于卷积高斯混合模型的统计压缩感知\*

汪韧<sup>1)†</sup> 郭静波<sup>2)</sup> 惠俊鹏<sup>1)</sup> 王泽<sup>1)</sup> 刘红军<sup>1)</sup> 许元男<sup>1)</sup> 刘韵佛<sup>1)</sup>

1) (中国运载火箭技术研究院研究发展部, 北京 100076)

2) (清华大学电机工程与应用电子技术系, 北京 100084)

(2019年3月24日收到; 2019年6月20日收到修改稿)

高斯混合模型被广泛应用于统计压缩感知中信号先验概率分布的建模. 利用高斯混合模型对图像的概率分布进行建模时, 通常需要先对图像分块, 再对图像块的概率分布进行建模. 本文提出卷积高斯混合模型对整幅图像的概率分布进行建模. 通过期望极大化算法求解极大边缘似然估计, 实现模型中未知参数的估计. 此外, 考虑到在整幅图像上计算的复杂度较高, 本文在卷积高斯混合模型和压缩测量模型中引入循环卷积, 所有的训练和恢复过程都可以利用二维快速傅里叶变换实现快速运算. 仿真实验表明, 本文所提的 MMLE-convGMM 算法的恢复性能要优于传统的压缩感知算法的恢复性能.

**关键词:** 卷积高斯混合模型, 统计压缩感知, 期望极大化算法, 卷积稀疏编码

**PACS:** 07.50.Qx, 05.45.-a, 05.45.Vx

**DOI:** 10.7498/aps.68.20190414

## 1 引言

压缩感知<sup>[1,2]</sup>实现了以远低于奈奎斯特的采样频率去采样稀疏信号, 并以高概率实现原信号的准确恢复. 贝叶斯压缩感知<sup>[3]</sup>从统计学的角度描述了信号的压缩采样与恢复过程. 根据稀疏贝叶斯学习理论, 需要对信号的先验分布进行建模. 高斯混合模型 (Gaussian mixture models, GMM) 因其具有强大且灵活的拟合能力被广泛应用于信号先验分布的建模中<sup>[4-6]</sup>. 从主成分分析 (principal components analysis, PCA) 的角度来说, GMM 模型中每一类高斯分布的协方差矩阵的 PCA 基张成了信号稀疏分解的子空间<sup>[7]</sup>. 压缩感知自提出以后被广泛应用于物理成像领域, 如光谱成像<sup>[8]</sup>、太赫兹成像<sup>[9]</sup>、图像加密<sup>[10]</sup>、图像重建<sup>[11,12]</sup>等.

尽管 GMM 具有简单灵活的优点, 但由于图像包含了丰富的信息, GMM 对图像 (或视频每一

帧) 的概率分布进行建模时先将整幅图像分割成多个重叠或者不重叠的图像块 (patches), 对每一小块的先验分布函数进行建模, 再独立恢复图像中的每一小块, 最后按其在图像中对应的位置组合成一整幅图像, 其中重叠区域的像素取平均, 这样做的缺点是容易产生图像的分割效应. 另外, 在硬件实现时, 压缩感知是对整幅图像直接进行压缩测量<sup>[13]</sup>. 一种直观的想法是能不能直接对整幅图像的先验分布进行建模, 但该想法实现的困难体现在两方面: 一是整幅图像的信息量相比于图像块要丰富得多, 拟合其先验分布函数的难度会大大增加, 简单的 GMM 对于整幅图像先验分布函数的建模效果不佳; 二是整幅图像的维数比图像块的维数大得多, 若对整幅图像的概率分布进行建模, 计算量将大大增加.

针对上述问题, 本文提出卷积高斯混合模型 (convolutional Gaussian mixture models, convGMM) 对整幅图像的概率分布进行建模. 本

\* 国家自然科学基金 (批准号: 51677094) 资助的课题.

† 通信作者. E-mail: wangren94@126.com

文的立意如下:

1) 将整幅图像包含的复杂信息分为两部分: 背景信息 (background information) 和细节信息 (detail information). 由于背景信息的内容相对较少, 这里延续 GMM 的思想, 将不同均值的线性加权求和来拟合背景信息. 针对复杂的细节信息, 本文从卷积稀疏编码 (convolutional sparse coding, CSC)<sup>[14]</sup> 和解卷积网络 (deconvolutional networks, DN)<sup>[15–17]</sup> 中得到启发, 利用多个滤波器 (filter) 与特征图 (feature map) 卷积的线性加权求和来拟合整幅图像的细节信息. 从图像块到整幅图像建模的关键是使用了解卷积网络. 从直观的角度来说, 解卷积神经网络中滤波器的作用类似于传统压缩感知中的稀疏基. 同时, 本文将 GMM 与解卷积网络相结合, 使得该模型兼具了多个解卷积网络线性加权的灵活性.

2) 考虑到对整幅图像的参数进行训练时, 计算复杂度高, 本文在信号模型和压缩测量模型中都引入了循环卷积, 根据循环卷积所对应的含有循环块的块循环矩阵 (block circulant matrix with circulant blocks, BCCB) 的数学性质<sup>[18]</sup>, 所有的训练和恢复过程都可以利用二维快速傅里叶变换 (two-dimensional fast Fourier transforms, 2D-FFTs) 实现快速运算.

本文的主要研究内容如下: 1) 首先提出 convGMM 对整幅图像的先验分布进行建模; 2) 针对模型中未知参数的学习, 本文采用经典的期望极大化 (expectation maximization, EM) 算法求解极大边缘似然估计 (maximizing the marginal log-likelihood estimation, MMLE); 3) 针对信号的恢复过程, 首先基于先验分布函数和似然函数推导出信号的后验分布函数, 并将后验分布的数学期望作为原信号的估计, 该估计是最小均方误差意义下的估计; 4) 最后在 CIFAR-10 数据集、Caltech 101 数据集和 CelebA 数据集上验证本文的 MMLE-convGMM 算法相比于传统压缩感知算法的优越性.

## 2 convGMM

令  $\mathbf{X}$  表示任意的图像, 传统的 GMM 为

$$p(\mathbf{X}) = \sum_{z=1}^K p(\mathbf{X}|z)p(z), \quad (1)$$

其中,  $p(\mathbf{X}|z)$  服从高斯分布,  $p(z)$  表示第  $z$  个高斯分布所占的权重. 使用 GMM 时需要先将整幅图像分割成多个图像块, 再对每一图像块的概率分布函数进行建模.

本文在此基础上提出 convGMM 对整幅图像  $\mathbf{X}$  的概率分布进行建模:

$$p(\mathbf{X}) = \sum_{z=1}^K \int p(\mathbf{X}|\tilde{\mathbf{S}}, z) p(\tilde{\mathbf{S}}|z) p(z) d\tilde{\mathbf{S}}, \quad (2)$$

其中,

$$p(\mathbf{X}|\tilde{\mathbf{S}}, z) = \mathcal{N}(\mathbf{X}|\mathbf{M}_z + \tilde{\mathbf{F}}_z * \tilde{\mathbf{S}}, \mathbf{E}), \quad \mathbf{E} = \gamma \mathbf{I}, \quad (3)$$

$$p(\tilde{\mathbf{S}}|z) = \mathcal{N}(\tilde{\mathbf{S}}|\mathbf{0}, \mathbf{D}), \quad \mathbf{D} = \mathbf{I}, \quad (4)$$

$$p(z) = \pi_z, \quad (5)$$

\* 表示二维卷积算子,  $\tilde{\mathbf{F}}_z$  表示卷积滤波器 (卷积核或者字典),  $\tilde{\mathbf{S}}$  表示特征图,  $\mathbf{M}_z$  描述的是不同卷积网络的背景. 多个  $\mathbf{M}_z$  ( $z = 1, \dots, K$ ) 的线性加权用来描述图像的背景信息, 多个滤波器与特征图卷积  $\tilde{\mathbf{F}}_z * \tilde{\mathbf{S}}$  ( $z = 1, \dots, K$ ) 的线性加权求和用来拟合图像的细节信息.  $\mathbf{I}$  表示单位阵,  $\pi_z$  表示第  $z$  个卷积网络所占的权重.

如果将图像  $\mathbf{X}$  向量化,  $\mathbf{x} = \text{vec}(\mathbf{X})$ , 则它的分布为

$$p(\mathbf{x}) = \sum_{z=1}^K \pi_z \int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_z + \mathbf{F}_z \mathbf{s}, \gamma \mathbf{I}) \mathcal{N}(\mathbf{s}|\mathbf{0}, \mathbf{I}) d\mathbf{s}, \quad (6)$$

其中,  $\boldsymbol{\mu}_z = \text{vec}(\mathbf{M}_z)$ ;  $\text{vec}(\tilde{\mathbf{F}}_z * \tilde{\mathbf{S}}) = \mathbf{F}_z \mathbf{s} = \mathbf{S} \mathbf{f}_z$ ,  $\mathbf{F}_z$  和  $\mathbf{S}$  表示将卷积转化为乘积所对应的卷积矩阵;  $\mathbf{s} = \text{vec}(\tilde{\mathbf{S}})$ ,  $\mathbf{f}_z = \text{vec}(\tilde{\mathbf{F}}_z)$ , 即将特征图  $\tilde{\mathbf{S}}$  和卷积滤波器  $\tilde{\mathbf{F}}_z$  写成向量形式.

需要特别指出的是, (3) 式中的卷积是循环卷积, 而不是线性卷积. 使用循环卷积的目的是为了在整幅图像上实现快速有效的运算. 如果图像的大小为  $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2 \times N_c}$  (对于灰度图像  $N_c = 1$ , 对于彩色图像  $N_c = 3$ ), 每个滤波器的大小为  $\tilde{\mathbf{F}}_z \in \mathbb{R}^{n_1 \times n_2 \times N_c}$  (一般情况下  $n_1 \ll N_1, n_2 \ll N_2$ ), 则在循环卷积的框架下特征图  $\tilde{\mathbf{S}}$  与图像  $\mathbf{X}$  有着相同的大小, 即  $\tilde{\mathbf{S}} \in \mathbb{R}^{N_1 \times N_2 \times N_c}$ . 根据卷积滤波器  $\tilde{\mathbf{F}}_z$  与卷积矩阵  $\mathbf{F}_z$  之间的对应关系, 卷积矩阵  $\mathbf{F}_z$  是一个有着循环块的块循环矩阵:

$$\mathbf{F}_z = \begin{bmatrix} \mathbf{F}_{z,0} & \mathbf{F}_{z,N_2-1} & \cdots & \mathbf{F}_{z,1} \\ \mathbf{F}_{z,1} & \mathbf{F}_{z,0} & \cdots & \mathbf{F}_{z,2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{z,N_2-1} & \mathbf{F}_{z,N_2-2} & \cdots & \mathbf{F}_{z,0} \end{bmatrix}, \quad (7)$$

其中, 每一块  $\mathbf{F}_{z,j}, j = 0, 1, \dots, N_2 - 1$  是 1 个  $N_1 \times N_1$  的循环矩阵, 根据 BCCB 的性质, 可得

$$\mathbf{F}_z = (\mathbf{W}_{N_2} \otimes \mathbf{W}_{N_1}) \Delta_z (\mathbf{W}_{N_2} \otimes \mathbf{W}_{N_1})^{-1}, \quad (8)$$

其中,  $\mathbf{W}_{N_1}$  表示  $N_1 \times N_1$  大小的离散傅里叶变换矩阵,  $\otimes$  表示 Kronecker 积,  $\Delta_z = \text{diag}(\hat{\mathbf{f}}_z)$ ,  $\hat{\mathbf{f}}_z = (\mathbf{W}_{N_2} \otimes \mathbf{W}_{N_1}) \tilde{\mathbf{f}}_z$ ,  $\tilde{\mathbf{f}}_z$  是卷积矩阵  $\mathbf{F}_z$  的第一行. 因此, 循环卷积运算  $\mathbf{F}_z \mathbf{s} = \text{vec}(\tilde{\mathbf{F}}_z * \tilde{\mathbf{S}})$  可以利用 (8) 式中的一次二维快速傅里叶逆变换 (two-dimensional inverse fast Fourier transforms, 2D-IFFTs)、两次 2D-FFTs 和频域上简单的分量式运算实现快速运算. 为了表示简洁, 这里令  $\mathbf{W}_{2d} = \mathbf{W}_{N_2} \otimes \mathbf{W}_{N_1}$  表示相应维度的 2D-FFTs 矩阵, 则 (8) 式可以简写为

$$\mathbf{F}_z = \mathbf{W}_{2d} \Delta_z \mathbf{W}_{2d}^{-1}, \quad (9)$$

将循环卷积运算从时域转化到频域是算法能够快速运算的关键. 为了更好地理解提出的模型, 需要指出的两点是:

1) 类似于传统压缩感知中多幅图像被同一组稀疏基或稀疏字典表示, 在这里  $N$  幅图像共同被  $K$  组卷积滤波器  $\tilde{\mathbf{F}}_z (z = 1, 2, \dots, K)$  与特征图的循环卷积表示;

2) 特征图  $\tilde{\mathbf{S}}$  没有标注下标  $z$  并不是表示每幅图像只对应一个特征图, 相反每幅图像都有特定的  $K$  个特征图, 每个特征图都有各自的后验分布, 如 (11) 和 (12) 式所示. 省去下标  $z$  的原因是所有特征图的先验分布相同.

### 3 从训练集中学习 convGMM

考虑从一组训练图像集  $\{\mathbf{X}_i\}_{i=1}^N$  中学习所提出的 convGMM, 记该模型中待估计的参数为  $\Theta = \{\pi_z, \boldsymbol{\mu}_z, \mathbf{f}_z\}_{z=1}^K$ .

在估计未知参数前, 先推导联合分布  $p(z, \mathbf{s}, \mathbf{x})$  和后验分布  $p(z, \mathbf{s} | \mathbf{x})$ .

$$\begin{aligned} & p(z, \mathbf{s}, \mathbf{x}) \\ &= p(z) p(\mathbf{s} | z) p(\mathbf{x} | z, \mathbf{s}) \\ &= \pi_z \mathcal{N}(\mathbf{s} | \mathbf{0}, \mathbf{I}) \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_z + \mathbf{F}_z \mathbf{s}, \gamma \mathbf{I}) \\ &= \pi_z \mathcal{N}\left(\begin{bmatrix} \mathbf{s} \\ \mathbf{x} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu}_z \end{bmatrix}, \begin{bmatrix} \mathbf{I} & \mathbf{F}'_z \\ \mathbf{F}_z & \gamma \mathbf{I} + \mathbf{F}_z \mathbf{F}'_z \end{bmatrix}\right). \end{aligned} \quad (10)$$

基于 (9) 式中  $\mathbf{F}_z$  的频域性质, 可得

$$\begin{aligned} & p(z, \mathbf{s}, \mathbf{x}) \\ &= \pi_z \mathcal{N}\left(\begin{bmatrix} \mathbf{s} \\ \mathbf{x} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu}_z \end{bmatrix}, \begin{bmatrix} \mathbf{I} & \mathbf{W}_{2d} \bar{\Delta}_z \mathbf{W}_{2d}^{-1} \\ \mathbf{W}_{2d} \Delta_z \mathbf{W}_{2d}^{-1} & \mathbf{W}_{2d} (\gamma \mathbf{I} + |\Delta_z|^2) \mathbf{W}_{2d}^{-1} \end{bmatrix}\right) \\ &= \pi_z \mathcal{N}(\mathbf{s} | \boldsymbol{\xi}_z, \mathbf{Q}_z) \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_z, \mathbf{R}_z), \end{aligned} \quad (11)$$

其中,  $\bar{\Delta}_z$  表示  $\Delta_z$  的共轭,  $|\Delta_z|^2 = \text{diag}(|\hat{\mathbf{f}}_z|^2)$ , (11) 式最后一行的参数满足

$$\boldsymbol{\xi}_z = \mathbf{W}_{2d} \left[ \bar{\Delta}_z (\gamma \mathbf{I} + |\Delta_z|^2)^{-1} \right] \mathbf{W}_{2d}^{-1} (\mathbf{x} - \boldsymbol{\mu}_z), \quad (12)$$

$$\mathbf{Q}_z = \gamma \mathbf{W}_{2d} (\gamma \mathbf{I} + |\Delta_z|^2)^{-1} \mathbf{W}_{2d}^{-1}, \quad (13)$$

$$\mathbf{R}_z = \mathbf{W}_{2d} (\gamma \mathbf{I} + |\Delta_z|^2) \mathbf{W}_{2d}^{-1}, \quad (14)$$

上述这些参数均可利用 2D-FFTs 和 2D-IFFTs 实现快速运算. 由 (11) 式可得信号  $\mathbf{x}$  的边缘分布

$$p(\mathbf{x} | \Theta) = \sum_{z=1}^K \pi_z \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_z, \mathbf{R}_z) \quad (15)$$

和给定  $\mathbf{x}$  时的  $(z, \mathbf{s})$  的后验分布

$$p(z, \mathbf{s} | \mathbf{x}, \Theta) = \rho_z \mathcal{N}(\mathbf{s} | \boldsymbol{\xi}_z, \mathbf{Q}_z), \quad (16)$$

其中,

$$\rho_z = \frac{\pi_z \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_z, \mathbf{R}_z)}{\sum_{l=1}^K \pi_l \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_l, \mathbf{R}_l)}. \quad (17)$$

本文将未知变量  $\{z_i, \mathbf{s}_i\}$  看作隐变量, 基于 MMLE 从训练数据集中学习 convGMM:

$$\begin{aligned} \Theta_{\text{MMLE}} &= \max_{\Theta} \sum_{i=1}^N \ln p(\mathbf{x}_i | \Theta) \\ &= \max_{\Theta} \sum_{i=1}^N \ln \sum_{z_i=1}^K \int p(z_i, \mathbf{s}_i, \mathbf{x}_i | \Theta) d\mathbf{s}_i, \end{aligned} \quad (18)$$

其中, 联合分布  $p(z_i, \mathbf{s}_i, \mathbf{x}_i | \Theta)$  由 (10) 式给出. 通过 EM 算法<sup>[19]</sup> 求解上述的目标函数, 具体步骤如下.

1) E-step: 基于上述公式 (16) 的后验分布  $p(z_i, \mathbf{s}_i | \mathbf{x}_i, \Theta^{(t-1)})$ , 求解对数似然函数  $\ln p(z_i, \mathbf{s}_i, \mathbf{x}_i | \Theta)$  的期望.

$$\begin{aligned}
 & \ell(\Theta | \Theta^{(t-1)}) \\
 &= \sum_{i=1}^N \mathbb{E}_{z_i, \mathbf{s}_i | \mathbf{x}_i, \Theta^{(t-1)}} \{ \ln p(z_i, \mathbf{s}_i, \mathbf{x}_i | \Theta) \} \\
 &= \sum_{i=1}^N \mathbb{E}_{z_i, \mathbf{s}_i | \mathbf{x}_i, \Theta^{(t-1)}} \{ \ln \pi_z \mathcal{N}(\mathbf{s}_i | \mathbf{0}, \mathbf{I}) \\
 & \quad \cdot \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_z + \mathbf{F}_z \mathbf{s}_i, \gamma \mathbf{I}) \} \\
 &= \text{constant} + \sum_{i=1}^N \sum_{z=1}^K \rho_{iz}^{(t-1)} \ln \pi_z \\
 & \quad - \frac{1}{2} \sum_{i=1}^N \mathbb{E} \left[ (\mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \mathbf{s}_i)' (\gamma \mathbf{I})^{-1} \right. \\
 & \quad \left. \cdot (\mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \mathbf{s}_i) \right], \tag{19}
 \end{aligned}$$

其中, constant 表示与参数  $\Theta$  无关项的和,  $\rho_{iz}^{(t-1)} = \rho_z(\mathbf{x}_i, \Theta^{(t-1)})$ . 为了表示简洁, 这里省略了期望  $\mathbb{E}_{z_i, \mathbf{s}_i | \mathbf{x}_i, \Theta^{(t-1)}}(\cdot)$  的下标.

从直观上看, 接下来可以直接推导 (19) 式最后一行的期望,  $\mathbb{E} \left[ (\mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \mathbf{s}_i)' (\gamma \mathbf{I})^{-1} \cdot (\mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \mathbf{s}_i) \right]$ , 从而完成整个期望的计算. 但在卷积运算转化为乘积运算的过程中,  $\text{vec}(\tilde{\mathbf{F}}_z * \tilde{\mathbf{S}}) = \mathbf{F}_z \mathbf{s}$ , 大的卷积矩阵  $\mathbf{F}_z \in \mathbb{R}^{(N_1 \cdot N_2) \times (N_1 \cdot N_2)}$  是由小的卷积滤波器  $\tilde{\mathbf{F}}_z \in \mathbb{R}^{n_1 \times n_2}$  按照循环卷积的对应关系生成的, 且  $\mathbf{F}_z$  中大部分元素都是 0. 所以在接下来 EM 算法的 M-step 中, 本文不能直接更新  $\mathbf{F}_z$ , 而应更新滤波器  $\tilde{\mathbf{F}}_z$ . 在本步中需要将 (19) 式最后一行的  $\mathbf{F}_z$  转化为用  $\tilde{\mathbf{F}}_z$  (或  $\mathbf{f}_z$ , 因为  $\mathbf{f}_z = \text{vec}(\tilde{\mathbf{F}}_z)$ ) 来表示. 为了使后续的结论更加具有普适性, 这里将协方差矩阵  $\gamma \mathbf{I}$  拓展为任意矩阵  $\Gamma$  的一般情形. (19) 式最后一行可进一步化简为

$$\begin{aligned}
 & - \frac{1}{2} \sum_{i=1}^N \mathbb{E} \left[ (\mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \mathbf{s}_i)' \Gamma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \mathbf{s}_i) \right] \\
 &= - \frac{1}{2} \sum_{i=1}^N \mathbb{E} \left[ \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} + \boldsymbol{\xi}_{iz}^{(t-1)} \right) \right)' \right. \\
 & \quad \left. \cdot \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} + \boldsymbol{\xi}_{iz}^{(t-1)} \right) \right) \right] \\
 &= - \frac{1}{2} \sum_{i=1}^N \mathbb{E} \left[ \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} - \mathbf{F}_z \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} \right) \right)' \right. \\
 & \quad \left. \cdot \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} - \mathbf{F}_z \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} \right) \right) \right] \\
 &= - \frac{1}{2} \sum_{i=1}^N \mathbb{E} \left[ \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right)' \right. \\
 & \quad \cdot \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right) - 2 \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right)' \\
 & \quad \cdot \Gamma^{-1} \mathbf{F}_z \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} \right) + \mathbf{s} \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} \right)' \\
 & \quad \left. \cdot \mathbf{F}_z' \Gamma^{-1} \mathbf{F}_z \left( \mathbf{s}_i - \boldsymbol{\xi}_{iz}^{(t-1)} \right) \right] \\
 &= - \frac{1}{2} \sum_{i=1}^N \sum_{z=1}^K \rho_{iz}^{(t-1)} \times \left[ \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right)' \right. \\
 & \quad \left. \cdot \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right) + \text{tr} \left( \mathbf{Q}_z^{(t-1)} \mathbf{F}_z' \Gamma^{-1} \mathbf{F}_z \right) \right]. \tag{20}
 \end{aligned}$$

(20) 式的最后一行的两项都含有卷积矩阵  $\mathbf{F}_z$ , 现将它们转化成用卷积滤波器  $\mathbf{f}_z$  来表示:

$$\begin{aligned}
 & \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right)' \\
 & \quad \times \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right) \\
 &= \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \text{vec} \left( \tilde{\mathbf{F}}_z * \tilde{\boldsymbol{\Xi}}_{iz}^{(t-1)} \right) \right)' \\
 & \quad \cdot \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \text{vec} \left( \tilde{\mathbf{F}}_z * \tilde{\boldsymbol{\Xi}}_{iz}^{(t-1)} \right) \right) \\
 &= \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z \right)' \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z \right), \tag{21}
 \end{aligned}$$

其中,  $\boldsymbol{\Xi}_{iz}^{(t-1)}$  是满足  $\mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} = \text{vec}(\tilde{\mathbf{F}}_z * \tilde{\boldsymbol{\Xi}}_{iz}^{(t-1)}) = \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z$  关系的卷积矩阵, 类似于 (6) 式中卷积运算到乘积运算的转化. 若考虑特殊情形  $\Gamma = \gamma \mathbf{I}$ , 则

$$\begin{aligned}
 & \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right)' \Gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \mathbf{F}_z \boldsymbol{\xi}_{iz}^{(t-1)} \right) \\
 &= \gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z \right)' \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z \right). \tag{22}
 \end{aligned}$$

(20) 式最后一行的第二项为

$$\begin{aligned} & \text{tr} \left( \mathbf{Q}_z^{(t-1)} \mathbf{F}'_z \mathbf{\Gamma}^{-1} \mathbf{F}_z \right) \\ &= \text{tr} \left( \mathbf{\Gamma}^{-1} \mathbf{F}_z \mathbf{Q}_z^{(t-1)} \mathbf{F}'_z \right) \\ &= \sum_j \sum_l [\mathbf{\Gamma}^{-1}]_{j,l} \left( \mathbf{F}_z \mathbf{Q}_z^{(t-1)} \mathbf{F}'_z \right)_{l,j} \\ &= \sum_j \sum_l [\mathbf{\Gamma}^{-1}]_{j,l} \left( [\mathbf{F}_z]_{l,:} \mathbf{Q}_z^{(t-1)} [\mathbf{F}'_z]_{:,j} \right) \\ &= \sum_j \sum_l [\mathbf{\Gamma}^{-1}]_{j,l} \mathbf{f}'_z \left[ \mathbf{Q}_z^{(t-1)} \right]_{\{l\},\{j\}} \mathbf{f}_z \\ &= \mathbf{f}'_z \left( \sum_j \sum_l [\mathbf{\Gamma}^{-1}]_{j,l} \left[ \mathbf{Q}_z^{(t-1)} \right]_{\{l\},\{j\}} \right) \mathbf{f}_z, \quad (23) \end{aligned}$$

如果  $\mathbf{\Gamma} = \gamma \mathbf{I}$ , 则

$$\begin{aligned} & \text{tr} \left( \mathbf{Q}_z^{(t-1)} \mathbf{F}'_z \mathbf{\Gamma}^{-1} \mathbf{F}_z \right) \\ &= \gamma^{-1} \mathbf{f}'_z \left( \sum_j \left[ \mathbf{Q}_z^{(t-1)} \right]_{\{j\},\{j\}} \right) \mathbf{f}_z \\ &= \gamma^{-1} \mathbf{f}'_z \boldsymbol{\Psi}_z^{(t-1)} \mathbf{f}_z, \quad (24) \end{aligned}$$

其中,  $[\cdot]_{l,:}$  和  $[\cdot]_{:,j}$  分别表示矩阵的第  $l$  行和第  $j$  列;  $\left[ \mathbf{Q}_z^{(t-1)} \right]_{\{l\},\{j\}}$  是矩阵  $\mathbf{Q}_z^{(t-1)}$  沿着主对角线的子矩阵, 它的行索引  $\{l\}$  是由  $\mathbf{f}'_z$  在行向量  $[\mathbf{F}_z]_{l,:}$  中的位置

决定, 列索引  $\{j\}$  是由  $\mathbf{f}_z$  在列向量  $[\mathbf{F}'_z]_{:,j}$  中的位置决定;  $\boldsymbol{\Psi}_z^{(t-1)} = \sum_j \left[ \mathbf{Q}_z^{(t-1)} \right]_{\{j\},\{j\}}$ .

将 (22) 和 (24) 式代入 (19) 和 (20) 式中, 可得对数似然函数的期望为

$$\begin{aligned} & ll \left( \boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t-1)} \right) \\ &= \text{constant} + \sum_{i=1}^N \sum_{z=1}^K \rho_{iz}^{(t-1)} \ln \pi_z \\ &\quad - \frac{1}{2} \sum_{i=1}^N \sum_{z=1}^K \rho_{iz}^{(t-1)} \times \left[ \gamma^{-1} \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z \right)' \right. \\ &\quad \cdot \left. \left( \mathbf{x}_i - \boldsymbol{\mu}_z - \boldsymbol{\Xi}_{iz}^{(t-1)} \mathbf{f}_z \right) \right] \\ &\quad - \frac{1}{2} \sum_{i=1}^N \sum_{z=1}^K \rho_{iz}^{(t-1)} \times \left( \gamma^{-1} \mathbf{f}'_z \boldsymbol{\Psi}_z^{(t-1)} \mathbf{f}_z \right). \quad (25) \end{aligned}$$

2) M-step: 通过极大化 (25) 式来更新参数  $\boldsymbol{\theta}^{(t)}$ ,  $\boldsymbol{\theta}^{(t)} = \arg \max_{\boldsymbol{\theta}} ll \left( \boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t-1)} \right)$ , 可得

$$\pi_z^{(t)} = \frac{\sum_{i=1}^N \rho_{iz}^{(t-1)}}{\sum_{z=1}^K \sum_{i=1}^N \rho_{iz}^{(t-1)}}, \quad (26)$$

$$\begin{bmatrix} \boldsymbol{\mu}_z^{(t)} \\ \mathbf{f}_z^{(t)} \end{bmatrix} = \left( \sum_{i=1}^N \rho_{iz}^{(t-1)} \cdot \left[ \begin{array}{c} \mathbf{I} \\ \left( \boldsymbol{\Xi}_{iz}^{(t-1)} \right)' \boldsymbol{\Psi}_z^{(t-1)} + \left( \boldsymbol{\Xi}_{iz}^{(t-1)} \right)' \boldsymbol{\Xi}_{iz}^{(t-1)} \end{array} \right] \right)^{-1} \times \left( \sum_{i=1}^N \rho_{iz}^{(t-1)} \left[ \begin{array}{c} \mathbf{x}_i \\ \left( \boldsymbol{\Xi}_{iz}^{(t-1)} \right)' \mathbf{x}_i \end{array} \right] \right). \quad (27)$$

在 (12)—(14) 式和 (26), (27) 式之间交替迭代组成了整个 MMLE-convGMM 算法的迭代过程. 根据 EM 算法的性质 [19], 边缘似然函数  $\sum_{i=1}^N \ln \sum_{z=1}^K \pi_z^{(t)} \mathcal{N} \left( \mathbf{x}_i \mid \boldsymbol{\mu}_z^{(t)}, \mathbf{R}_z \left( \boldsymbol{\theta}^{(t)} \right) \right)$  将随迭代次数逐渐增加直至收敛.

#### 4 基于 convGMM 的压缩测量

在压缩测量过程中, 本文考虑将原图像与随机核矩阵进行循环卷积, 再依据采样率对循环卷积的结果进行降采样. 该过程的模型为

$$\mathbf{Y} = \mathcal{P}_{\Omega} \left( \tilde{\boldsymbol{\Phi}} * \mathbf{X} \right) + \mathbf{N}, \quad \mathbf{N} \sim \mathcal{N} \left( \mathbf{0}, \sigma^2 \mathbf{I} \right), \quad (28)$$

其中,  $\tilde{\boldsymbol{\Phi}}$  表示随机卷积核, 它与图像的大小相同;

$\mathcal{P}_{\Omega}$  表示在索引集  $\Omega$  下的降采样算子;  $\mathbf{Y}$  表示循环压缩测量结果;  $\mathbf{N}$  表示噪声. 现将 (28) 式向量化:

$$\mathbf{y} = \mathbf{P}_{\Omega} \boldsymbol{\Phi} \mathbf{x} + \mathbf{n}, \quad (29)$$

其中,  $\mathbf{P}_{\Omega}$  是降采样算子  $\mathcal{P}_{\Omega}$  对应的降采样矩阵,  $\boldsymbol{\Phi}$  是由卷积核  $\tilde{\boldsymbol{\Phi}}$  生成的有着循环块的块循环矩阵, 其结构如 (7) 式所示, 故  $\boldsymbol{\Phi}$  可表示为

$$\boldsymbol{\Phi} = \mathbf{W}_{2d} \boldsymbol{\Delta}_{CS} \mathbf{W}_{2d}^{-1}, \quad (30)$$

其中,  $\boldsymbol{\Delta}_{CS} = \text{diag} \left( \mathbf{W}_{2d} \tilde{\boldsymbol{\Phi}} \right)$ ,  $\tilde{\boldsymbol{\phi}} = \text{vec} \left( \tilde{\boldsymbol{\Phi}} \right)$ .

基于 convGMM 的压缩测量如图 1 所示, 本文的目标是基于极大后验估计, 从压缩测量结果  $\mathbf{y}$  中恢复原信号  $\mathbf{x}$ .

$(z, \mathbf{s}, \mathbf{x}, \mathbf{y})$  的联合分布为



$$\begin{aligned}
 & p(z, \mathbf{s}, \mathbf{x}, \mathbf{y}) \\
 &= p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{s}, z)p(\mathbf{s})p(z) \\
 &= \pi_z \mathcal{N}(\mathbf{y} | P_\Omega \Phi \mathbf{x}, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{x} | \mu_z + F_z \mathbf{s}, \gamma \mathbf{I}) \mathcal{N}(\mathbf{s} | \mathbf{0}, \mathbf{I}) \\
 &= \pi_z \mathcal{N} \left( \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \\ \mathbf{s} \end{bmatrix} \middle| \begin{bmatrix} P_\Omega \Phi \mu_z \\ \mu_z \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \sigma^{-2} \mathbf{I} & & \\ -\sigma^{-2} \Phi' P'_\Omega & \sigma^{-2} \Phi' P'_\Omega P_\Omega \Phi + \gamma^{-1} \mathbf{I} & \\ \mathbf{0} & -\gamma^{-1} F'_z & \mathbf{I} + \gamma^{-1} F'_z F_z \end{bmatrix}^{-1} \right). \quad (31)
 \end{aligned}$$

根据贝叶斯理论,  $(z, \mathbf{s}, \mathbf{x}, \mathbf{y})$  的联合分布还可写为

$$\begin{aligned}
 & p(z, \mathbf{s}, \mathbf{x}, \mathbf{y}) \\
 &= p(\mathbf{x}, \mathbf{s} | \mathbf{y}, z) p(\mathbf{y} | z) p(z) \\
 &= \pi_z \mathcal{N} \left( \begin{bmatrix} \mathbf{x} \\ \mathbf{s} \end{bmatrix} \middle| \begin{bmatrix} \eta_z \\ \xi_z \end{bmatrix}, \begin{bmatrix} C_z & \Omega_z \\ \Omega'_z & Q_z \end{bmatrix} \right) \\
 & \cdot \mathcal{N}(\mathbf{y} | \Phi \mu_z, \mathbf{R}_z), \quad (32)
 \end{aligned}$$

其中的参数满足

$$\begin{aligned}
 & \begin{bmatrix} C_z & \Omega_z \\ \Omega'_z & Q_z \end{bmatrix} \\
 &= \begin{bmatrix} \sigma^{-2} \Phi' P'_\Omega P_\Omega \Phi + \gamma^{-1} \mathbf{I} & -\gamma^{-1} F'_z \\ -\gamma^{-1} F'_z & \mathbf{I} + \gamma^{-1} F'_z F_z \end{bmatrix}^{-1}, \quad (33)
 \end{aligned}$$

$$\begin{aligned}
 & \begin{bmatrix} \eta_z \\ \xi_z \end{bmatrix} = \begin{bmatrix} \mu_z \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} C_z & \Omega_z \\ \Omega'_z & Q_z \end{bmatrix} \\
 & \times \begin{bmatrix} -\sigma^{-2} \Phi' P'_\Omega \\ \mathbf{0} \end{bmatrix} (\mathbf{y} - P_\Omega \Phi \mu_z), \quad (34)
 \end{aligned}$$

$$\begin{aligned}
 & \mathbf{R}_z = \left( \sigma^{-2} \mathbf{I} - \begin{bmatrix} -\sigma^{-2} P_\Omega \Phi & \mathbf{0} \end{bmatrix} \begin{bmatrix} C_z & \Omega_z \\ \Omega'_z & Q_z \end{bmatrix} \right. \\
 & \left. \times \begin{bmatrix} -\sigma^{-2} \Phi' P'_\Omega \\ \mathbf{0} \end{bmatrix} \right)^{-1}, \quad (35)
 \end{aligned}$$

化简可得

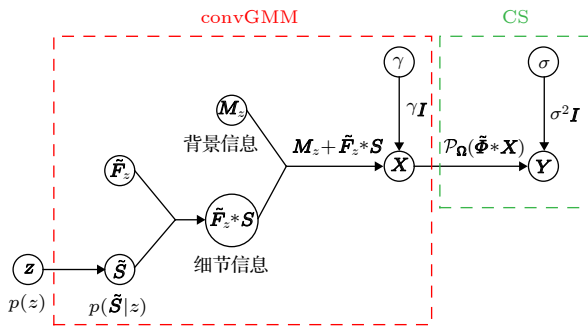


图 1 基于 convGMM 的压缩测量

Fig. 1. Structure of convGMM with application to compressive sensing.

$$\begin{aligned}
 & \mathbf{R}_z = \sigma^2 \mathbf{I} + P_\Omega \Phi (\gamma \mathbf{I} + F_z F'_z) \Phi' P'_\Omega \\
 &= P_\Omega \mathbf{W}_{2d} \left[ \sigma^2 \mathbf{I} + |\Delta_{CS}|^2 (\gamma \mathbf{I} + |\Delta_z|^2) \right] \mathbf{W}_{2d}^{-1} P'_\Omega, \quad (36)
 \end{aligned}$$

$$\begin{aligned}
 & C_z = (\gamma \mathbf{I} + F_z F'_z) - (\gamma \mathbf{I} + F_z F'_z) \Phi' P'_\Omega \mathbf{R}_z^{-1} \\
 & \quad \times P_\Omega \Phi (\gamma \mathbf{I} + F_z F'_z) \\
 &= \mathbf{W}_{2d} \left( \gamma \mathbf{I} + |\Delta_z|^2 \right) \mathbf{W}_{2d}^{-1} \\
 & \quad - \left[ \mathbf{W}_{2d} \left( (\gamma \mathbf{I} + |\Delta_z|^2) \bar{\Delta}_{CS} \right) \mathbf{W}_{2d}^{-1} \right] \\
 & \quad \times (P'_\Omega \mathbf{R}_z^{-1} P_\Omega) \\
 & \quad \times \left[ \mathbf{W}_{2d} \left( (\gamma \mathbf{I} + |\Delta_z|^2) \Delta_{CS} \right) \mathbf{W}_{2d}^{-1} \right], \quad (37)
 \end{aligned}$$

$$\begin{aligned}
 & Q_z = \mathbf{I} - F'_z \Phi' P'_\Omega \mathbf{R}_z^{-1} P_\Omega \Phi F_z \\
 &= \mathbf{I} - \left[ \mathbf{W}_{2d} (\bar{\Delta}_z \bar{\Delta}_{CS}) \mathbf{W}_{2d}^{-1} \right] \times (P'_\Omega \mathbf{R}_z^{-1} P_\Omega) \\
 & \quad \times \left[ \mathbf{W}_{2d} (\Delta_z \Delta_{CS}) \mathbf{W}_{2d}^{-1} \right], \quad (38)
 \end{aligned}$$

$$\begin{aligned}
 & \Omega_z = F_z - (\gamma \mathbf{I} + F_z F'_z) \Phi' P'_\Omega \mathbf{R}_z^{-1} P_\Omega \Phi F_z \\
 &= \mathbf{W}_{2d} \Delta_z \mathbf{W}_{2d}^{-1} - \left[ \mathbf{W}_{2d} \left( (\gamma \mathbf{I} + |\Delta_z|^2) \bar{\Delta}_{CS} \right) \mathbf{W}_{2d}^{-1} \right] \\
 & \quad \times (P'_\Omega \mathbf{R}_z^{-1} P_\Omega) \times \left[ \mathbf{W}_{2d} (\Delta_z \Delta_{CS}) \mathbf{W}_{2d}^{-1} \right], \quad (39)
 \end{aligned}$$

$$\begin{aligned}
 & \eta_z = \mu_z + (\gamma \mathbf{I} + F_z F'_z) \Phi' P'_\Omega \mathbf{R}_z^{-1} (\mathbf{y} - P_\Omega \Phi \mu_z) \\
 &= \mu_z + \left[ \mathbf{W}_{2d} \left( (\gamma \mathbf{I} + |\Delta_z|^2) \bar{\Delta}_{CS} \right) \mathbf{W}_{2d}^{-1} \right] \\
 & \quad \times P'_\Omega \mathbf{R}_z^{-1} \cdot (\mathbf{y} - P_\Omega \mathbf{W}_{2d} \Delta_{CS} \mathbf{W}_{2d}^{-1} \mu_z), \quad (40)
 \end{aligned}$$

$$\begin{aligned}
 & \xi_z = F'_z \Phi' P'_\Omega \mathbf{R}_z^{-1} (\mathbf{y} - P_\Omega \Phi \mu_z) \\
 &= \left[ \mathbf{W}_{2d} (\bar{\Delta}_z \bar{\Delta}_{CS}) \mathbf{W}_{2d}^{-1} \right] \times P'_\Omega \mathbf{R}_z^{-1} \\
 & \quad \times (\mathbf{y} - P_\Omega \mathbf{W}_{2d} \Delta_{CS} \mathbf{W}_{2d}^{-1} \mu_z), \quad (41)
 \end{aligned}$$

其中,  $\bar{\Delta}_{CS}$  表示  $\Delta_{CS}$  的共轭. 值得注意的是上述这些参数均可利用 2D-FFTs 实现快速运算.

从 (32) 式的最后一行可得  $\mathbf{y}$  的边缘分布

$$p(\mathbf{y}|\Theta) = \sum_{z=1}^K \pi_z \mathcal{N}(\mathbf{y} | \mathbf{P}_\Omega \Phi \mu_z, \mathbf{R}_z), \quad (42)$$

以及给定  $\mathbf{y}$  时,  $(z, \mathbf{s}, \mathbf{x})$  的后验分布

$$p(z, \mathbf{s}, \mathbf{x} | \mathbf{y}, \Theta) = \rho_z(\mathbf{y}, \Theta) \mathcal{N}\left(\left[\begin{array}{c} \mathbf{x} \\ \mathbf{s} \end{array}\right] \middle| \left[\begin{array}{c} \boldsymbol{\eta}_z \\ \boldsymbol{\xi}_z \end{array}\right], \left[\begin{array}{cc} \mathbf{C}_z & \boldsymbol{\Omega}_z \\ \boldsymbol{\Omega}'_z & \mathbf{Q}_z \end{array}\right]\right), \quad (43)$$

其中,

$$\rho_z(\mathbf{y}, \Theta) = \frac{\pi_z \mathcal{N}(\mathbf{y} | \mathbf{P}_\Omega \Phi \mu_z, \mathbf{R}_z)}{\sum_{l=1}^K \pi_l \mathcal{N}(\mathbf{y} | \mathbf{P}_\Omega \Phi \mu_l, \mathbf{R}_l)}. \quad (44)$$

由 (43) 式可得信号  $\mathbf{x}$  的后验分布为

$$p(\mathbf{x} | \mathbf{y}, \Theta) = \sum_{z=1}^K \rho_z(\mathbf{y}, \Theta) \mathcal{N}(\mathbf{x} | \boldsymbol{\eta}_z(\mathbf{y}, \Theta), \mathbf{C}_z(\mathbf{y}, \Theta)), \quad (45)$$

可以看出信号  $\mathbf{x}$  的后验分布也是一个 GMM, 各高斯分布的期望、协方差矩阵和权重都与压缩测量结果以及未知参数有关.

令  $\hat{\mathbf{x}}$  表示从压缩测量结果  $\mathbf{y}$  中恢复的信号, 这里采用均方误差 (mean square error, MSE) 作为恢复算法的评价指标, 则

$$MSE = \iint \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 p(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} d\mathbf{y}, \quad (46)$$

令  $\frac{\partial MSE}{\partial \hat{\mathbf{x}}} = 0$ , 可得

$$\hat{\mathbf{x}} = \int \mathbf{x} p(\mathbf{x} | \mathbf{y}) \, d\mathbf{x} = \mathbb{E}_{\mathbf{x} | \mathbf{y}, \Theta}(\mathbf{x}), \quad (47)$$

由此可得信号  $\mathbf{x}$  后验分布的期望  $\mathbb{E}_{\mathbf{x} | \mathbf{y}, \Theta}(\mathbf{x})$  恰好是最小均方误差 (minimum mean square error, MMSE) 意义下的估计<sup>[20]</sup>, 即

$$\begin{aligned} \hat{\mathbf{x}}_{\text{MMSE}}(\mathbf{y}, \Theta) &= \mathbb{E}_{\mathbf{x} | \mathbf{y}, \Theta}(\mathbf{x}) \\ &= \sum_{z=1}^K \rho_z(\mathbf{y}, \Theta) \cdot \boldsymbol{\eta}_z(\mathbf{y}, \Theta). \end{aligned} \quad (48)$$

## 5 计算复杂度分析

由于本文所提的 convGMM 对整幅图像的概率分布进行建模, 因而存在矩阵维数大, 计算复杂的问题, 为了提高 MMLE-convGMM 算法运算的效率, 本文做了如下两个重要的降低计算复杂度的工作.

1) 在 convGMM 中采用循环卷积 ((2) 和

(3) 式), 因而循环卷积运算  $\mathbf{F}_z \mathbf{s} = \text{vec}(\tilde{\mathbf{F}}_z * \tilde{\mathbf{S}})$  可以利用有着循环块的块循环矩阵的数学性质, 基于 (8) 式中的一次 2D-IFFTs、两次 2D-FFTs 和频域上简单的分量式运算实现快速运算. 计算复杂度从  $O((N_1 \cdot N_2)^2)$  减小到  $O((N_1 \cdot N_2) \log_2(N_1 \cdot N_2))$ .

更重要的是, 在第 3 节 convGMM 模型的训练中, (12)—(14) 式中参数的计算, 以及第 4 节从压缩测量结果  $\mathbf{y}$  中恢复原信号  $\mathbf{x}$ , (48) 式中参数的计算, 均可利用 (8) 式中的 2D-FFTs 和 2D-IFFTs 实现快速运算.

2) 在 MMLE-convGMM 算法的迭代过程中, (27) 式中矩阵求逆的计算复杂度较大. 令

$$\begin{aligned} \mathbf{A} &= \sum_{i=1}^N \rho_{iz}^{(t-1)} \\ &\times \left[ \begin{array}{cc} \mathbf{I} & \boldsymbol{\Xi}_{iz}^{(t-1)} \\ \left(\boldsymbol{\Xi}_{iz}^{(t-1)}\right)' & \boldsymbol{\Psi}_z^{(t-1)} + \left(\boldsymbol{\Xi}_{iz}^{(t-1)}\right)' \boldsymbol{\Xi}_{iz}^{(t-1)} \end{array} \right] \\ &= \left[ \begin{array}{cc} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{array} \right], \end{aligned} \quad (49)$$

其中,

$$\begin{aligned} \mathbf{A}_{11} &= \sum_{i=1}^N \rho_{iz}^{(t-1)} \mathbf{I}, \\ \mathbf{A}_{12} &= \sum_{i=1}^N \rho_{iz}^{(t-1)} \boldsymbol{\Xi}_{iz}^{(t-1)}, \\ \mathbf{A}_{21} &= \sum_{i=1}^N \rho_{iz}^{(t-1)} \left(\boldsymbol{\Xi}_{iz}^{(t-1)}\right)', \end{aligned}$$

$$\mathbf{A}_{22} = \sum_{i=1}^N \rho_{iz}^{(t-1)} \left( \boldsymbol{\Psi}_z^{(t-1)} + \left(\boldsymbol{\Xi}_{iz}^{(t-1)}\right)' \boldsymbol{\Xi}_{iz}^{(t-1)} \right).$$

由于矩阵  $\boldsymbol{\Xi}_{iz}^{(t-1)}$  的大小为  $(N_1 \cdot N_2) \times (n_1 \cdot n_2)$ , 则矩阵  $\mathbf{A}$  的大小为  $(N_1 \cdot N_2 + n_1 \cdot n_2) \times (N_1 \cdot N_2 + n_1 \cdot n_2)$ . 如果直接对 (49) 式求逆, 则利用 Cholesky 分解, 其计算复杂度为

$$O\left((N_1 \cdot N_2 + n_1 \cdot n_2)^3 / 3 + (N_1 \cdot N_2 + n_1 \cdot n_2)^2\right).$$

为了降低矩阵  $\mathbf{A}$  求逆的复杂度, 本文考虑分块矩阵求逆

$$\begin{aligned} \mathbf{A}^{-1} &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{M} & -\mathbf{M}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M} & \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{M}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \end{bmatrix}, \end{aligned} \quad (50)$$

其中,

$$\mathbf{M} = (\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1}, \quad (51)$$

再根据 Woodbury 矩阵恒等式, 将矩阵  $\mathbf{M}$  化简为

$$\mathbf{M} = \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1}. \quad (52)$$

由 (50) 和 (52) 式可以看出, 计算  $\mathbf{A}^{-1}$  的主要计算量在于求  $\mathbf{A}_{22}^{-1}$ ,  $(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1}$  和  $\mathbf{A}_{11}^{-1}$ . 由于矩阵  $\mathbf{A}_{11}$  是对角矩阵,  $\mathbf{A}_{11}^{-1}$  的计算量可以忽略. 矩阵  $\mathbf{A}_{22}^{-1}$  和  $(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1}$  的大小都为  $(n_1 \cdot n_2) \times (n_1 \cdot n_2)$ , 他们求逆的复杂度为  $O((n_1 \cdot n_2)^3/3 + (n_1 \cdot n_2)^2)$ . 在卷积网络中滤波器的大小  $\tilde{\mathbf{F}}_z \in \mathbb{R}^{n_1 \times n_2}$  远小于图像的大小  $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}$ . 利用上述的矩阵分块求逆和 Woodbury 矩阵恒等式, 矩阵  $\mathbf{A}$  求逆的计算复杂度从  $O((N_1 \cdot N_2 + n_1 \cdot n_2)^3/3 + (N_1 \cdot N_2 + n_1 \cdot n_2)^2)$  降低到  $O((n_1 \cdot n_2)^3/3 + (n_1 \cdot n_2)^2)$ .

## 6 仿真结果及分析

本节将在三个标准数据集上验证 MMLE-convGMM 算法的有效性. 此外, 还与其他压缩感知算法相比较, 包括基于 GMM 的 MMLE-GMM 算法 [6], 贪婪算法中的正交匹配追踪算法 (orthogonal matching pursuit, OMP) [21], 凸优化算法中的 YALL1 算法 [22] 以及通过极小化  $l_{2,1}$  范数求解群基追踪 (group basis pursuit) 问题的广义交替投影 (generalized alternating projection, GAP) 算法 [23]. 每一种算法的恢复性能通过峰值信噪比 (peak signal to noise ratio, PSNR) 来评价. 对于 OMP, YALL1 和 GAP, 这里选取离散余弦变换 (discrete cosine transform, DCT) 矩阵作为稀疏基, 记为 “DCT-OMP”, “DCT-YALL1” 和 “DCT-GAP”. 此外, 文献 [24] 还提供了 KSVD-YALL1 算法用于整幅图像的恢复, 该算法首先利用 KSVD 算法在图像块上学习稀疏表示的字典, 再将图像块上学习的稀疏字典构建出整幅图像的稀疏字典, 因此本文也增加了 KSVD-YALL1 算法

作为比较.

### 6.1 CIFAR-10 数据集

首先在 CIFAR-10 数据集 [25] 上验证 MMLE-convGMM 算法的有效性. CIFAR-10 数据集是由 10 类  $32 \times 32$  大小的自然图像组成的数据集, 每一类图像有 60000 张, 其中 50000 张训练图像、10000 张测试图像. 这里随机从中选取了 3000 张图像 (每一类图像约 300 张) 并将其转化为灰度图像用于压缩感知. 基于第 4 节的压缩测量方法, 首先将每张图像与高斯核矩阵做循环卷积, 再对卷积的结果进行降采样. 采样率 (sampling rate) 定义为压缩采样数  $m$  与图像的像素  $n$  之比, 这里设定采样率从 0.05 增加到 0.4, 增加的步长为 0.05, 即  $m/n \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$ . 压缩测量噪声的标准差设为  $\sigma = 10^{-4}$ .

图 2 给出了在 CIFAR-10 图像中, 不同压缩感知算法恢复出图像的 PSNR 随采样率的变化情况, 可以看出 MMLE-convGMM 明显优于传统的压缩感知算法, MMLE-GMM 算法的性能次之. DCT-YALL1 与 DCT-GAP 两大类凸优化算法的恢复性能相当. 贪婪算法 DCT-OMP 的计算速度较快, 但其恢复性能比凸优化算法要差.

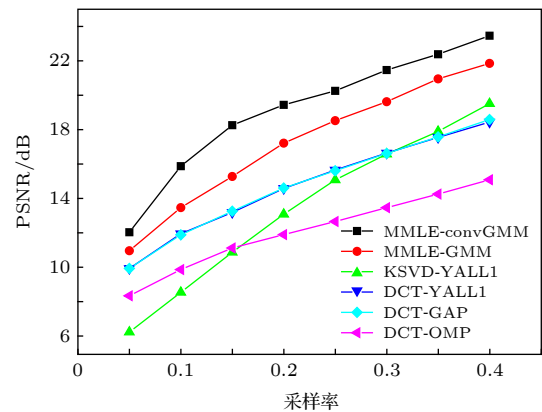


图 2 CIFAR-10 图像, 不同算法下恢复图像的 PSNR 随采样率的变化

Fig. 2. Averaged PSNR of reconstructed images from CIFAR-10 dataset as a function of sampling rate.

### 6.2 Caltech 101 数据集

为了进一步在大一些的图像数据集上测试 MMLE-convGMM 算法的有效性, 本节选取 Caltech 101 数据集 [26]. Caltech 101 数据集由 101 类自然图像所组成, 每一类图像有 40—800 张, 且每一张



图像的大小约为  $300 \times 200$ . 这里选取其中“飞机”图像, 共 800 张. 首先将它们转化为灰度图像, 再将其统一成  $128 \times 128$  的像素. 所有其他的设置, 包括采样率、测量矩阵、噪声水平同 CIFAR-10 数据集的仿真. 图 3 是不同压缩感知算法的恢复 PSNR 随着采样率的变化情况. 此外, 在图 4 中展示了随机选取的 12 张 Caltech 101“飞机”图像在不同算法下的恢复情况, 采样率设定为 0.4.

由图 3 和图 4 可以看出, 类似于 CIFAR-10 的仿真结果, MMLE-convGMM 算法明显优于其他的压缩感知恢复算法, 且在采样率为 0.4 时, 恢复图像的 PSNR 达到了 28 dB.

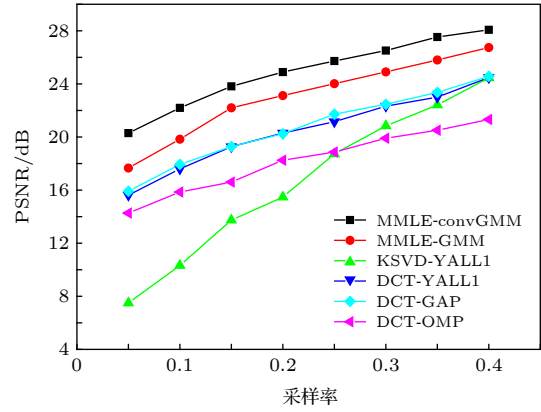


图 3 Caltech 101 图像, 不同算法下恢复图像的 PSNR 随采样率的变化

Fig. 3. Averaged PSNR of reconstructed images from Caltech 101 dataset as a function of sampling rate.

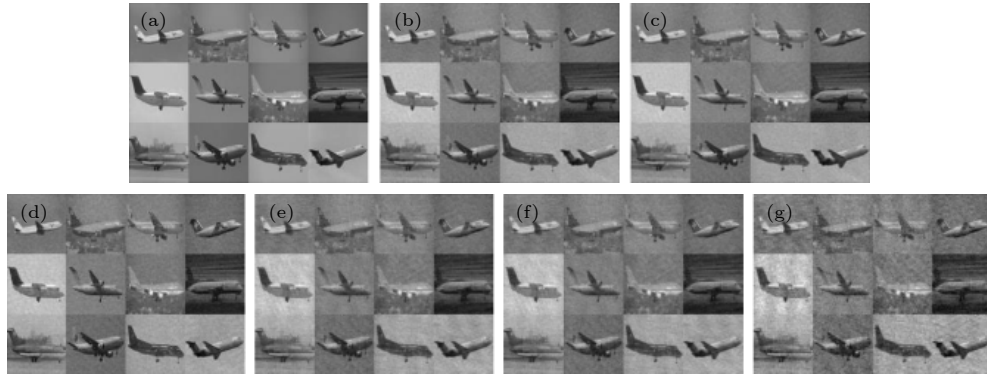


图 4 采样率为 0.4 时, 12 张 Caltech 101“飞机”图像在不同算法下的恢复情况 (a) 原图像; (b) MMLE-convGMM 下的恢复图像; (c) MMLE-GMM 下的恢复图像; (d) KSVD-YALL1 下的恢复图像; (e) DCT-YALL1 下的恢复图像; (f) DCT-GAP 下的恢复图像; (g) DCT-OMP 下的恢复图像

Fig. 4. Reconstructed performance comparison of 12 randomly selected “airplane” images from Caltech 101: (a) Original images; (b) images reconstructed by MMLE-convGMM; (c) images reconstructed by MLE-GMM; (d) images reconstructed by KSVD-YALL1; (e) images reconstructed by DCT-YALL1; (f) images reconstructed by DCT-GAP; (g) images reconstructed by DCT-OMP. All of the sampling rates are 0.4.

### 6.3 CelebA 数据集

最后在含有更大图像的 CelebA (Large-scale CelebFaces Attributes) 数据集 [27] 上验证 MMLE-convGMM 算法的有效性. CelebA 数据集含有超过 200000 张的名人图像, 每幅图像有 40 个属性注释. 该数据集中的图像包含了各种人物的姿势和背景. 这里将图像统一为  $256 \times 256$  的像素. 图 5 是 CelebA 图像的恢复 PSNR 随着采样率的变化情况. 图 6 展示了随机选取的 CelebA 图像的恢复情况, 采样率设定为 0.4, 可以看出 MMLE-convGMM 算法在大图像数据集上有着很好的恢复效果. 当采样率为 0.4 时, 恢复图像的 PSNR 为 30 dB.

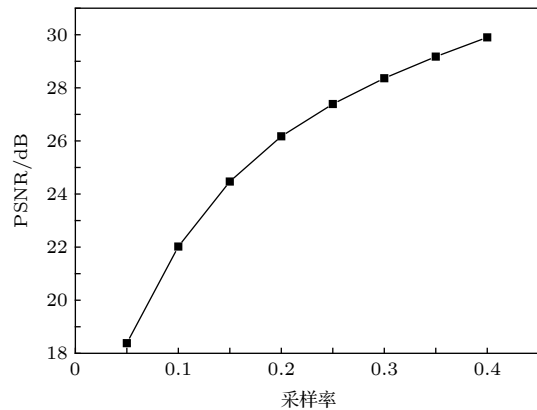


图 5 MMLE-convGMM 算法恢复 CelebA 图像的 PSNR 随采样率的变化

Fig. 5. Averaged PSNR of reconstructed images from CelebA dataset as a function of sampling rate by MMLE-convGMM.

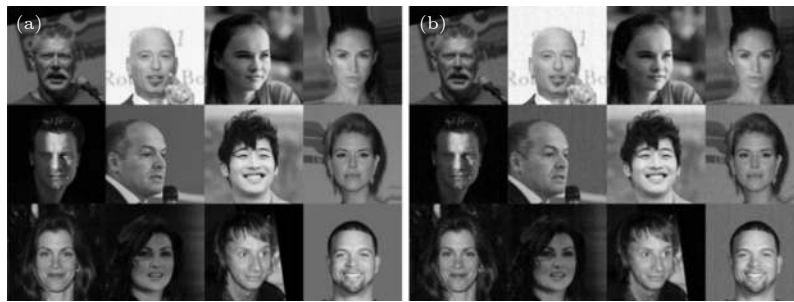


图 6 随机选取的 CelebA 图像的恢复情形 (a) 原图像; (b) MMLE-convGMM 算法恢复的图像, 采样率为 0.4

Fig. 6. Reconstructed performance of randomly selected CelebA face images: (a) Original images; (b) images reconstructed by MMLE-convGMM. The sampling rates are 0.4.

## 7 结 论

本文提出了 convGMM 对整幅图像的概率分布进行建模, 该模型不仅利用了解卷积神经网络的思想, 对于整幅图像先验分布的建模具有非常好的效果, 而且兼具 GMM 的灵活性. 对于 convGMM 中未知参数的学习, 本文提出了通过 EM 算法求解 MMLE. 基于后验分布的期望从压缩测量结果中估计原信号, 且该估计是最小均方误差意义下的估计. 为解决对整幅图像直接进行计算时的维数较大、计算复杂度较高的问题, 在 convGMM 和压缩测量模型中都使用了循环卷积, 所有的训练和恢复过程都可以利用 2D-FFTs 实现快速运算. 仿真实验表明, 本文所提的 MMLE-convGMM 算法明显优于传统的压缩感知算法.

## 参考文献

- [1] Donoho D L 2006 *IEEE Trans. Inform. Theory* **52** 1289
- [2] Candès E J, Romberg J, Tao T 2006 *IEEE Trans. Inform. Theory* **52** 489
- [3] Ji S, Xue Y, Carin L 2008 *IEEE Trans. Sig. Process.* **56** 2346
- [4] Chen M, Silva J, Paisley J, Wang C, Dunson D, Carin L 2010 *IEEE Trans. Sig. Process.* **58** 6140
- [5] Yu G, Sapiro G 2011 *IEEE Trans. Sig. Process.* **59** 5842
- [6] Yang J, Liao X, Yuan X, Lhull P, Brady D J, Sapiro G, Carin L 2015 *IEEE Trans. Sig. Process.* **24** 106
- [7] Yu G, Sapiro G, Mallat S 2012 *IEEE Tran. Image Process.* **21** 2481
- [8] Wang C, Liu X F, Yu W K, Yao X R, Zheng F, Dong Q, Lan R M, Sun Z B, Zhai G J, Zhao Q 2017 *Chin. Phys. Lett.* **34** 104203
- [9] Wang Y Y, Ren Y C, Chen L Y, Song C, Li C Z, Zhang C, Xu D G, Yao J Q 2018 *Chin. Phys. B* **27** 114204
- [10] Xiao D, Cai H K, Zheng H Y 2015 *Chin. Phys. B* **24** 060505
- [11] Ning F L, He B J, Wei J 2013 *Acta Phys. Sin.* **62** 174212 (in Chinese) [宁方立, 何碧静, 韦娟 2013 物理学报 **62** 174212]
- [12] Li S D, Chen Y B, Liu R H, Ma X Y 2017 *Acta Phys. Sin.* **66** 038401 (in Chinese) [李少东, 陈永彬, 刘润华, 马晓岩 2017 物理学报 **66** 038401]
- [13] Duarte M F, Davenport M A, Takhar D, Laska J N, Sun T, Kelly K F, Baraniuk R G 2008 *IEEE Sig. Process. Mag.* **25** 83
- [14] Zeiler M D, Krishnan D, Taylor G W, Fergus R 2010 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* San Francisco, June 13–18, 2010 p2528
- [15] Grosse R, Raina R, Kwong H, Ng A Y 2007 *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence* Vancouver, July 19–22, 2007 p149
- [16] Yang J, Yu K, Huang T 2010 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* San Francisco, June 13–18, 2010 p3517
- [17] Wohlberg B 2016 *IEEE Trans. Image Process.* **25** 301
- [18] Davis P J 2012 *Circulant Matrices* (Providence, Rhode Island: American Mathematical Society)
- [19] Dempster A, Laird N, Rubin D 1977 *J. R. Stat. Soc.* **39** 1
- [20] Renna F, Calderbank R, Carin L, Rodrigues M R D 2014 *IEEE Trans. Sig. Process.* **62** 2265
- [21] Tropp J A, Gilbert A C 2007 *IEEE Trans. Inform. Theory* **53** 4655
- [22] Yang J, Zhang Y 2011 *SIAM J Sci. Comput.* **33** 250
- [23] Liao X, Li H, Carin L 2014 *SIAM Imaging Sci.* **7** 797
- [24] Xu Y, Yin W, Osher S 2014 *Inverse Probl. Imag.* **8** 901
- [25] Krizhevsky A, Hinton G 2009 *Learning Multiple Layers of Features from Tiny Images* (Toronto: University of Toronto) Technical Report Vol. 1, No. 4, p7
- [26] Li F F, Rob F, Pietro P 2007 *Comput. Vis. Image Und.* **106** 59
- [27] Liu Z W, Luo P, Wang X G, Tang X O 2015 *IEEE International Conference on Computer Vision* Santiago, December 7–13, 2015 p3730

# Statistical compressive sensing based on convolutional Gaussian mixture model\*

Wang Ren<sup>1)†</sup> Guo Jing-Bo<sup>2)</sup> Hui Jun-Peng<sup>1)</sup> Wang Ze<sup>1)</sup> Liu Hong-Jun<sup>1)</sup>  
Xu Yuan-Nan<sup>1)</sup> Liu Yun-Fo<sup>1)</sup>

1) (*China Academy of Launch Vehicle Technology R&D Center, Beijing 100076, China*)

2) (*Department of Electrical Engineering, Tsinghua University, Beijing 100084, China*)

( Received 24 March 2019; revised manuscript received 20 June 2019 )

## Abstract

Statistical compressive sensing needs to use the statistical description of source signal. By decomposing a whole image into a set of non-overlapping or overlapping patches, the Gaussian mixture model (GMM) has been used to statistically represent patches in an image. Compressive sensing, however, always imposes compression on the whole image. It is obvious that the entire image contains much richer information than the small patches. Extending from the small divided patches to an entire image, we propose a convolutional Gaussian mixture model (convGMM) to depict the statistics of an entire image and apply it to compressive sensing. We present the algorithm details by learning a convGMM from training images based on maximizing the marginal log-likelihood estimation. The learned convGMM is used to perform the model-based compressive sensing by using the convGMM as a model of the underlying image. In addition, aiming at the problem of high-dimensional image that makes learning, estimation and optimization suffer high computational complexity, all of the training and reconstruction process in our method can be fast and efficiently calculated in the frequency-domain by two-dimensional fast Fourier transforms. The performance of the convGMM on compressive sensing is demonstrated on several image sets.

**Keywords:** convolutional Gaussian mixture models, statistical compressive sensing, expectation maximization, convolutional sparse coding

**PACS:** 07.50.Qx, 05.45.-a, 05.45.Vx

**DOI:** 10.7498/aps.68.20190414

\* Project supported by the National Natural Science Foundation of China (Grant No. 51677094).

† Corresponding author. E-mail: wangren94@126.com