Machine Learning First Assignment

唐永承
4105053128
國立中興大學應用數學系

I. Introduction

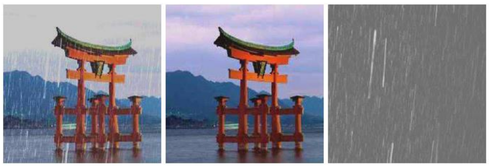
訊號分離的目標是要讓一個混合的訊號分離成原本所組成的成分。假設 y 是一個由兩個不同圖片所混合的訊號

$$y = \sum_{i \in C} x_i + \epsilon \tag{1}$$

$\{x_i\}_{i \in C}$ 是 y 所組成的成分。使用特定演算法讓 y 分離為 $\{x_i\}_{i \in C}$。以圖 1. 為例，將 y 拆分為 x_1 與 x_2。圖 2. 則是訊號分離的相關應用，可將下雨得痕跡移除。

y = x_1 + x_2

圖 1. 訊號分離示意圖



(a) Input (b) Ground Truth (c) Rain streak

圖 2. 訊號分離相關應用

II. Background

A. Sparse Approximation

假設有 L 個訊號，可表示為$\mathrm{Y} = [\mathrm{y}_{1,\dots,}\mathrm{y}_\mathrm{L}] \in \mathbb{R}^{\mathrm{N}\times\mathrm{L}}$，且有個字典 $\mathrm{D}= [\mathrm{d}_{1,\dots,}\mathrm{d}_\mathrm{M}] \in \mathbb{R}^{\mathrm{N}\times\mathrm{M}}$，然後稀疏近似問題可寫為

$$\arg\min_{A \in \mathbb{R}^{M\times L}} \frac{1}{2}|Y - DA|_F^2 + \lambda\Omega(A) \tag{2}$$

這裡$\Omega(A)$表示一個函數計算稀疏程度，也就是計算矩陣中每行L0 範數的合 $\Omega(A) = \sum_i |A_{\cdot,i}|_0$，$\lambda$則是用來平衡的正規化參數，通常會用貪婪演算法來解，例如匹配追蹤(Matching Pursuit, MP)或正交匹配追蹤(Orthogonal Matching Pursuit, OMP)。若用 L1 範數來替代 L0 範數，則可使用這些凸化演算法：ISTA、FISTA 或 ADMM 來解。

然而當面對較大的訊號時，常常會切成數個小塊，不過分割成小塊可能會破壞影像的結構，因此使用卷積稀疏編碼(Convolutional Sparse Coding, CSC)會是一個更好的方式，利用卷積來替代矩陣相乘，且字典元素也不需要和訊號有著相同形狀。讓$\mathrm{D} = \{\mathrm{d}_{1,\dots,}\mathrm{d}_\mathrm{M}\}$ 為一個有 M 個元素的字典，且使用*表示卷積，因此 CSC 問題可表示為

$$\arg\min_{\{\alpha_{l,m}\}} \frac{1}{2}\sum_l \left|\sum_m d_m * \alpha_{l,m} - y_l\right|_F^2 + \lambda \sum_l \sum_m \Omega(\alpha_{l,m}) \tag{3}$$

l 表示訊號的編號，$\alpha_{l,m}$ 為第 m 個字典對第 l 個訊號的係數圖(coefficient map)，且 d_m 可以與 y_l 有著不同的形狀。當 $\Omega(\alpha_{l,m})$ 是一個 L0 的稀疏程度計算的函數，就可以使用匹配追蹤(MP)或正交匹配追蹤(OMP)等貪婪演算法解此問題，此外若使用 L1 來近似 L0，也可以使用凸優化演算法來解此問題，例如：FISTA、ADMM。

B. Dictionary Leanrning

通常字典學習是試圖解以下問題

$$\arg\min_{D,A} \frac{1}{2}|Y - DA|_F^2 + \lambda_1\Omega(A) + \lambda_2 \sum_m \Gamma_C(d_m) \tag{4}$$

在此字典D是藉由稀疏矩陣A來近似大量訊號Y，然後定義指示函數為

$$\Gamma_S(x) = 0, \text{if } \mathrm{x} \in \mathrm{S}$$
$$\Gamma_S(x) = \infty, \text{otherwise} \tag{5}$$

這裡 S 為一些集合，指示函數會把 S 內的元素映射到 0，其他則映射到無限大。在式(4)中，字典長度被指示函數Γ_C所限制，C是一個非凸單位曲面($|\mathrm{d}_\mathrm{m}|_2^2 = 1$)或一個非凸單位球($|\mathrm{d}_\mathrm{m}|_2^2 \leq 1$)。如同稀疏編碼可延伸至卷積稀疏編碼，字典學習也可進一步延伸至卷積字典學習，問題可寫為以下式子

$$\arg\min_{\{\alpha_{l,m}\},\{d_m\}} \frac{1}{2}\sum_l \left|\sum_m d_m * \alpha_{l,m} - y_l\right|_F^2 + \lambda_1 \sum_l \sum_m \Omega(\alpha_{l,m}) + \lambda_2 \sum_m \Gamma_C(d_m) \tag{6}$$

在此式子中，$\{d_m\}$是由數個字典元素所組成，同時為了近似$\{y_l\}$，卷積稀疏近似可藉由式(3)來根據已學習到的字典來解。

通常會使用交替更新的迭代演算法，每次迭代包含兩步驟，分別為稀疏編碼和字典更新。在第一步時，可固定字典並使用式子(2)或(3)來解並獲得稀疏解，在第二步時，則更新字典。第一步在上面已提過，第二步字典學習可使用 Method of Optimal Directions (MOD)演算法或 K-SVD 演算法。概念上來說，MOD 是基於最小平方法，KSVD 則是基於 K-means 演算法。近年來有許多其他的方法，都是由這兩個方法所延伸的。

C. Singal Separation

在參考論文[1]，作者使用稀疏字典學習來解訊號分離問題，並將問題寫為

$$y = y_c + y_t \tag{7}$$

讓 $y \in \mathbb{R}^{N\times N}, y_c \in \mathbb{R}^{N\times N}, y_t \in \mathbb{R}^{N\times N}$，假設 y 是由兩個不同影像相加，y_c為卡通或較平滑的圖像，y_t為紋理圖案。作者將會對兩種類型的圖片使用卷積稀疏編碼(CSC)分別先訓練各自的濾波器。在 CSC 中，會先使用 M 張

訓練圖片 $\{y_m\}_{l=1}^{M}$.，目標是學習出一組卷積式的濾波器 (Convolutional Filters) $\{d_k\}_{l=1}^{K}$，藉由解出以下最佳化問題

$$\arg\min_{d,x} \frac{1}{2}\sum_{m=1}^{M}\left\|y_m - \sum_{k=1}^{K} d_x * X_{m,k}\right\|_2^2 + \lambda \sum_{m=1}^{M}\sum_{k=1}^{K}\left\|x_{m,k}\right\|_1 \quad (8)$$

$x_{m,k} \in \mathbb{R}^{N\times N}$ 是對於訓練資料 y_m的稀疏表示，λ為正規化參數。分別學習好卡通和紋理圖案的字典$\{d_{c,k}\}_{k=1}^{K_c}$和$\{d_{t,k}\}_{k=1}^{K_t}$，然後卡通圖片和紋理圖片可表示成$y_c = \sum_{k=1}^{K_c} d_{c,k} * x_{c,k}$ 和 $y_t = \sum_{k=1}^{K_c} d_{t,k} * x_{t,k}$，最後再解以下最佳化問題

$$\hat{x}_{c,k}\text{，}\hat{x}_{t,k} = \arg\min_{x_{c,k},\ x_{t,k}} \frac{1}{2}\left\|y - \sum_{k=1}^{K_c} d_{c,k} * x_{c,k} - \sum_{k=1}^{K_t} d_{t,k} * x_{t,k}\right\|_2^2 + \lambda_c \sum_{k=1}^{K_c}\left\|x_{c,k}\right\|_1 + \lambda_t \sum_{k=1}^{K_t}\left\|x_{t,k}\right\|_1 + \beta TV\left(\sum_{k=1}^{K_c} d_{c,k} * x_{c,k}\right) \quad (9)$$

一旦得到 $\hat{x}_{c,k}$ 和 $\hat{x}_{t,k}$，即可分離為卡通圖片 $\hat{y}_c = \sum_{k=1}^{K_c} d_{c,k} * \hat{x}_{c,k}$和紋理圖片$\hat{y}_t = \sum_{k=1}^{K_c} d_{t,k} * \hat{x}_{t,k}$。

III. Proposed Model

A. Signal separation with two dictionaries

如以下式子，預計使用兩個字典，分別學習兩類圖片，並加上一些限制來使兩個字典正交

$$\underset{D1,D2,\ x1,\ x2}{argmin} \|y_1 - D_1x_1\|_F^2 + \|y_2 - D_2x_2\|_F^2 + \lambda\left\|{D_1}^T D_2\right\|_F^2 + \Gamma_x(x_1) + \Gamma_x(x_2) + \Gamma_D(D_1) + \Gamma_D(D_2) \quad (10)$$

這裡y_1和y_2分別是兩類圖片，D_1和D_2分別為對應兩類圖片的字典，x_1和x_2是分別對應的稀疏表示，λ為政規化參數，Γ_x 和Γ_D 分別為稀疏限制式以及字典標準化。

利用學習完畢的兩個字典來分類混合的訊號，如以下式子

$$\underset{\hat{x}1,\ \hat{x}2}{argmin}\|y - D_1\hat{x}_1 - D_2\hat{x}_2\|_F^2 + \Gamma_x(\hat{x}_1) + \Gamma_x(\hat{x}_2) \quad (11)$$

最後可獲得分離後得到$D_1\hat{x}_1$和$D_2\hat{x}_2$。

B. Signal separation with three dictionaries

如以下式子，預計使用三個字典，分別學習兩類圖片，並加上一些限制來使D_1和D_2兩個字典與第三個共同字典D 正交，第三個字典目標要吸收掉無法分類的特徵

$$\underset{D1,D2,\ x1,\ x2}{argmin} \|y_1 - D_1x_1 - D\overline{x_1}\|_F^2 + \|y_2 - D_2x_2 - D\overline{x_2}\|_F^2 + \lambda\left\|{D_1}^T D_2\right\|_F^2 + \lambda_1\|D^T D_1\|_F^2 + \lambda_2\|D^T D_2\|_F^2 + \Gamma_x(x_1) + \Gamma_x(x_2) + \Gamma_D(D_1) + \Gamma_D(D_2) + \Gamma_D(D) \quad (12)$$

這裡y_1和y_2分別是兩類圖片，D_1和D_2分別為對應兩類圖片的字典，x_1和x_2是分別對應的稀疏表示，$\overline{x_1}$和$\overline{x_2}$是分別對應 D 的稀疏表示，λ為正規化參數，Γ_x 和Γ_D 分別為稀疏限制式以及字典標準化。

利用學習完畢的兩個字典來分類混合的訊號，如以下式子

$$\underset{\hat{x}1,\ \hat{x}2,x}{argmin}\|y - D_1\hat{x}_1 - D_2\hat{x}_2 - Dx\|_F^2 + \Gamma_x(\hat{x}_1) + \Gamma_x(\hat{x}_2) \quad (13)$$

最後可獲得分離後得到$D_1\hat{x}_1$和$D_2\hat{x}_2$。

IV. Experiment Result

A. Experiment result of two dicitonaries

先前訓練出一組字典，分別都有 64 個 atoms，x 則有 6 個非零數值，且λ = 100、$||{D_1}^T * D_2||_F$= 0.3490、$||{D_1}^T * D_1||_F$=36.9216、$||{D_2}^T * D_2||_F$= 41.0088。

做訊號分離的稀疏編碼時，丟入(11)式的 y 為當初訓練獲得的$y = D_1x_1 + D_2x_2$，結果分離出的兩張圖與原始的D_1x_1 和 D_2x_2 的 psnr 分別為 34.4100 和 30.5587。

B. Experiment result of three dictionaries

先前訓練出一組字典，D_1 和 D_2 分別有 64 個 atoms，D 則有 64 個 atoms，x 有 6 個非零數值，$\bar{x}$ 有 3 個非零數值，且λ = 100、λ_1=10、λ_2=10、$||{D_1}^T * D_2||_F$= 0.5138、$||D^T * D_1||_F$= 0.2322、$||D^T * D_2||_F$= 0.2190。

做訊號分離的稀疏編碼時，丟入(13)式的 y 為當初訓練獲得的$y = D_1x_1 + D_2x_2 + D\overline{x_1} + D\overline{x_2}$，結果分離出的兩張圖與原始的$D_1x_1$ 和 D_2x_2 的 psnr 分別為 34.0296 和 30.6101。

V. Conclution

從實驗的結果可以看出經由稀疏編碼得到的 x 與訓練時所得到的 x 的非零數值是在不同的位置上，稀疏編碼得到的 x 並沒有收斂到與訓練時一樣的解，故猜測 x 的初始值有重要的影響，因此下次實驗預計先加入 L1 proximal mapping。先使用 L1 讓這問題變成凸優化問題，進而得到較固定的起始值，接著再使用 L0 proximal mapping 來求得最後解。