Dictionary Learning

A representation learning method

Yihan Zhou(Joey)

April 15th, 2019

The University of British Columbia

Table of contents

1. Background
2. Introduction
3. Supervised Dictionary Learning
4. Online Dictionary Learning
5. Summary

Background

Sparse and overcomplete image models

”The **mammalian visual cortex** has evolved over millions of years to effectively cope with images of the natural environment. Given the importance of using resources efficiently in the competition for survival, it is reasonable to think that the cortex has **discovered efficient coding strategies for representing natural images**.”
— Olshausen & Field, 1996, *Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1 ?*

Sparse and overcomplete image models

- Sparse
  - Natural images may generally be described in terms of a small number of structural primitives
  - Model appropriate distribution
  - Capture higher order correlation
- Overcomplete
  - Robust, less sensitive to noise and other form of degradation
  - Flexible in matching the generative model to the input structure

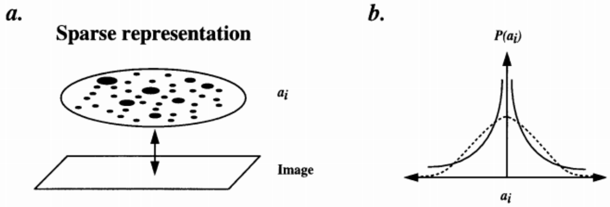


FIGURE 1. Sparse coding. (a) An image is represented by a small number of "active" code elements, a_i, out of a large set. Which elements are active varies from one image to the next. (b) Since a given element in a sparse code will most of the time be inactive, the probability distribution of its activity will be highly peaked around zero with heavy tails. This is in contrast to a code where the probability distribution of activity is spread more evenly among a range of values (such as a Gaussian).

Introduction

Definitions and goal

- We want to represent **signal** $x \in \mathbb{R}^m$
- A basis called **dictionary** $D = [d_1 \cdots d_k] \in \mathbb{R}^{m \times k}$
  Note that $k > m$ so D is overcomplete
- Each column d_i in the dictionary is called an **atom**

Goal: We want to find a linear combination of $\underbrace{\text{a ”few” atoms}}_{\text{sparsity}}$ from D that is $\underbrace{\text{”close” to the original signal } x}_{\text{low reconstruction error}}$.

Definitions and goal

What is the dictionary D?

Definitions and goal

What is the dictionary D?

- We can use predefined dictionary D, e.g. wavelet transform.
- However, **learned** dictionary has led to state-of-the-art performance for numerous tasks

Definitions and goal

What is the dictionary D?

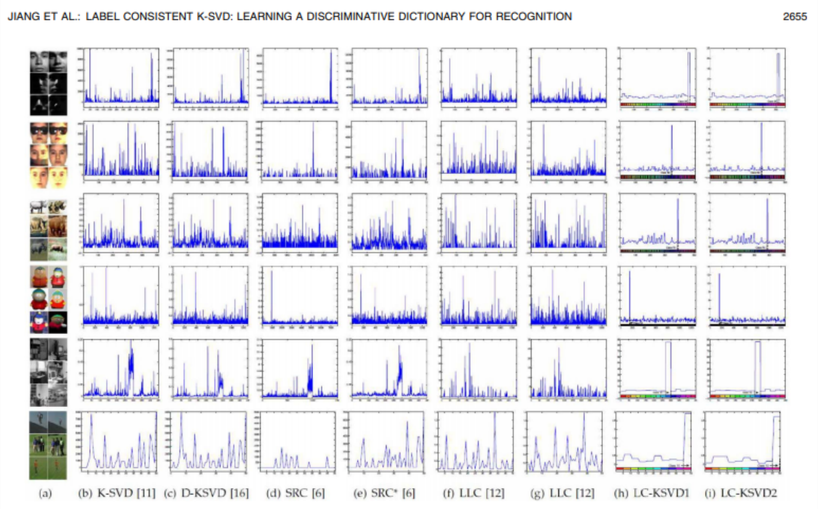
- We can use predefined dictionary D, e.g. wavelet transform.
- However, **learned** dictionary has led to state-of-the-art performance for numerous tasks

This is when the problem gets really interesting!
Real goal: Learn the dictionary D and a sparse representation with low reconstruction error.

Applications

Dictionary learning for classification:

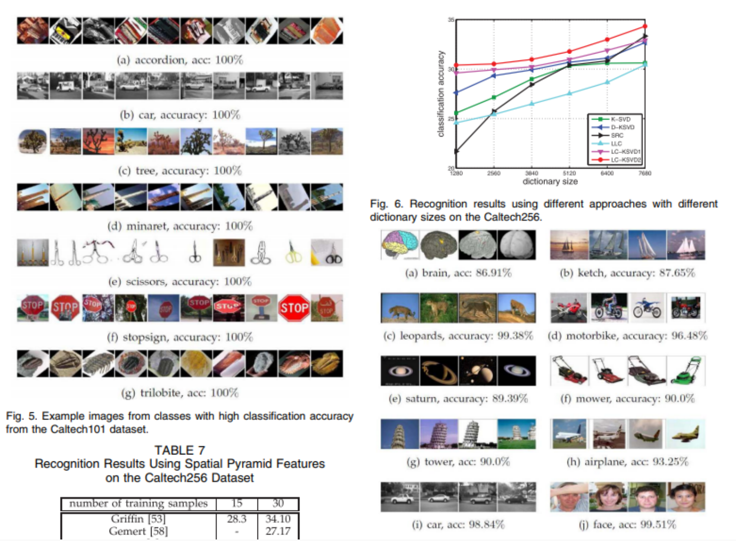
- Associate label information with dictionary learning



JIANG ET AL.: LABEL CONSISTENT K-SVD: LEARNING A DISCRIMINATIVE DICTIONARY FOR RECOGNITION 2655

(a) (b) K-SVD [11] (c) D-KSVD [16] (d) SRC [6] (e) SRC* [6] (f) LLC [12] (g) LLC [12] (h) LC-KSVD1 (i) LC-KSVD2

Applications



(a) accordion, acc: 100%

(b) car, accuracy: 100%

(c) tree, accuracy: 100%

(d) minaret, accuracy: 100%

(e) scissors, accuracy: 100%

(f) stopsign, accuracy: 100%

(g) trilobite, acc: 100%

Fig. 5. Example images from classes with high classification accuracy from the Caltech101 dataset.

TABLE 7
Recognition Results Using Spatial Pyramid Features
on the Caltech256 Dataset

number of training samples	15	30
Griffin [53]	28.3	34.10
Gemert [58]	-	27.17

Fig. 6. Recognition results using different approaches with different dictionary sizes on the Caltech256.

(a) brain, acc: 86.91%

(b) ketch, accuracy: 87.65%

(c) leopards, accuracy: 99.38%

(d) motorbike, accuracy: 96.48%

(e) saturn, accuracy: 89.39%

(f) mower, accuracy: 90.0%

(g) tower, acc: 90.0%

(h) airplane, acc: 93.25%

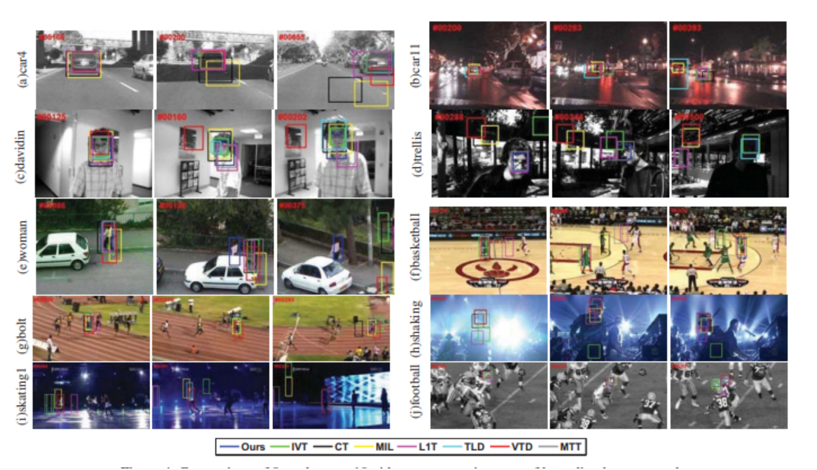
(i) car, acc: 98.84%

(j) face, acc: 99.51%

Jiang, Lin and Davis, *Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition*, 2013

Applications

Online dictionary learning for visual tracking:



Naiyan Wang, Jingdong Wang and Dit-Yan Yeung, *Online Robust Non-negative Dictionary Learning for Visual Tracking*, 2013

Supervised Dictionary Learning

Problem setting

Remember that for a **fixed** dictionary $D = [d_1 \cdots d_k] \in \mathbb{R}^{n \times k}$ and a signal $x \in \mathbb{R}^n$, our goal is to learn a sparse coding with low reconstruction error, so the objective is

$$R^*(x, D) = \min_{\alpha \in \mathbb{R}^k} \|x - D\alpha\|_2^2 + \lambda_1 \|\alpha\|_1$$

- l_1 norm leads to sparsity, but no analytic link between value of λ_1 and sparsity
- We can use l_0 norm instead, but then the objective function will not be convex
- In practice, l_1 norm is more stable

Problem setting

For a learned dictionary D, the objective function becomes

$$\min_{\alpha,D} \|x - D\alpha\|_2^2 + \lambda_1 \|\alpha\|_1$$

Note that D and α can be scaled at the same time, so we need to add constraint $\|d_i\|_2 \leq 1$ for every i.

Problem setting

For a learned dictionary D, the objective function becomes

$$\min_{\alpha,D} \|x - D\alpha\|_2^2 + \lambda_1 \|\alpha\|_1$$

Note that D and α can be scaled at the same time, so we need to add constraint $\|d_i\|_2 \leq 1$ for every i.
Now consider the classic classification setting, i.e., each signal belongs to one of p different classes. For simplicity, we assume that $p = 2$ for now and the label value $y \in \{-1, +1\}$.

Problem setting

For a learned dictionary D, the objective function becomes

$$\min_{\alpha,D} \|x - D\alpha\|_2^2 + \lambda_1 \|\alpha\|_1$$

Note that D and α can be scaled at the same time, so we need to add constraint $\|d_i\|_2 \leq 1$ for every i.
Now consider the classic classification setting, i.e., each signal belongs to one of p different classes. For simplicity, we assume that $p = 2$ for now and the label value $y \in \{-1, +1\}$.
Aside from learning dictionary D and sparse coding α, we also want to learn the **classification model**.

Classification model

Two simple classification models are used here:

- Linear in α: $f(x, \alpha, \theta) = w^T \alpha + b$, where $\theta = \{w \in \mathbb{R}^k, b \in \mathbb{R}\}$ parametrizes the model
    - Just a hyperplane, simplest model
- Bilinear in x and α: $f(x, \alpha, \theta) = x^T W \alpha + b$, where $\theta = \{W \in \mathbb{R}^{n \times k}, b \in \mathbb{R}\}$
    - W has more parameters than w so this model can be more complex
    - W can be viewed as a linear filter encoding x into a model for the coefficients α

Classification model

The objective for logistic regression is:

$$\min_{\theta} \sum_{i=1}^{m} C(y_i f(x_i, \alpha_i, \theta)) + \lambda_2 \|\theta\|_2^2$$

where $C = \log(1 + e^{-x})$ is the logistic loss.
Since we want to learn jointly dictionary D, coefficients α and model parameter θ, we put the two objectives together and get

$$\min_{D,\theta,\alpha} \left(\sum_{i=1}^{m} C(y_i f(x_i, \alpha_i, \theta)) + \lambda_0 \|x_i - D\alpha_i\|_2^2 + \lambda_1 \|\alpha_i\|_1 \right) + \lambda_2 \|\theta\|_2^2$$

We will refer to this model as SDL-G(supervised dictionary learning, generative)

Notation simplification

Now we simplify the notation, let
$S(\alpha_i, x_i, D, \theta, y_i) = C(y_i f(x_i, \alpha_i, \theta)) + \lambda_0 \|x_i - D\alpha_i\|_2^2 + \lambda_1 \|\alpha_i\|_1$ and
$S^*(x_i, D, \theta, y_i) = \min_\alpha S(\alpha, x_i, D, \theta, y_i)$.
So we can write the objective as

$$\min_{D,\theta,\alpha} \sum_{i=1}^{m} S(\alpha_i, x_i, D, \theta, y_i) + \lambda_2 \|\theta\|_2^2 = \min_{D,\theta} \sum_{i=1}^{m} S^*(x_i, D, \theta, y_i) + \lambda_2 \|\theta\|_2^2$$

And for a new signal $\hat{x}$, the prediction would be

$$\underset{y}{\operatorname{argmin}} S^*(\hat{x}, D, \theta, y)$$

A more discriminative version

We can make this model more discriminative because we want to have $S^*(x_i, D, \theta, y_i)$ less than $S^*(x_i, D, \theta, -y_i)$. Thus, we can put $S^*(x_i, D, \theta, -y_i) - S^*(x_i, D, \theta, y_i)$ into the logistic loss $C(\cdot)$ and get:

$$\min_{D,\theta} \left(\sum_{i=1}^{m} C(S^*(x_i, D, \theta, -y_i) - S^*(x_i, D, \theta, y_i)) \right) + \lambda_2 \|\theta\|_2^2$$

However, this problem is more difficult to solve this than SDL-G.(Not convex)

SDL-D

To make the discriminative objective easier to solve, we mix it with SDL-G and get

$$\left(\sum_{i=1}^{m} \mu C(S^*(x_i, D, \theta, -y_i) - S^*(x_i, D, \theta, y_i)) + (1-\mu)S^*(x_i, D, \theta, y_i)\right) + \lambda_2 \|\theta\|_2^2$$

Note that we have constraint $\|d_i\| \leq 1$ for each i.

We refer to this model as SDL-D(supervised dictionary learning, discriminative)

Extension to multiclass

- All of the above models admits a straightforward multiclass extension, by using **softmax** function $C_i(x_1, \cdots, x_p) = \log(\sum_{j=1}^{p} e^{x_j - x_i})$ and learn one model θ_i per class
- Other possible approaches such as one-vs-all or one-vs-one are also possible, and which one is the best is still remain open

Probabilistic interpretation of the linear model

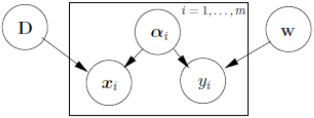


Figure 1: Graphical model for the proposed generative/discriminative learning framework.

Mairal, Bach, Ponce, Sapiro and Zisserman, *Supervised Dictionary Learning*, 2009

- w has a Gaussian prior, $p(w) \propto e^{-\lambda_2 \|w\|_2^2}$
- Each atom $d_i \sim S^{n-1}$ and are independent
- α_i are latent variables with a Laplace prior, $p(\alpha_i) \propto e^{-\lambda_1 \|\alpha_i\|_1}$
- Conditional probability of x_i is Gaussian, i.e., $p(x_i | \alpha_i, D) \propto e^{-\lambda_0 \|x_i - D\alpha_i\|_2^2}$. All x_i's are independent
- Conditional probability of y_i is given by $p(y_i = \epsilon | \alpha_i, w) = e^{-\epsilon w^T \alpha_i} / (e^{-w^T \alpha_i} + e^{w^T \alpha_i})$

Probabilistic interpretation of the linear model

- Under this graphical model, MAP estimation of the joint distribution $p(\{x_i, y_i\}_{i=1}^m, D, w)$ gives SDL-G.
- Similarly, MAP estimation of the conditional distribution $p(\{y_i\}_{i=1}^m, D, w | \{x_i\}_{i=1}^m)$.
- SDL-D is a classical trade-off between generative and discriminative, where generative components are added to discriminative frameworks to add robustness, e.g., to noise and occlusions.

Kernel interpretation of the bilinear model

- The bilinear model does not admit a straightforward probabilistic interpretation with kernel $K(x_1, x_2) = \alpha_1^T \alpha_2 x_1^T x_2$.
- However, it admits a kernel interpretation. For simplicity, we ignore the constant b here, so logistic regression objective can be written as

$$g = \sum_{i=1}^{m} \log(1 + e^{-y_i f(x_i)}) + \frac{\lambda_2}{2} \|\theta\|_2^2$$

  where $f(x_i) = x_i^T W \alpha_i$

- By Representer Theorem(Schlkopf, Herbrich, and Smola), the optimal f can be written as

$$f(x) = \sum_{i=1}^{m} \beta_i K(x, x_i)$$

Kernel interpretation of the bilinear model

If you are not convinced, here is my derivation:

$$\frac{\partial g}{\partial W_{pq}} = -\sum_{i=1}^{m} \frac{1}{e^{-y_i f(x_i)}+1} x_i^p \alpha_i^q + W_{pq}$$

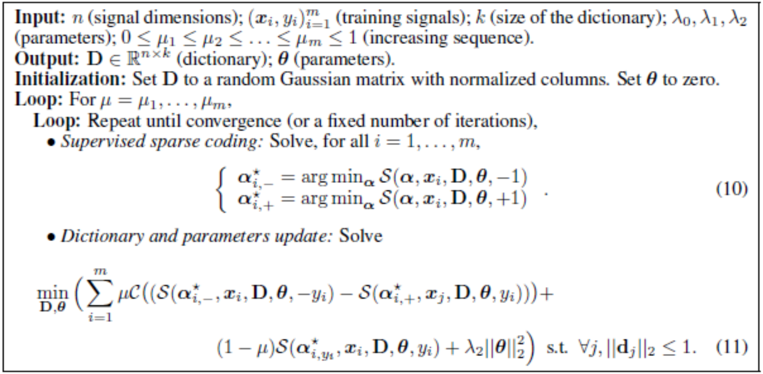
$$W_{pq}^* = \sum_{i=1}^{m} \frac{1}{e^{-y_i f(x_i)}+1} x_i^p \alpha_i^q$$

$$W^* = \sum_{i=1}^{m} \frac{1}{e^{-y_i f(x_i)}+1} x_i \cdot \alpha_i^T = \sum_{i=1}^{m} \beta_i x_i \cdot \alpha_i^T$$

Then the prediction is

$$f(x) = x^T W^* \alpha = x^T \left(\sum_{i=1}^{m} \beta_i x_i \cdot \alpha_i^T \right) \alpha = \sum_{i=1}^{m} \beta_i \alpha^T \alpha_i x_i^T x = \sum_{i=1}^{m} \beta_i K(x, x_i)$$

Training procedure



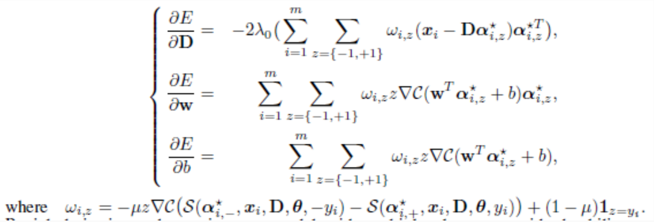
```
Input: $n$ (signal dimensions); $(\boldsymbol{x}_i, y_i)_{i=1}^m$ (training signals); $k$ (size of the dictionary); $\lambda_0, \lambda_1, \lambda_2$ (parameters); $0 \leq \mu_1 \leq \mu_2 \leq \ldots \leq \mu_m \leq 1$ (increasing sequence).
Output: $\mathbf{D} \in \mathbb{R}^{n \times k}$ (dictionary); $\boldsymbol{\theta}$ (parameters).
Initialization: Set $\mathbf{D}$ to a random Gaussian matrix with normalized columns. Set $\boldsymbol{\theta}$ to zero.
Loop: For $\mu = \mu_1, \ldots, \mu_m$,
  Loop: Repeat until convergence (or a fixed number of iterations),
    • Supervised sparse coding: Solve, for all $i = 1, \ldots, m$,
        $\begin{cases} \boldsymbol{\alpha}_{i,-}^\star = \arg\min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -1) \\ \boldsymbol{\alpha}_{i,+}^\star = \arg\min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, +1) \end{cases} .$   (10)
    • Dictionary and parameters update: Solve
        $\min_{\mathbf{D},\boldsymbol{\theta}} \Big( \sum_{i=1}^{m} \mu \mathcal{C}\big((\mathcal{S}(\boldsymbol{\alpha}_{i,-}^\star, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}(\boldsymbol{\alpha}_{i,+}^\star, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}, y_i))\big) +$
        $(1-\mu)\mathcal{S}(\boldsymbol{\alpha}_{i,y_i}^\star, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) + \lambda_2 ||\boldsymbol{\theta}||_2^2 \Big)$ s.t. $\forall j, ||\mathbf{d}_j||_2 \leq 1.$   (11)
```

Figure 2: *SDL:* Supervised dictionary learning algorithm.

- Supervised sparse coding
- Dictionary and parameters update

Training procedure

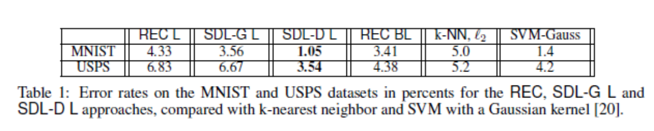
- Supervised sparse coding can be solved efficiently by **fixed-point continuation method(FPC)**.
- The dictionary updating objective is not convex in general, but a **local minimum** can be obtained using projected gradient descent. This local minimum is good enough for classification tasks in practice.



$$\begin{cases} \dfrac{\partial E}{\partial \mathbf{D}} = & -2\lambda_0 \Big(\sum_{i=1}^{m} \sum_{z=\{-1,+1\}} \omega_{i,z} (\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}^{\star}_{i,z}) \boldsymbol{\alpha}^{\star T}_{i,z} \Big), \\ \dfrac{\partial E}{\partial \mathbf{w}} = & \sum_{i=1}^{m} \sum_{z=\{-1,+1\}} \omega_{i,z} z \nabla \mathcal{C}(\mathbf{w}^T \boldsymbol{\alpha}^{\star}_{i,z} + b) \boldsymbol{\alpha}^{\star}_{i,z}, \\ \dfrac{\partial E}{\partial b} = & \sum_{i=1}^{m} \sum_{z=\{-1,+1\}} \omega_{i,z} z \nabla \mathcal{C}(\mathbf{w}^T \boldsymbol{\alpha}^{\star}_{i,z} + b), \end{cases}$$

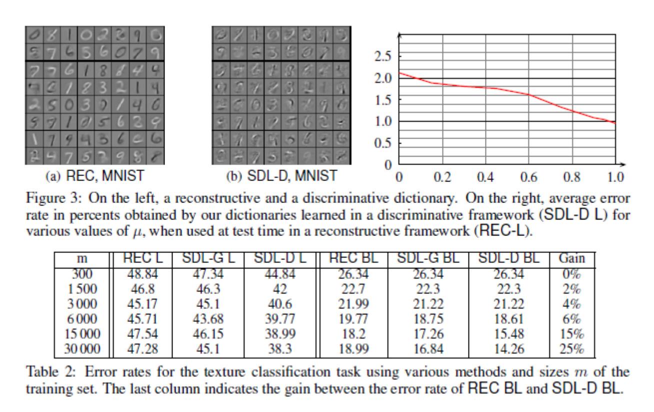
where $\omega_{i,z} = -\mu z \nabla \mathcal{C}\big(\mathcal{S}(\boldsymbol{\alpha}^{\star}_{i,-}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}(\boldsymbol{\alpha}^{\star}_{i,+}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)\big) + (1-\mu)\mathbf{1}_{z=y_i}$.

Experiments



	REC L	SDL-G L	SDL-D L	REC BL	k-NN, ℓ_2	SVM-Gauss
MNIST	4.33	3.56	**1.05**	3.41	5.0	1.4
USPS	6.83	6.67	**3.54**	4.38	5.2	4.2

Table 1: Error rates on the MNIST and USPS datasets in percents for the REC, SDL-G L and SDL-D L approaches, compared with k-nearest neighbor and SVM with a Gaussian kernel [20].



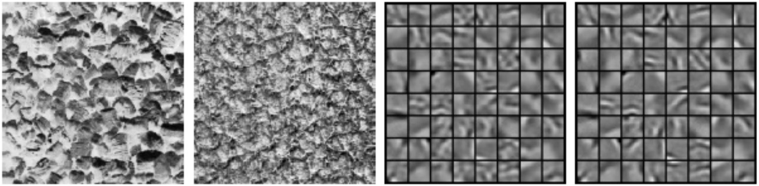
(a) REC, MNIST (b) SDL-D, MNIST

Figure 3: On the left, a reconstructive and a discriminative dictionary. On the right, average error rate in percents obtained by our dictionaries learned in a discriminative framework (SDL-D L) for various values of μ, when used at test time in a reconstructive framework (REC-L).

m	REC L	SDL-G L	SDL-D L	REC BL	SDL-G BL	SDL-D BL	Gain
300	48.84	47.34	44.84	26.34	26.34	26.34	0%
1 500	46.8	46.3	42	22.7	22.3	22.3	2%
3 000	45.17	45.1	40.6	21.99	21.22	21.22	4%
6 000	45.71	43.68	39.77	19.77	18.75	18.61	6%
15 000	47.54	46.15	38.99	18.2	17.26	15.48	15%
30 000	47.28	45.1	38.3	18.99	16.84	14.26	25%

Table 2: Error rates for the texture classification task using various methods and sizes m of the training set. The last column indicates the gain between the error rate of REC BL and SDL-D BL.

Experiments



(a) Texture 1 (b) Texture 2 (c) REC (d) SDL-D BL

Figure 4: Left: test textures. Right: reconstructive and discriminative dictionaries

Online Dictionary Learning

Why online learning?

The optimization problem is a significant computation challenge for dictionary learning.

- Very large training sets, particularly in the context of image processing tasks.
- Dynamic training data changing over time, such as video sequences.

Why online learning?

The optimization problem is a significant computation challenge for dictionary learning.

- Very large training sets, particularly in the context of image processing tasks.
- Dynamic training data changing over time, such as video sequences.

Online approach can address this issue.

- Process one or a small batch of the training set at one time.
- Techniques based on stochastic approximation like first-order stochastic gradient descent with projection works, and it is possible to go further.

Problem formulation

The **empirical cost** function is

$$f_n(D) = \frac{1}{n}\sum_{i=1}^{n} l(x_i, D)$$

where $l(x_i, D) = \min_{\alpha \in \mathbb{R}^k} \frac{1}{2}\|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1$
Recall that our dictionary learning problem formulation:

$$\min_{D \in \mathcal{C}} f_n(D)$$

where $\mathcal{C} = \{D \in \mathbb{R}^{m \times k} \text{ s.t. } \forall j = 1, \cdots, k,\ d_j^T d_j \leq 1\}$

Stochastic approximation

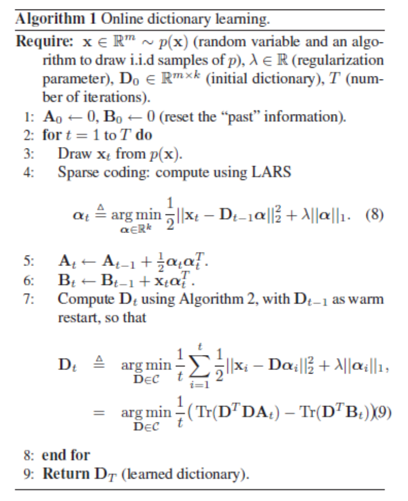
Actually, we are not interested in minimizing the **empirical cost** $f_n(D)$, but minimizing the **expected cost** $f(D)$ defined as the following:

$$f(D) = \mathbb{E}_x\left[l(x, D)\right] = \lim_{n \to \infty} f_n(D)$$

Thus, we can use stochastic approximation.
The classical projected first-order stochastic gradient descent consists of a sequence of updates of D:

$$D_t = \Pi_{\mathcal{C}}\left[D_{t-1} - \frac{\rho}{t}\nabla_D l(x_t, D_{t-1})\right]$$

Online dictionary learning algorithm



Algorithm 1 Online dictionary learning.

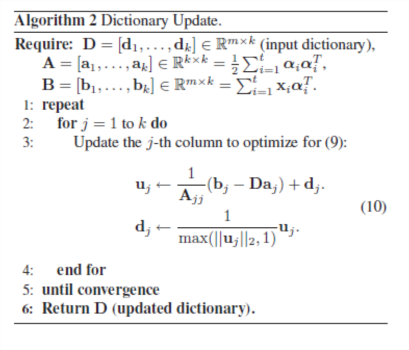
```
Require: $\mathbf{x} \in \mathbb{R}^m \sim p(\mathbf{x})$ (random variable and an algorithm to draw i.i.d samples of $p$), $\lambda \in \mathbb{R}$ (regularization parameter), $\mathbf{D}_0 \in \mathbb{R}^{m \times k}$ (initial dictionary), $T$ (number of iterations).
$\mathbf{A}_0 \leftarrow 0$, $\mathbf{B}_0 \leftarrow 0$ (reset the "past" information).
for $t = 1$ to $T$ do
    Draw $\mathbf{x}_t$ from $p(\mathbf{x})$.
    Sparse coding: compute using LARS
        $\boldsymbol{\alpha}_t \triangleq \underset{\boldsymbol{\alpha} \in \mathbb{R}^k}{\arg\min} \frac{1}{2}||\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1.$   (8)
    $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \frac{1}{2}\boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^T$.
    $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t\boldsymbol{\alpha}_t^T$.
    Compute $\mathbf{D}_t$ using Algorithm 2, with $\mathbf{D}_{t-1}$ as warm restart, so that
        $\mathbf{D}_t \triangleq \underset{\mathbf{D} \in \mathcal{C}}{\arg\min} \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1,$
        $= \underset{\mathbf{D} \in \mathcal{C}}{\arg\min} \frac{1}{t}\big(\text{Tr}(\mathbf{D}^T\mathbf{D}\mathbf{A}_t) - \text{Tr}(\mathbf{D}^T\mathbf{B}_t)\big)$(9)
end for
Return $\mathbf{D}_T$ (learned dictionary).
```

- In the dictionary update step, we compute D_t by minimizing over $\mathcal{C}$ the function

$$\hat{f}_t(D) = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\|x_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1$$

- Note that $\hat{f}_t$ and f_t are different! f_t minimize over D and all α while $\hat{f}_t$ minimize over D given α. $\hat{f}_t$ acts as a surrogate of f_t.
- $\hat{f}_t$ aggregates the past information computed during previous steps, namely α_i, it is an upper bound of f_t.
- $\hat{f}_t$ is close to $\hat{f}_{t-1}$, so D_t can be obtained efficiently using D_{t-1} as warm restart.

Dictionary update



```
Algorithm 2 Dictionary Update.
Require: $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ (input dictionary),
    $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_k] \in \mathbb{R}^{k \times k} = \frac{1}{2}\sum_{i=1}^{t} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T$,
    $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_k] \in \mathbb{R}^{m \times k} = \sum_{i=1}^{t} \mathbf{x}_i \boldsymbol{\alpha}_i^T$.
repeat
    for $j = 1$ to $k$ do
        Update the $j$-th column to optimize for (9):
            $\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}}(\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j.$
            $\mathbf{d}_j \leftarrow \frac{1}{\max(||\mathbf{u}_j||_2, 1)}\mathbf{u}_j.$        (10)
    end for
until convergence
Return D (updated dictionary).
```

- This algorithm is block-coordinate descent with warm restart.
- This convex optimization problem admits separable constraints in the updated blocks(columns), convergence to a global optimum is guaranteed(Bertsekas, 1999).
- In practice, α is sparse so A in general concentrated on the diagonal, making block-coordinate descent more efficient.(One iteration!)

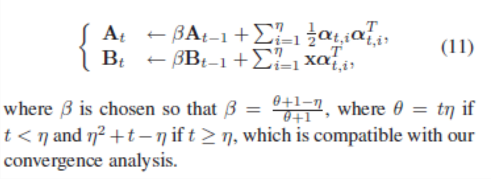
Optimization of the algorithm

The above algorithm can be further optimized.

- Handling fixed-size datasets
  - If the dataset is finite, same data point may be examined several times. Instead, we can cycle over a randomly permuted training set.
  - In our training, we can remove the "old" information concerning x from A_t and B_t by

$$A_t \leftarrow A_{t-1} + \alpha_t \alpha_t^T - \alpha_{t_0} \alpha_{t_0}^T$$

- Mini-batch extension: Use a batch of data points instead of one in each iteration



$$\begin{cases} \mathbf{A}_t & \leftarrow \beta \mathbf{A}_{t-1} + \sum_{i=1}^{\eta} \frac{1}{2} \boldsymbol{\alpha}_{t,i} \boldsymbol{\alpha}_{t,i}^T, \\ \mathbf{B}_t & \leftarrow \beta \mathbf{B}_{t-1} + \sum_{i=1}^{\eta} \mathbf{x} \boldsymbol{\alpha}_{t,i}^T, \end{cases} \tag{11}$$

where β is chosen so that $\beta = \frac{\theta+1-\eta}{\theta+1}$, where $\theta = t\eta$ if $t < \eta$ and $\eta^2 + t - \eta$ if $t \geq \eta$, which is compatible with our convergence analysis.

Convergence analysis

This algorithm is simple. What nice property does it have?

Convergence analysis

This algorithm is simple. What ”nice” property does it have?

- Stochasticity
- Non-convexity

Convergence analysis

This algorithm is simple. What ”nice” property does it have?

- Stochasticity
- Non-convexity

This means the convergence analysis is hard.

Sketch of proof sketch

First we need to have the following three assumptions:
(A) The data admits a bounded probability density p with compact support K.
(B) The quadratic surrogate function $\hat{f}(t)$ are strictly convex with lower-bounded Hessians.
(C) The sparse coding solution is unique under some sufficient conditions.

Proposition 1

Under assumptions (A)-(C):

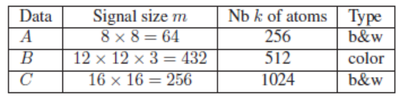
- $\hat{f}_t(D_t)$ *converges a.s.;*
- $f(D_t) - \hat{f}_t(D_t)$ *converges a.s. to* 0*; and*
- $f(D_t)$ *converges a.s.*

Proposition 2

Under assumptions (A) to (C), D_t *is asymptotically close to the set of stationary points of the dictionary learning prob- lem with probability one.*

Core theorem used is in Borwein, J., & Lewis, A. (2006). *Convex analysis and nonlinear optimization: theory and examples.* Springer.

Experiments



Data	Signal size m	Nb k of atoms	Type
A	$8 \times 8 = 64$	256	b&w
B	$12 \times 12 \times 3 = 432$	512	color
C	$16 \times 16 = 256$	1024	b&w

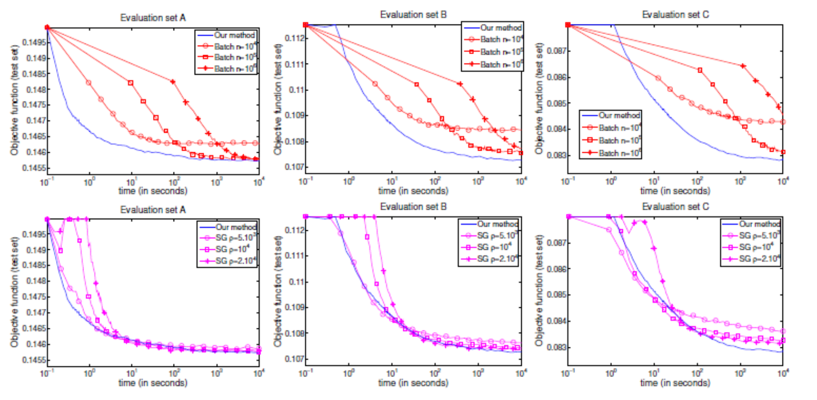


Figure 1. **Top:** Comparison between online and batch learning for various training set sizes. **Bottom:** Comparison between our method and stochastic gradient (SG) descent with different learning rates ρ. In both cases, the value of the objective function evaluated on the test set is reported as a function of computation time on a logarithmic scale. Values of the objective function greater than its initial value are truncated.

Experiments

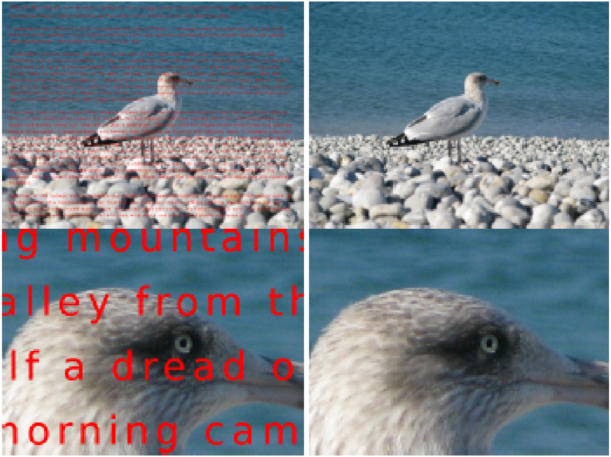


Figure 2. Inpainting example on a 12-Megapixel image. Top: Damaged and restored images. Bottom: Zooming on the damaged and restored images. (Best seen in color)

Summary

Summary and my opinion

- Dictionary learning learns an overcomplete basis called dictionary and a sparse representation of the training data.
- The sparse coding is efficient in representing certain signals, e.g. natural images and achieves state-of-art performance in some tasks.
- We can use supervised dictionary learning for classification tasks.
- Online learning techniques can be used for large-scaled or dynamic dataset.

References

This presentation uses content from [1, 2, 3, 4, 5, 6]

References i

Z. Jiang, Z. Lin, and L. S. Davis.
Label consistent k-svd: Learning a discriminative dictionary for recognition.
IEEE transactions on pattern analysis and machine intelligence, 35(11):2651–2664, 2013.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro.
Online dictionary learning for sparse coding.
In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach.
Supervised dictionary learning.
In *Advances in neural information processing systems*, pages 1033–1040, 2009.

References ii

B. A. Olshausen and D. J. Field.
Sparse coding with an overcomplete basis set: A strategy employed by v1?
Vision research, 37(23):3311–3325, 1997.

N. Wang, J. Wang, and D.-Y. Yeung.
Online robust non-negative dictionary learning for visual tracking.
In *Proceedings of the IEEE international conference on computer vision*, pages 657–664, 2013.

J. Zhu and T. Hastie.
Kernel logistic regression and the import vector machine.
In *Advances in neural information processing systems*, pages 1081–1088, 2002.

Questions?

Thank you and wish everyone a great summer!