# Dictionary Learning and Sparse Coding

Kuei Hsien Chu

May 13, 2016

## 1 Introduction

One of the prime objective for machine learning is to formulate a decision-making process by "learning" a set of data. The "learning" is emphasized here because instead of being a forecasting model, where the end goals may be the formalization of relationships between variables in the form of mathematical equations, machine learning is a method of data analysis that automates analytical model building [1]. In short, machine learning is more closely related to the goal of artificial intelligence.

To formulate this decision-making process that can be implemented in a computer, decisions are typically translated to optimization problems. In our paper, we introduces the concepts that are used in Dictionary Learning.

## 2 Convexity and norms

**Convex set** A set $S \subset \mathbb{R}^n$ is convex if and only if

$$\left.\begin{aligned} &\forall x \in S \\ &\forall y \in S \\ &\forall \lambda (0 \leq \lambda \leq 1) \end{aligned}\right\} \qquad \Rightarrow \lambda x + (1-\lambda) y \in S$$

**Norm** A norm is a way of measuring the length of a vector. Let $V$ be a vector space. A *norm* on $V$ is a function $||\cdot|| : V \to [0, \infty)$ satisfying

(a) $(\forall v \in V) \quad ||v|| \geq 0$ and $||v|| = 0$ iff $||v|| = 0$.

(b) $(\forall \alpha \in \mathbb{R})(\forall v \in V) \quad ||\alpha v|| = |\alpha| \cdot ||v||$

(c) $(\forall v, w \in V) \quad ||u + w|| \leq ||u|| + ||w||$ (triangular inequality)

**example** For a given vector $x \in \mathbb{R}^n$, the Euclidean norm is defined as $||x||_2 := \sqrt{x_1^2 + ... + x_n^2}$.

**$\ell_p$-norm** Besides Euclidean norm, there may be problems in different context that require us to use other ways to measure the distance. P-norm or $\ell_p$ norm is a more general case to define the distance. Let $x \in \mathbb{R}^n$, and $p \in \mathbb{R}$ such that $p \geq 1$, $\ell_p$-norm is defined as

$$||x||_p := (\sum_{i=1}^{n} |x_i|^p)^{1/p}$$

**$\ell_0$ norm** Following the definition of $\ell_p$-norm, for $p = 0$, we can see that the equation $(\sum_{i=1}^{n} |x_i|^0)^{1/0}$ does not make sense since the denominator of the exponent is zero. Mathematician just define $\ell_0$-norm as the number non-zero elements in a vector, that is

$$||x||_0 := \#\{x_i, \neq 0\}$$

**Convex function** A function $f : \mathbb{R}^n \to \mathbb{R}$ defined on a convex set $S$ is convex, if it satisfies

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

for all $x, y \in \mathbb{R}^n$ and $\lambda \in [0, 1]$.

## 3 Convex Optimization theory

**Mathematical Optimization** A *mathematical optimization problem*, or just *optimization problem*, has the form

$$\begin{aligned} &\text{minimize } f_O(x) \\ &\text{subject to } f_i(x) \leq b_i, i = 1, 2, ..., m \end{aligned}$$

where the vector $x$ is the *optimization variable* of the problem, the function $f_O : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, the functions $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, ..., m$, are the *constraint functions*, and the constraints $b_1, b_2, ..., b_m$ are the limits, or bounds, for the constraints. A vector $x^*$ is called *optimal*, or a solution of the problem, if it has the smallest value among all vectors that satisfy the constraints: for any $z$ with $f_1(z) \leq b_1, ..., f_m(z) \leq b_m$, we have $f_O(z) \geq f_O(x^*)$ [3].

**Convex Optimization** A convex optimization problem has the same form as in mathematical optimization, however, the objective function $f_O : \mathbb{R}^n \to \mathbb{R}$ and constraint functions $f_1, ..., f_m : \mathbb{R}^n \to \mathbb{R}$ are convex functions.

**Local and global optima** For convex optimization, every local minimum is also a global minimum.

## 4 Optimization problems

### 4.1 Least Squares minimization

Let $\vec{y} \in \mathbb{R}^k$ be a vector and $A \in \mathbb{R}^{k \times n}$ be a matrix. These are fixed objects, and in practice they are often some sort of observed data. The least squares minimization problem is

$$\min_{\beta \in \mathbb{R}^n} ||\vec{y} - A\vec{\beta}||_2^2.$$

**Least squares minimization is convex**

**Solution to least squares** Often times, the solution to $A\vec{\beta} = \vec{y}$ does not exist, that is, there is no set of linear combinations $\vec{a_1}\beta_1 + \vec{a_2}\beta_2 + ... + \vec{a_n}\beta_n$, where we can get $\vec{y}$.

**Geometric intuition of least squares - example** Let's say we want to estimate the relationship between the rooms of a apartment and the housing price of a particular area:

| Rooms (X) | Price (Y, \$10,000) |
|---|---|
| 3 | 5 |
| 4 | 7 |
| 6 | 9 |

The graph of the regression problem we have:

Now, let $\theta_0, \theta_1 \in \mathbb{R}$ be the best fit coefficients of our linear equation, then the errors, $\epsilon_1$, $\epsilon_2$, and $\epsilon_3$, according to out data entries are:

$$\begin{aligned}
\epsilon_1 &= y_1 - (\theta_0 + \theta_1 x_1) = 5 - (\theta_0 + 3\theta_1) \\
\epsilon_2 &= y_2 - (\theta_0 + \theta_1 x_2) = 7 - (\theta_0 + 4\theta_1) \\
\epsilon_3 &= y_3 - (\theta_0 + \theta_1 x_3) = 9 - (\theta_0 + 6\theta_1)
\end{aligned}$$

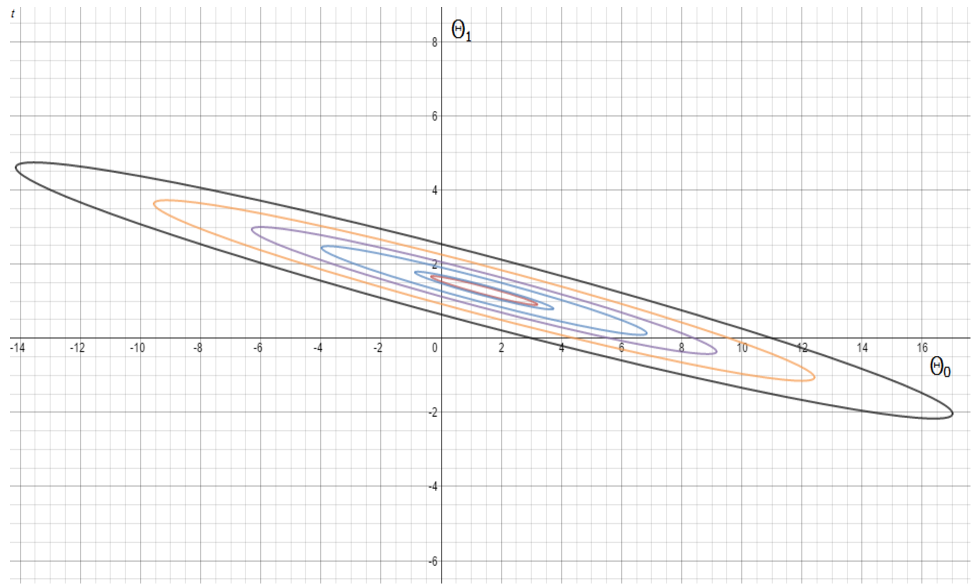
Recall the Euclidean norm, $||\cdot|| = \sqrt{z_1^2 + z_2^2 + ... + z_n^2}$ for some given vector $z$. The error following the example above thus becomes

$$\begin{aligned} ||\epsilon|| &= \sqrt{\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2} \\ &= \sqrt{\sum_1^3 (y_i - (\theta_0 + \theta_1 x_i))^2} \end{aligned}$$

Let X = $[x_0^{(i)} \quad x_1^{(i)}]$, where $x_0^{(i)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ and $x_1^{(i)} = \begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix}$ following the above examples. Note that the superscript $(i)$ means the $i$th training example, and $x_0$ by common practice is set to as vector of ones for computation purposes. By the definition of least squares, the problem becomes

$$\begin{aligned} \min_{\theta_0,\theta_1 \in \mathbb{R}} ||\epsilon||^2 &= \min_{\theta_0,\theta_1 \in \mathbb{R}} ||Y - X\theta||^2 \\ &= \min_{\theta_0,\theta_1 \in \mathbb{R}} \sum_{i=1}^3 (y_i - (\theta_0 + \theta_1 x_1^{(i)}))^2 \\ &= \min_{\theta_0,\theta_1 \in \mathbb{R}} [(5 - (\theta_0 + 3\theta_1))^2 + (7 - (\theta_0 + 4\theta_1)^2 + (9 - (\theta_0 + 6\theta_1))^2] \end{aligned}$$

Here, we can see the equation $[(5 - (\theta_0 + 3\theta_1))^2 + (7 - (\theta_0 + 4\theta_1)^2 + (9 - (\theta_0 + 6\theta_1))^2]$ is in the form of a tilted ellipse (Appendix B). Set $P = [(5 - (\theta_0 + 3\theta_1))^2 + (7 - (\theta_0 + 4\theta_1)^2 + (9 - (\theta_0 + 6\theta_1))^2]$, and let P = 1,7,14,28,56, respectively, we can then generate different level curves of $P$, where the typical x-axis is the values of $\theta_0$ and the typical y-axis is the values of $\theta_1$



This is related to the Least Squares optimization problem because the ellipses above are level curves for the objective function $f(\theta) = ||Y - X\theta||^2$, where these level curves are given by the set of $\theta$ values such that $f(\theta) = P$. Hence the Least Squares minimization problem is exactly finding an $\theta$ value that lies on the level curve $f(\theta) = P$ corresponding to the small $P$ possible. If there is a solution to the exertion $f(\theta) = 0$, then the minimization can be obtained with zero error and the optimal $\theta^*$ yields $X\theta^* = Y$.

## 4.2 Ridge Least Squares minimization

Let $\vec{y} \in \mathbb{R}^k$ be a vector, $t \in \mathbb{R}$ and $A \in \mathbb{R}^{k\times n}$ be a matrix. Ridge regression is defined as

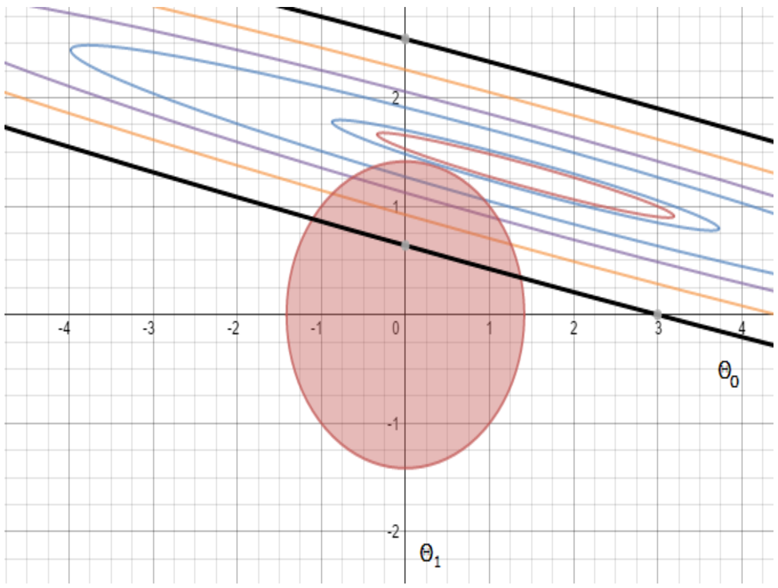
$$\min_{\beta\in\mathbb{R}^n} ||\vec{y} - A\vec{\beta}||_2^2.$$
$$\text{s.t.} \quad ||\vec{\beta}||_2 \leq t$$

This is a convex optimization problem because $B_2(0,t) = \{\vec{\beta} \in \mathbb{R}^n : ||\vec{\beta}||_2 \leq t\}$ is a convex set, and $f(\vec{\beta}) = ||\vec{y} - A\vec{\beta}||_2^2$ is a convex function on $B_2(0,t)$. Therefore the Ridge Least Square minimization problem is a convex minimization problem, and hence it has a solution.

To view the geometrically, let us re-use the housing price example again. Here, the problem now becomes

$$\min_{\theta_0,\theta_1\in\mathbb{R}} \sum_{1}^{3}(y_i-(\theta_0+\theta_1 x_i))^2 \quad \text{s.t.} \quad ||\theta||_2 \le t$$

where $\theta = \begin{bmatrix}\theta_0\\ \theta_1\end{bmatrix}$ and $||\theta||_2^2$ is the square of the euclidean norm. For $t=\sqrt{2}$, the picture is shown below:



and which we can see this minimization problem can have $\theta_0, \theta_1$ anywhere on the graph as long as $\sqrt{\theta_0^2+\theta_1^2} \le \sqrt{2}$. In this graph, the minimum point is the point that lies on the boundary (where $||\theta|| = \sqrt{||\theta_0^2+\theta_1^2} = \sqrt{2}$),and in particular it lies in intersection of the boundary of $B_2(0,\sqrt{2})$ and the smallest ellipse $f(\vec{\beta}) = P$ that intersects $B_2(0,\sqrt{2})$. Note that based on this description of the minimization problem, we can conclude that the minimizer $(\theta_0,\theta_1)$ will have no zero components; that is, the minimizer $(\theta_0,\theta_1)$ does not lie on the coordinate axes. So this solution is not "sparse." In fact, it is not hard to see that solutions to the Ridge Least Squares will rarely result in sparse optimizers, nor will the unrestricted Least Squares for that matter.

## 4.3 The Lasso optimization problem

The Lasso is a constrained version of least squares. Let $\vec{y} \in \mathbb{R}^n$ be a vector, $t \in \mathbb{R}$ and $A \in \mathbb{R}^{n \times k}$ be a matrix. Lasso regression is defined as

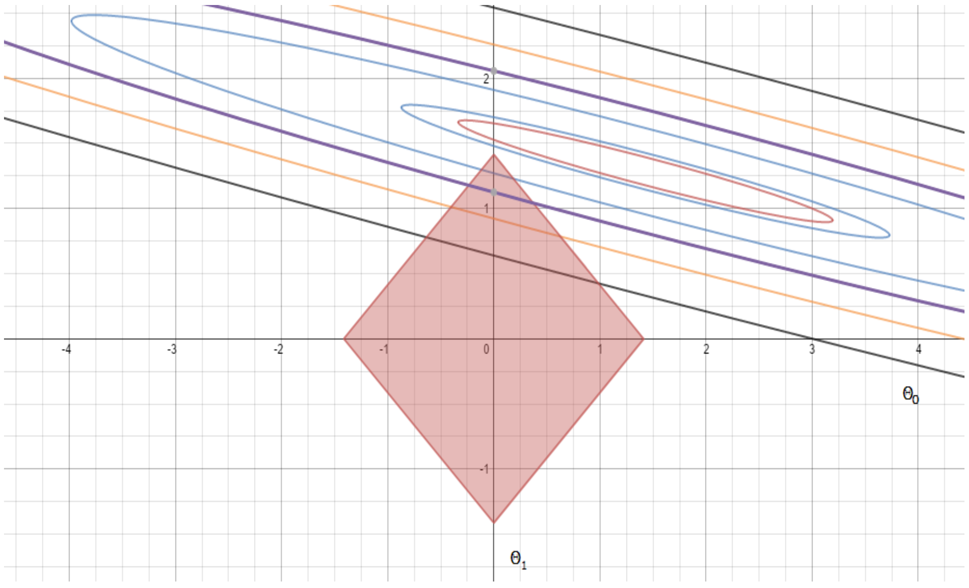
$$\min_{\beta \in \mathbb{R}^n} ||\vec{y} - A\vec{\beta}||_2^2.$$
$$\text{s.t.} \quad ||\vec{\beta}||_1 \leq t$$

where $\beta$ is the regression co-efficient.

To view the geometric picture using the same housing price example, the problem becomes

$$\min_{\theta_0, \theta_1 \in \mathbb{R}} \sum_{1}^{3} (y_i - (\theta_0 + \theta_1 x_i))^2 \quad \text{s.t.} \quad ||\theta||_1 \leq t$$

where $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ and $||\theta||_1$ is the $\ell_1$ norm. For $t = \sqrt{2}$, the picture is shown below:



This solution is $t$, as long as $t$ is small enough. In the above graph, it's obvious that if $t$ is smaller than about $3/2$, then the optimizer will be $(\theta_0, \theta_2) = (0, t)$. This is a "sparse" solution for this example. If $t$ is large, then the minimizer will not be sparse; it will be some $(\theta_0, \theta_1)$ where $\theta_0 \neq 0$ and $\theta_1 \neq 0$.

**Least Angle regression** is a way to solve the Lasso problem [2].

- Set all coefficients $\theta_i$ equal to zero.
- Find the predictor $x_i$ most correlated with $y$, and add it to the model.
- Increase the coefficient $\theta_i$ in the direction of the sign of its correlation with $y$. Take residuals $r = y - \hat{y}$ along the way.Stop when some other predictor $x_k$ has as much correlation with $r$ as $x_j$ has.
- Increase $(\theta_j, \theta_k)$ in their joint least squares direction, until some other predictor $x_m$ has as much correlation with the residual $r$.
- Continue until: all predictors are in the model.

## 5 Sparse coding and convex approximation

**Sparse coding** The definition of sparse approximation is to represent the signal as a sparse linear combination of atoms for a given dictionary. The dictionary means a matrix and atom means the columns of the dictionary. Mathematically, one way to formalize it is to give a signal $x \in \mathbb{R}^n$ and a dictionary $D \in \mathbb{R}^{n \times k}$, the sparse coding problem can be stated as

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad x = D\alpha$$

This problem is NP-hard [4], thus it is commonly approximated by substituting $\ell_0$-norm to $\ell_1$-norm. If the signal is noisy, the equality constraint is relaxed as well. A way to resolve these issues is to formulate the problem as a Lasso optimization problem

$$\min_{\alpha} ||x - D\alpha||_2^2 \quad \text{s.t.} \quad ||\alpha||_1 \le t$$

where $t$ is the regularization constant. This is a nice formulation in a way that it not only turns the problem into a convex optimization problem, but it also induces sparsity for the $\alpha$ coefficient vector.

## 6 Dictionary Learning for images

Take images $y_1, ..., y_M$, where each $y_i \in \mathbb{R}^n$. The goal is to make a matrix (called "dictionary") $A$ such that $y_1, ..., y_M$ can be "well approximate" by $Ax_i$ for some sparse vectors $x_1, ..., x_M \in \mathbb{R}^k$.

Moreover, we want to "choose" $A$ so that for each image sample $y_i$, it is possible find a sparse vector $x_i$ that makes the error $||y_i - Ax_i||_2^2$ small. To do this, we take $x_i$ to be the solution of the LASSO problem

$$x_i = \arg \min_{||x||_1 \leq t} ||y_i - Ax||_2^2$$

.

To put this in the setting of the classification algorithm, let's say we have $k$ number of images $y_1, ..., y_k$. We then have to train $k$ number of dictionaries such that each dictionary $D_i$ is trained using the training set $y_i$ so that each is a good for sparsely representing the image $y_i$.

## 7 An application of dictionary learning to character recognition

### 7.1 Overview of the implementation

The purpose of this task to train the Dictionary algorithm to recognize capital letters from A to Z. The programming language used for this task is Matlab. The data set of the letters was found on The Chars74K dataset. Each picture is reduce to size 20 by 20 using IrfanView64 software.

The letter images are stored under the letter-set folder. Capital letters are stored in the letter-set folders of those capital letter folders, and small letters are stored in double-small-letter folders. Note that although we have small letter data-set, this implementation does not include the small letters to our testing yet.

To run through the implementation, simply use the file "main.m" and execute the code line by line in Matlab. The variable $k$ is the default number of atoms in our dictionary, the variable lambda is the $\ell_1$ norm regularization constant. And the variable "num-test-sets" means the current complete training set numbers. We currently have 26 capital letters, so we set it as 26.

## 7.2 Training-set preparation

The file training-set.m contains the training-set function as it returns an array of cells, and each cell stores all the samples of that specific letter image. For example, let's say we have image of letters A to Z, letter A has 88 images, letter B has 109 images, and so on. The training-set function will return an array of cells, the first cell is will be letter A with 88 images in it, and the second cell will be letter B with 109 images in it, and so on. Aggregate all data into this simple data structure is a convenient for later use. At least for the scale of this project.

## 7.3 Dictionary Learning

To initialize the set of dictionaries that we will train, we initialize these dictionaries as random matrices that each element in the matrix is close to zero.

```
for i = 1:num_test_sets ,
    Dictionaries{i} = rand(n*p, k);
    Dictionaries{i} = Dictionaries{i} ./ 100;
end
```

This will give us a convex matrix (What is a convex matrix?), that satisfies def. (3) in Online Dictionary paper. And then, we set up the $\ell_1$ norm regularization term and train our dictionaries

```
lambda = 50;
for i = 1:num_test_sets ,
        t = training_sets{i};
        D = Dictionaries{i};
        Dictionaries{i} = dictionary_learning(t, D, lambda);
end
```

### 7.3.1 The learning part

To train our dictionaries to recognize each of the images, we implement the dictionary learning function in the file "dictionary-learning.m". Let $k \in \mathbb{N}$ be the number of image samples we have for each letter image, we train the designated dictionary $k$ numbers of times to update the dictionary through the use of least angle regression.

### 7.3.2 Problem

One problem that we've found in the algorithm is the updated part of the equation: $\arg\min_\alpha \frac{1}{t}(Tr(D^T D A_t) - Tr(D^T B))$ from the Bach-Ponce-Sapiro paper (algorithm 1). The problem is that if we update the dictionary multiple times each round, it causes the Least Angle Regression algorithm unable to evaluate the inverse of the gram vector. Since the Bach-Ponce-Sapiro paper indicated that the dictionary updated step is empirically found that one update is enough for each run, we leave the dictionary update step to run only once in each image training session. We did not setup the updated step to optimize the dictionary until convergence.

## 7.4 Error estimation

Once we've done the learning, we are ready to test our dictionaries. To test our dictionary, we feed an image that is not in our training set and run that image with Least Angle Regression to generate different regression coefficients with respect to each dictionary.

There are two ways to find the correlation of two image, one is to find the minimal square error between the test image and the other is to use normalized 2-D cross-correlation.

### 7.4.1 Minimum square error

Minimum square error is to calculate the error $||x - D\alpha||_2^2$ where $D$ is the dictionary, $\alpha$ is the regression coefficient, and $x$ is the signal that we feed into the learning algorithm. This is implemented in the "err.m" file, and the code is:

```
for i = 1:num_test_sets,
    D = Dictionaries{i};
    a = alphas{i}
    % error
    err(i)=norm(D*a-image_vec)
end
```

We then simply find the minimum, and return the result.

## 8 Conclusions

The dictionary learning is able to recognize letters, through finding the minimum squared error by running least angle regression using input signal with each of the dictionaries we've generated. However, the accuracy of the result could still be improved.

We consider that improvement could be pursue in the following directions:

- Use letters that are typed rather than the hand-written ones (tried)
- Explore if there are other means to resale each element's value in the the reconstruction $D\alpha$ to $0.0 - 255.0$ without using matlab's built-in function "mat2gray". Our comparisons have shown that "mat2gray" function may cause the image more blurred.
- Further examine the algorithm's implementation details (7.3.2)
- increase the sample size and use over-complete dictionaries.

## References

[1] "Machine Learning: What It Is and Why It Matters." Analytics, Business Intelligence and Data Management. Web. 06 May 2016.

[2] "A Simple Explanation of the Lasso and Least Angle Regression." Home. Web. 06 May 2016.

[3] Boyd, Stephen, and Lieven Vandenberghe. Convex Optimization. Cambridge UP. Print.

[4] Mairal, Julien, Francis Bach, Jean Ponce, and Guillermo Sapiro. "Online Dictionary Learning for Sparse Coding." Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09 (2009). Web.