Basics of L1 Regularization

Issues we may face with L2 regularization:

Consider the case where we have 2 variables in a weight vector, say \( w_1 \) and \( w_2 \) that are highly correlated. If \( w_1 \uparrow \), \( w_2 \downarrow \), in a way, canceling the effect of \( w_1 \). Models that can have high variance, implying different predictions under the RSS (residual sum of squares).
We considered the regression problem earlier

\[ X := \begin{bmatrix} x_{ij} \end{bmatrix}_{n \times k} \quad \forall i \in \mathbb{R}^k \]

- \( y \) data response

\[ y \in \mathbb{R}^n \]

\[ \frac{w}{(X^T X + \lambda I)^{-1} X^T y} \quad \text{ (min - norm soln)} \]

Cost functional: \[ \| X w - y \|_2^2 + \lambda \| w \|_2^2 \]

(a) \( w \) must not be too large

(b) Balance the large variables in \( w \) meeting the target cost.
1) L₂ norm does not account for the parsimony of the model, i.e., sparsity constraints are not taken into account.

2) L₂ models may have non-zero values associated with inconsequential variables.

Costs involving L₁ penalty impose sparsity constraints $\| w \|_1$.
1) If the data matrix $X_{(n \times k)}$ has irrelevant features, $L_1$ seems to be better than $L_2 \Rightarrow$ low variance, feature selection

2) $L_1$ can yield a better variable/attribute selection
   a) Simplification of models for interpretability.
   b) Shorter training times
   c) Avoid the problem of overfitting $\Rightarrow$ avoid the curse of dimensionality
Unconstrained formulation

\[
\min_{w} \| Xw - y \|_2^2 + \lambda \| w \|_1
\]

1. \( L_1 \) norm

Clearly (1) has issues of differentiability at the origin.

\[ \| w \|_1 = |w_1| + |w_2| + \ldots + |w_k| \]

\( |x| \) is not differentiable at \( x = 0 \).
Constrained Formulation

\[
\min_{w} \| xw - y \|_2^2 \quad \text{s.t.} \quad \| w \|_1 \leq t \quad \overset{\uparrow}{\Rightarrow} \quad \text{chosen value}
\]

Non-differentiable constraints are converted to a set of linear constraints

\[
\Rightarrow \quad \text{feasible region is a polyhedron!}
\]
Consider solving the problem (2).
Suppose we have 2 variables in $\omega$, say $w_1$ and $w_2$. 

**LASSO**

**ALGORITHMS**

Least Absolute Selection and Shrinkage Operator.
By considering the sign of $w_1$ and $w_2$,

$$
\begin{align*}
 w_1 + w_2 &\leq t \\
 w_1 - w_2 &\leq t \\
 -w_1 + w_2 &\leq t \\
 -w_1 - w_2 &\leq t
\end{align*}
$$

Home Work:

Plot the constraints on $w_1 - w_2$ plane and show the feasible region.
Any minimizer to the RSS subject to (1)
will minimize the cost (2)

Problem: If we have \( k \) variables, we have \( 2^k \) constraints \( \Rightarrow \) exponential increase in complexity.

Over \( \mathbb{R}^{40} \), \( 2^{40} \) are possible (Infeasible for optimization)
JIBSHIRANI'S APPROACH

Constraint Set = ∅  (In practice 't' can be small)

while ( || w ||₁ ≤ t )

. Add sign (w) and fold this into the constraint set.

. Opt || X w - y ||₂² subject to the constraints.

end while
Instead of testing if $\|w\|_1 < t$, we can introduce $\varepsilon > 0$ and we consider $\|w\|_1 \leq t + \varepsilon$.

At every iteration, $\|w\|_1$ shrinks.

1) The solution from a previous iteration may not be suited to the present constraint $\Rightarrow$ One needs to optimize again.

2) Adding the sign constraint can have variables that can have large swings from $+ve$ to $-ve$. (Correlated variables)
Introduce non-negative variables

Express each $w_i$ as a difference of two non-negative variables

$$w_i = w_i^+ - w_i^-$$

$$\begin{cases} w_i^+ > 0 & (w_i^+ > 0) \\ w_i^- > 0 & (w_i^- > 0) \end{cases}$$

If $w_i > 0$; $w_i^+ = w_i$, $w_i^- = 0$

If $w_i < 0$; $w_i^+ = 0$, $w_i^- = w_i$

$w_i^+ = w_i^- = 0$ if $w_i = 0$
For $k$ variables in $w$, we introduce $2k$ non-negative variables $\Rightarrow$ introducing degeneracy in the constraints.

For e.g., if we consider a 2 variable case

\[ w_1^+ \geq 0, \quad w_1^- \geq 0 \]
\[ w_2^+ \geq 0, \quad w_2^- \geq 0 \]
\[ \sum_{i=1}^{2} (w_i^+ + w_i^-) \leq t \]
Grafting: (Perkins et al., JMLR)

Idea: Incrementally build a subset of parameters allowed to differ from 0s.

At each iteration, we use a fast grad. meta heuristic to decide which zero wt. should be adjusted away from zero to decrease the opt. criterion by max. amount.
Recall: \[ \| Xw - y \|_2^2 + \lambda \| w \|_1, \]

\[ \nabla_w = X^T (y - Xw) + \lambda \text{sign}(w) \]

For variables that are zero, \( w_i = 0 \)

\( \text{sign}(w_i) = 1 \) if \( X_i^T (y - Xw) > \lambda \)

By convention, if \( X_i^T (y - Xw) = 2 \), grad is set to 0
A procedure can be evolved as follows

1) Consider all variables in the zero set initially. At each iteration, test if

$$|X_i^T(y - Xw)| \leq \lambda$$

for each 'i'.

2) If Condition (A) is false, the variable whose derivative has the largest mag. is added to the free set.

3) Any popular method (QN, BFGS algo) can be used to optimize the variables in the free set.
Discussion on VC-dimension

Defn: A dichotomy of a set 'S' is a partition of S into 2 disjoint subsets

\[ S = \{ x_1, x_2, \ldots, x_{100} \} \]
\[ S_x = \{ x_1, x_3, \ldots, x_{99} \} \]
\[ S_o = \{ x_2, x_4, \ldots, x_{100} \} \]

(2-class problem)
Defn: A set of instances $S$ is 'shattered' by a hypothesis space $\mathcal{H}$ iff for every dichotomy of $S$, $\exists$ a hypothesis that is consistent with the dichotomy.

$\exists h \in \mathcal{H} / h_1$ classifies $S_x$ as $\rightarrow 1$, $h_2$ classifies $S_o$ as $\rightarrow 2$. 
Motivate the notion of VC-dimension

Consider points on a line (points $\in S_x$ or $S_0$)

```
  x   0   0   x
```

2 points on a line

"Can shatter the 2 points on a line"
Let us consider 3 points on a line.

Works! (c)

8 different configurations over 3 points on a line.

Problem cases (d) (e)
Let us consider points in \( \mathbb{R}^2 \), i.e., on a plane.

[Diagram showing points on a plane with annotations: 3 points can be shattered in \( \mathbb{R}^2 \).]
Let us increase by \('1\) extra point
i.e., \(4\) points in \(\mathbb{R}^2\)

\[
\begin{array}{cccc}
\times & \times & \times & \checkmark \\
\circ & \circ & \circ & \checkmark \\
\times & \checkmark & \circ & \checkmark \\
\circ & \times & \times & \times \\
\end{array}
\]

(Not possible)
If I consider a hyperplane for shattering points over d-dimensional
\[ \mathbf{w}^T \mathbf{x} = 0 \]
Expanding out
\[ \sum_{i=1}^{d} w_i x_i + w_0 \cdot 1 = 0 \]
Projection of \( d \) features onto a vector normal to the plane to accommodate bias

The number of points that can be shattered in \( \mathbb{R}^d \) is \( d+1 \)!
Defn: The VC dimension of a hypothesis space $\mathcal{H}$ defined over a data set $X$ is the size of the largest subset of $X$ shattered by $\mathcal{H}$.

The reader can refer to the PAC bound derivation (probably approximately correct) in any standard M.L. text book.
For linear classifiers with $x \in \mathbb{R}^d$

\[ VC(H) = d + 1 \]

$d$ : # of features

For neural networks

\[ VC(H) = \# \text{ parameters in the } n/w \]

$L$ : # of neurons in a hidden layer

for a single hidden layer MLP.
Auto Encoders

This is a neural network for data encodings and to learn a representation of a data vector in a reduced dimension to ignore signal "noise".

Idea: We have a sequence of layers from the i/p to learn local features all the way to less local features and eventually the object. Decoding layers will decode the learnt encoded information.
Structure

Layers

Encoded information

Typically, we use feed forward N/w such as MLP

0/p has the same # nodes as the input
Let $\phi_E$ and $\phi_D$ be non-linear mappings

$\phi_E : \mathbb{R}^d \rightarrow \mathbb{R}^k$

$\phi_D : \mathbb{R}^k \rightarrow \mathbb{R}^d$

$\phi_E^*, \phi_D^* = \arg \min E \left\| x - (\phi_D \circ \phi_E)(x) \right\|_2$

$x \in \mathbb{R}^d = \mathcal{X}$

$z \in \mathbb{R}^k = \mathcal{F}$

$k < d$
For purposes of simplicity, let us consider a single hidden layer.

\[ z = \sigma(W^T x + b) \]

- Sigmoid / ReLU
- Coded representation / latent representation
Reconstruction
\[
\hat{x} = \sigma (W^T z + b)
\]

Loss function
\[
L (x, \hat{x}) = \| x - \hat{x} \|^2
\]

\[
= \| x - \sigma (W^T (\sigma (W^T x + b)) + b) \|^2
\]

One can also minimize the "average loss" \( \hat{x} \) computed by taking the \( E (\cdot) \) over \( L (x, \hat{x}) \)
Denoising autoencoder

Idea: Take a partially corrupted input during training to recover the original undistorted input data vector.

Hope: We have an efficient representation that can robustly obtain the clean input from a corrupted representation!
Assumptions (JMLR, Vincent et al)

1) Higher level representations are robust and stable

2) Extract features that are useful to represent the original data (pdf).
Stochastic corruption

\[ S_c: \ x \rightarrow \tilde{x} \]

Loss is \( L(x, \tilde{x}) \)

Feed \( \{\tilde{x}\} \) to the auto encoder for learning the encoded version of the corrupted input (details).

Original uncorrupted input
Sparsity constraint with the N.N.

More hidden units than \(i/p\)s but very few of them could be "active".

One can bring in "regularization" constraint.
Geometric Interpretation

\( q_D (\tilde{x} | x) \)

\( g_\theta (f_\theta (\tilde{x})) \)

\( M \) manifold

Stochastic corruption of \( x \) lie away from the manifold \( M \)

Denoising

Interpret it as a coord. sys. for points \( x \) on the manifold

\( \dim (f_\theta (\cdot)) \)

\( < \dim (x) \)