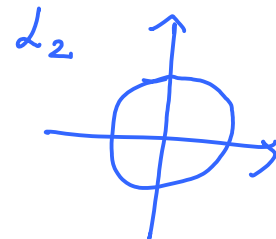


Basics of L1 Regularization



Issues we may face with L_2 regularization:

Consider the case where we have 2 variables in a weight vector, say w_1 and w_2 that are highly correlated.

If $w_1 \uparrow\uparrow$, $w_2 \downarrow\downarrow$, in a way, canceling the effect of $w_1 \Rightarrow$ models that can have high variance, implying different predictions under the RSS (residual sq. sum)

We considered the regression problem earlier

$$X := [x_{ij}]_{n \times k}$$

$$\underline{x}_i \in \mathbb{R}^k$$

$$\underline{y}_{n \times 1} \quad \text{data response}$$

$$\underline{w} = \left(X^T X + \lambda I \right)^{-1} X^T \underline{y} \quad \left(\begin{array}{l} \text{min-norm} \\ \text{soln} \end{array} \right)$$

$$\text{Cost functional: } \left\| X \underline{w} - \underline{y} \right\|_2^2 + \lambda \left\| \underline{w} \right\|_2^2$$

(a) \underline{w} must not be too large

(b) Balance the large variables in \underline{w} meeting the target cost.

L_2 norm

1) does not account for the parsimony of the model
i.e., sparsity constraints are not taken into account.

2) L_2 models may have non-zero values associated with
inconsequential variables.

Costs involving L_1 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

$$\min_{\underline{w}} \left\| X \underline{w} - \underline{y} \right\|_2^2 + \lambda \left\| \underline{w} \right\|_1$$

L_1 norm

Clearly (1) has issues of differentiability @ the origin

$$\left\| w \right\|_1 = |w_1| + |w_2| + \dots + |w_k|$$

$|x|$ is not differentiable @ $x = 0$.

Constrained Formulation

$$\min_{\underline{w}} \quad \|X\underline{w} - y\|_2^2 \quad \text{s.t.} \quad \|\underline{w}\|_1 \leq t \quad \text{--- } \textcircled{2}$$

↑
Chosen value

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

⇒ Feasible region is a polyhedron!

ALGORITHMS

LASSO

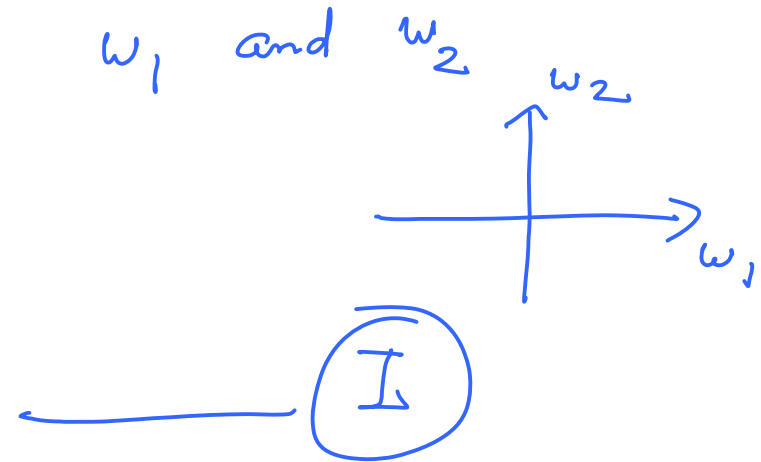
Least Absolute Selection and Shrinkage Operator.

Consider solving the problem (2).

Suppose we have 2 variables in \underline{w} , say w_1 and w_2

By considering the sign of w_1 and w_2

$$\begin{array}{rcl} w_1 + w_2 & \leq & t \\ w_1 - w_2 & \leq & t \\ -w_1 + w_2 & \leq & t \\ -w_1 - w_2 & \leq & t \end{array}$$



Home Work

Plot the constraints on $w_1 - w_2$ plane and show the feasible region.

Any minimizer to the RSS subject to (I)
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 = ϕ

(In practice 't' can be small)

while ($\| \underline{w} \|_1 \leq t$)

. Add sign (\underline{w}) and fold this into the constraint set.

. Opt $\| X \underline{w} - \underline{y} \|_2^2$ subject to the constraints.

end while

Instead of testing if $\|w\|_1 \leq t$, we can introduce $\epsilon > 0$ / we consider $\|w\|_1 \leq t + \epsilon$

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

- 1) The soln from a previous iteration may not be suited to the present constraint \Rightarrow One needs to optimize again
- 2) Adding the sign constraints can have variables that can have large swings from +ve / -ve. (Correlated variables)

Introduce non-negative variables

Idea:

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

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

$$\begin{pmatrix} w_i^+ \geq 0 \\ w_i^- \geq 0 \end{pmatrix}$$

$$\text{If } \left. \begin{array}{l} w_i > 0; \quad w_i^+ = w_i; \quad w_i^- = 0 \\ w_i < 0; \quad w_i^+ = 0; \quad w_i^- = w_i \\ w_i^+ = w_i^- = 0 \quad \text{if} \quad w_i = 0 \end{array} \right\}$$

For k variables in \underline{w} , we introduce $2k$
non negative variables \Rightarrow introducing degeneracy
in the constraints

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

$$w_1^+ \geq 0$$

$$w_1^- \geq 0$$

$$w_2^+ \geq 0$$

$$w_2^- \geq 0$$

$$\sum_{i=1}^2 (w_i^+ + w_i^-) \leq t$$

Grafting : (Perkins et al, JMLR) 

Idea : Incrementally build a subset of param. 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 : $\|X\underline{w} - y\|^2 + \lambda \|\underline{w}\|_1$

$$\nabla_{\underline{w}} = X^T (y - X\underline{w}) + \lambda \text{sign}(\underline{w})$$

For variables that are zero, $w_i = 0$
 $\text{sign}(w_i) = 1$ if $\underline{x}_i^T (y - X\underline{w}) > \lambda$

By convention, if $\underline{x}_i^T (y - X\underline{w}) = \lambda$, grad is set to 0

the data point i.e., attribute 'i' over all data

A procedure can be evolved as follows

1) Consider all variables in the zero set initially

At each iteration, test if

2)
$$\left| X_i^T (y - X \underline{w}) \right| \leq \lambda \quad \text{for each 'i'}$$

$$\text{(A)}$$

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

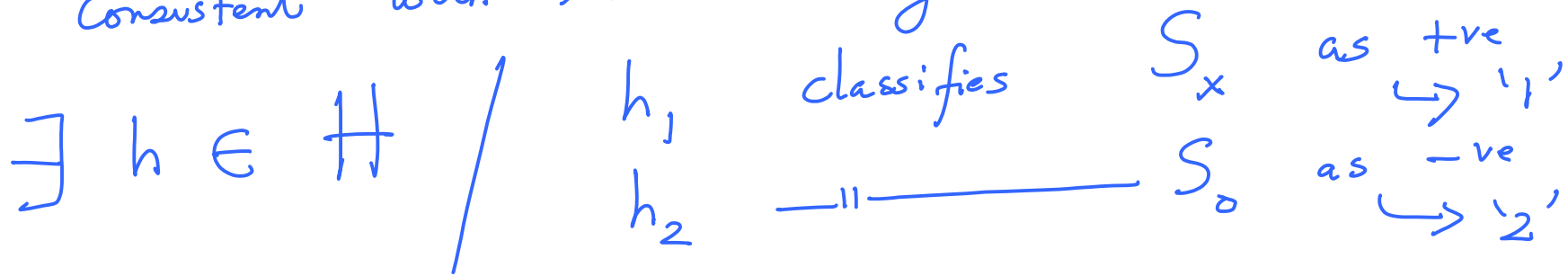
$$S = \{ \underline{x}_1, \underline{x}_2, \dots, \underline{x}_{100} \}$$

$$S_x = \{ \underline{x}_1, \underline{x}_3, \dots, \underline{x}_{99} \}$$

$$S_o = \{ \underline{x}_2, \underline{x}_4, \dots, \underline{x}_{100} \}$$

(2-class problem)

Defn : A set of instances S is shattered by a hypothesis space H iff for every dichotomy of S \exists a hypothesis that is consistent with the dichotomy



Motivate the notion of VC-dimension

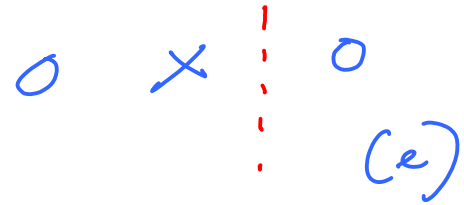
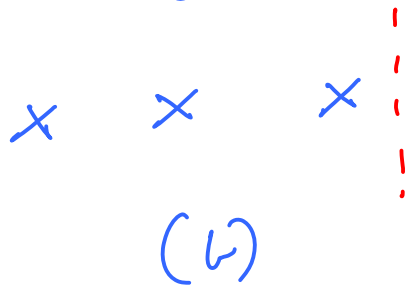
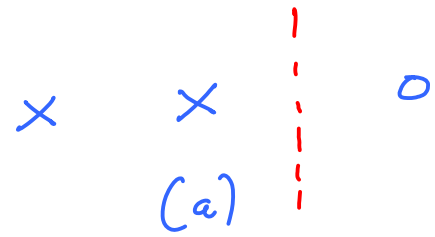
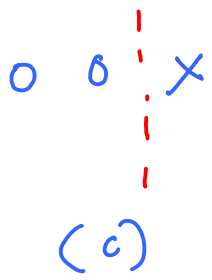
Consider points on a line (points $\in S_x$ or S_o)



2 points on a line

"Can shatter
the 2 points
on a line"

Let us consider 3 points on a line



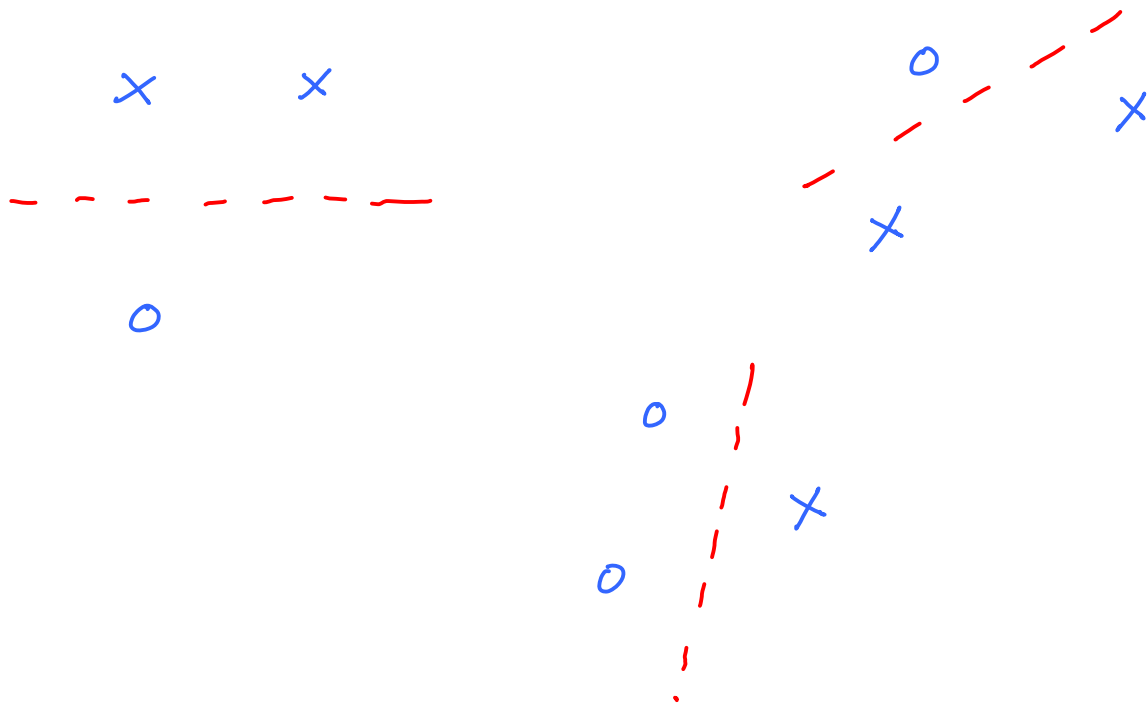
Works!

Problem cases

" 3 different

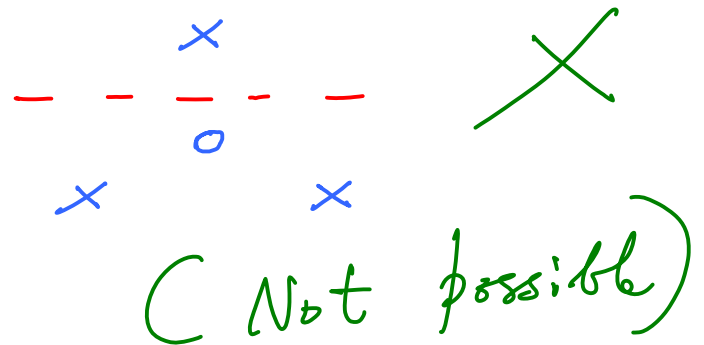
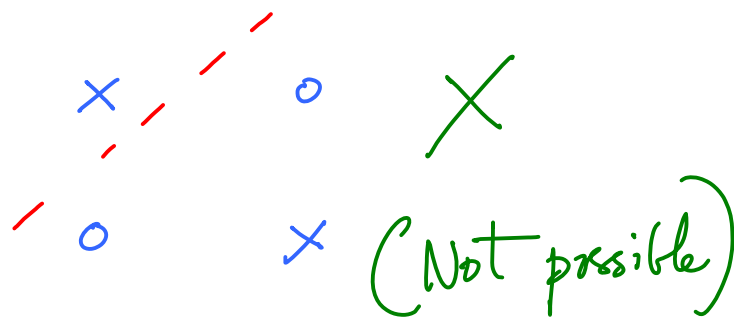
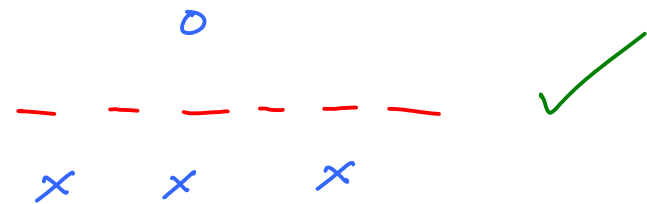
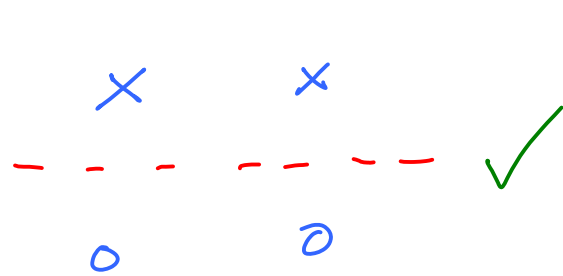
configurations over 3 points on a line "

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



3 points can
be shattered
in \mathbb{R}^2

Let us increase by '1' extra point
i.e., 4 points in \mathbb{R}^2



If I consider a hyperplane for shattering
points over d -dimensions,

$$\underline{w}^T \underline{x} = 0$$

Expanding out

$$\sum_{i=1}^d w_i x_i + w_0 \cdot 1 = 0$$

projection of
a vector normal to the plane
of d features on to
a vector normal to the plane

to accommodate bias

points that
can be shattered
in \mathbb{R}^d is

$$\underline{\underline{d+1!}}$$

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 $\underline{x} \in \mathbb{R}^d$

$$VC(H) = d + 1 ; \quad d : \# \text{ of features}$$

For neural networks

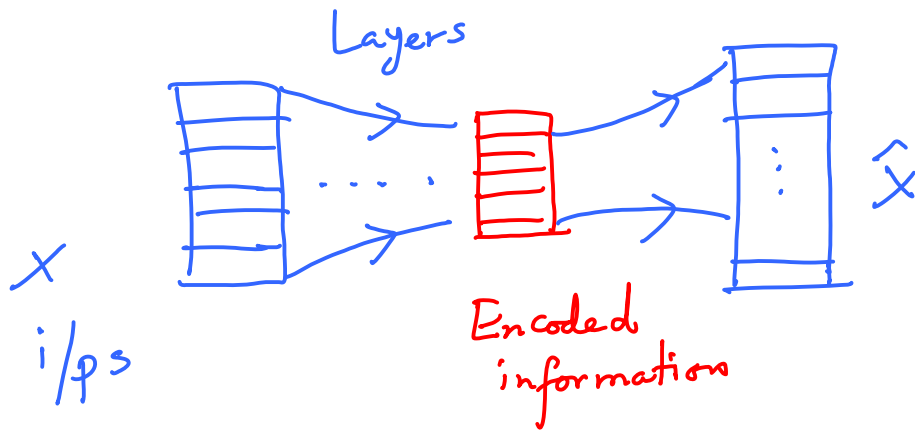
$$VC(H) = \# \text{ parameters in the n/w} \\ \leq \# \text{ of neurons in a hidden layer} \\ \text{for a single hidden layer MLP.}$$

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



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

O/p has the same # nodes as the inputs


Let ϕ_E and ϕ_D be non-linear mappings

$$\phi_E : \mathcal{X} \longrightarrow \mathcal{F}$$

$$\phi_D : \mathcal{F} \longrightarrow \mathcal{X}$$

$$\begin{aligned} \underline{x} &\in \mathbb{R}^d = \mathcal{X} \\ \underline{z} &\in \mathbb{R}^k = \mathcal{F} \\ &k < d \end{aligned}$$

$$\phi_E^*, \phi_D^* = \arg \min_{\phi_E, \phi_D} \left\| \mathcal{X} - \left(\phi_D \circ \phi_E \right) \mathcal{X} \right\|^2$$


order of composition

For purposes of simplicity, let us consider a single hidden layer

$$\underline{z} = \sigma \left(\underline{w}^T \underline{x} + b \right)$$

Sigmoid / ReLU

coded representation /
latent representation

Re construction

$$\hat{x} = \sigma \left(\hat{w}^T z + \hat{b} \right)$$

Loss function

$$L(x, \hat{x}) = \|x - \hat{x}\|^2$$
$$= \left\| x - \underbrace{\sigma \left(\hat{w}^T (\sigma(w^T x + b)) + \hat{b} \right)}_{\hat{x}} \right\|^2$$

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

Denoising auto encoder

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

Hope : We have an efficient representation that can robustly obtain the clean i/p 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

$$S_c: \underline{x} \longrightarrow \underline{\tilde{x}}$$

Feed $\{\underline{\tilde{x}}\}$ to the auto encoder for learning

Loss is $L(\underline{x}, \underline{\hat{\tilde{x}}})$

decoded o/p based on the encoded version of the corrupted inputs (detail!)

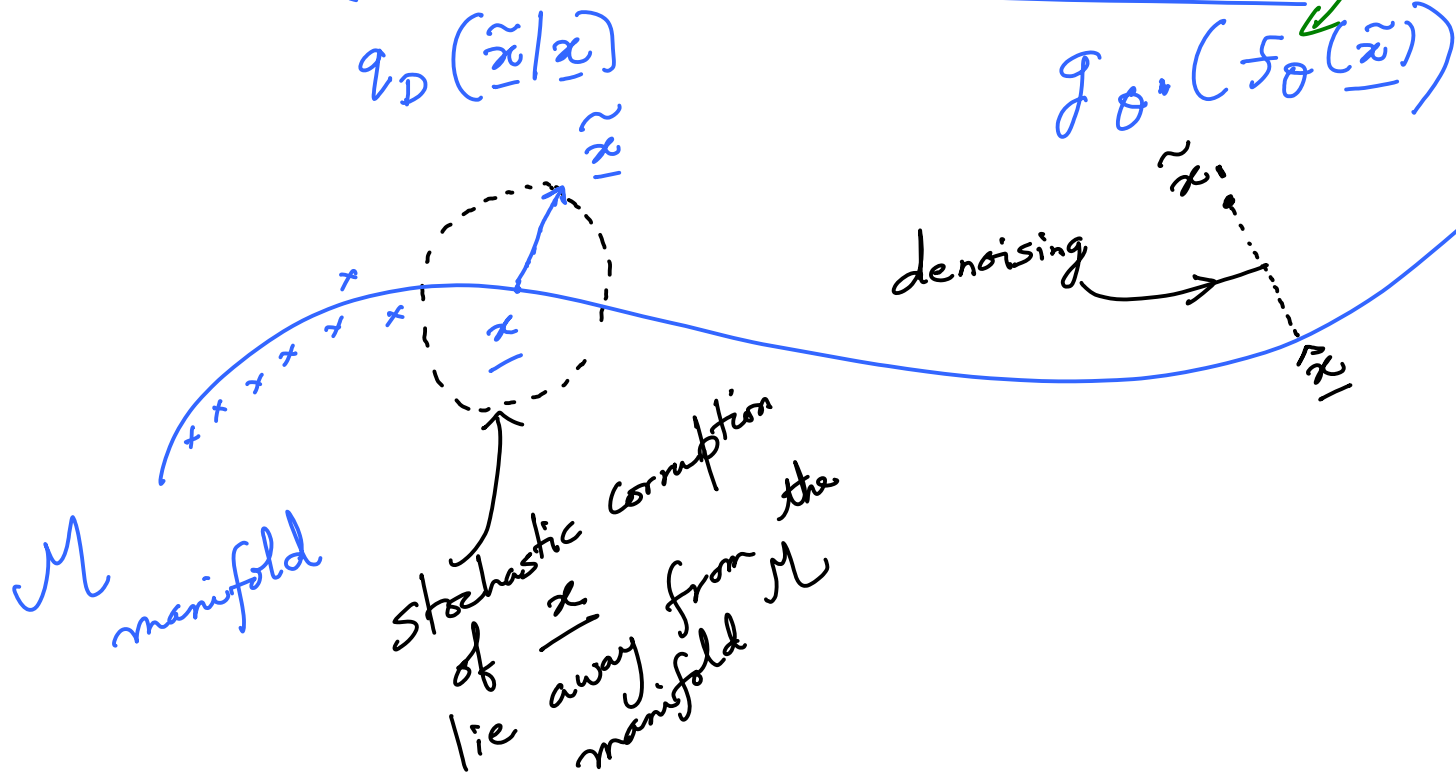
original uncorrupted i/p

Sparsity constraints with the N.N.

More hidden units than i/ps
but very few of them could be "active"

One can bring in "regularization" constraints

Geometric Interpretation



Interim representation
↓

Interpret it as a coord. sys. for points \underline{x} on the manifold

$$\dim(f_0(\cdot)) < \dim(\underline{x})$$