Discussion Section – Week 3

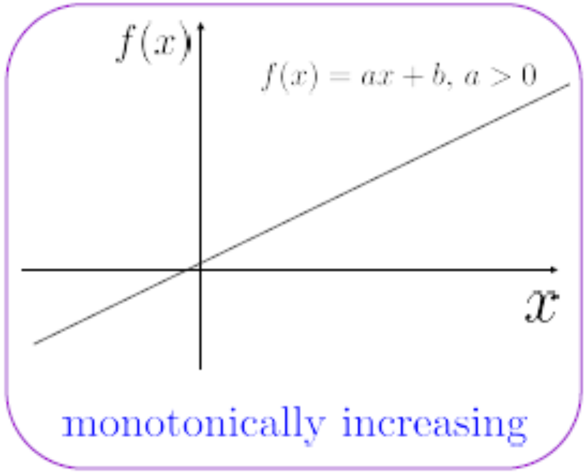
10/21/2020

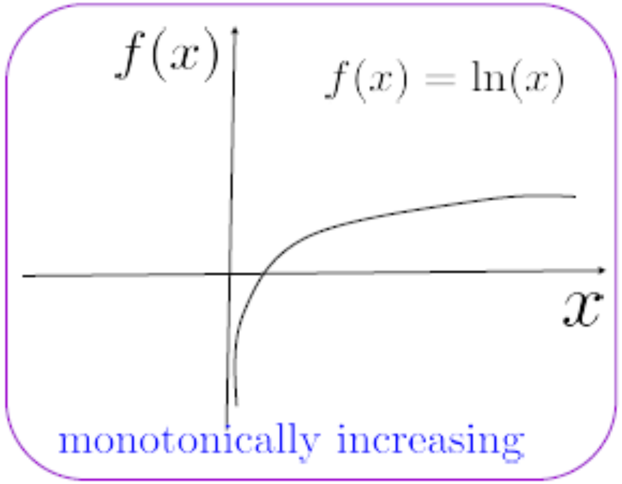
Xuehai He

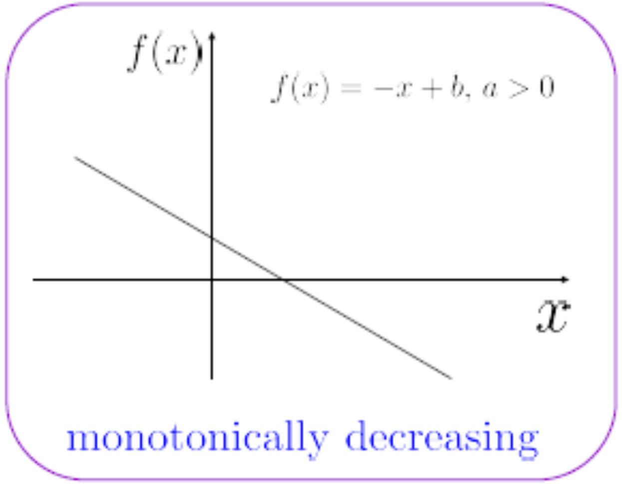
What we have learnt so far?

- Monotonic functions
- Optimization & Convexity
- Linear Regression with OLS
  - Polynomial Regression
- Robust estimation
  - Norms
  - Regularization
- Error metrics

Monotonic functions







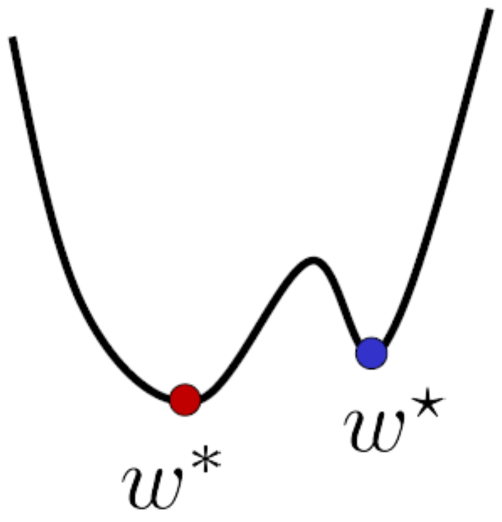
Brain teaser: Is f(x) = 5 monotonic?

Why apply log to ML problems?

In general: $W^* = \arg\min_W \mathcal{L}(W)$, where $\mathcal{L}(W) = e_{training}$ defines a loss/objective function in machine learning.

- Provides numeric stability & prevent the dealing with too large or too small numbers.
- If L(w) -> ln(L(w)), the landscape of L(w) is still retained because ln is a monotonic function.

Optimization



“Set of optima”

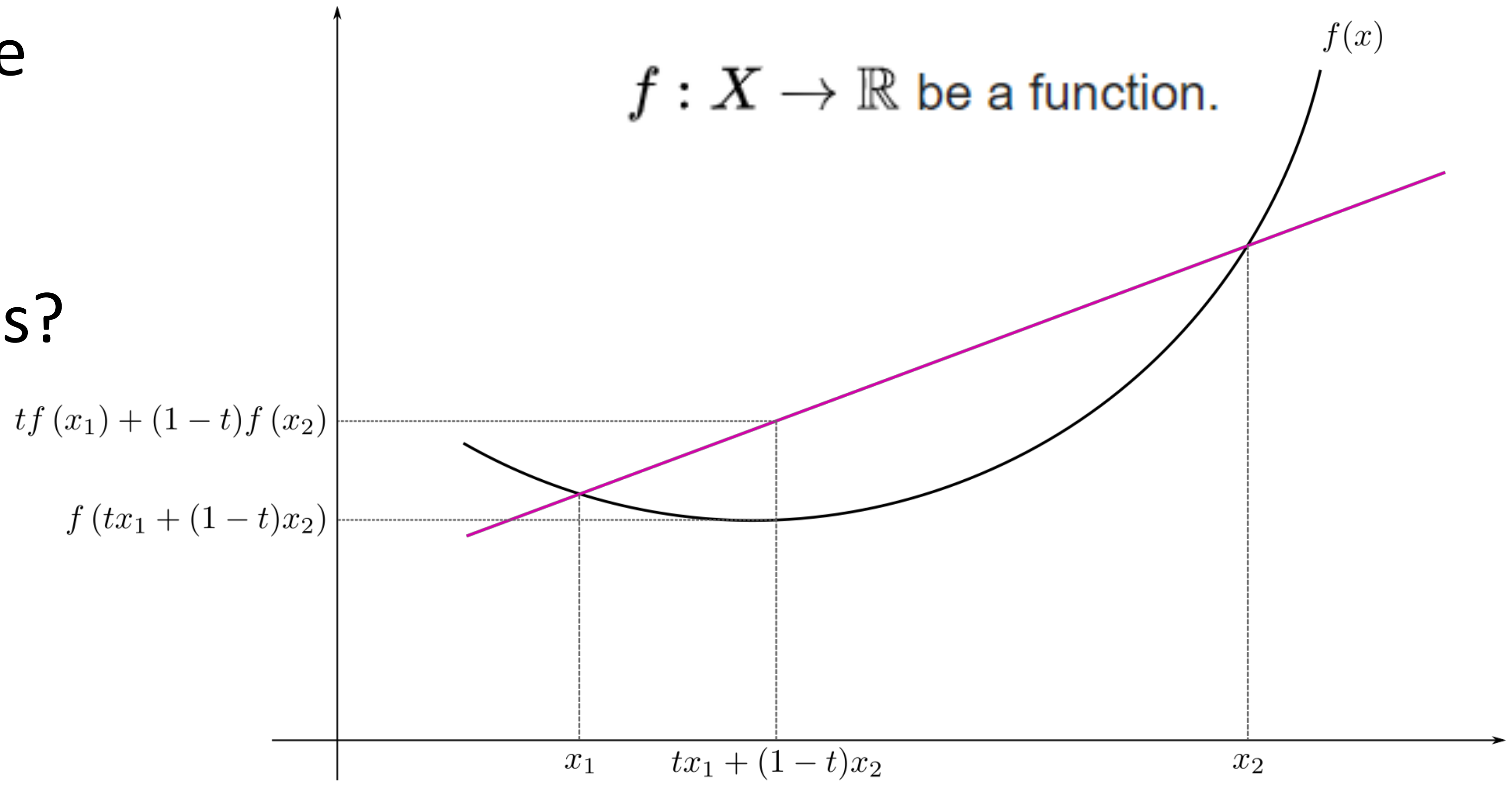
Definition:

1. w^* is a globally optimal solution for $w^* \in \Omega$ and $L(w^*) \leq L(w) \forall w \in \Omega$
2. $w^\star$ is a locally optimal solution if there is a neighborhood $\mathcal{N}$ around w such that $w^\star \in \Omega$, $L(w^\star) \leq L(w)$, $\forall w \in \mathcal{N} \cap \Omega$.

You can end up in local minima very easily when optimizing!!!

Guaranteed optimum & Convexity

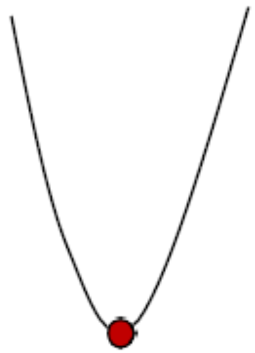
- When the loss function is convex, it is guaranteed that we would converge to the global optimum.
- Closed form solution if the function is differentiable at all points.
- What are convex functions?



f is convex if,

$$\forall x_1, x_2 \in X, \forall t \in [0,1] : \qquad f(tx_1 + (1-t)x_2) \le tf(x_1) + (1-t)f(x_2)$$

Linear Regression with OLS



Obtain/train: $f(x, W) = w_0 + w_1 x$

$$W^* = \arg min_W \sum_i (\mathbf{x}_i^T \cdot W - y_i)^2$$

$$W = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \quad \mathbf{x}_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$$

$$W^* = \arg min_W = \arg min_W L(W) = (XW - Y)^T (XW - Y)$$

$$L(W) = W^T X^T X W - W^T X^T Y - Y^T X W + Y^T Y$$

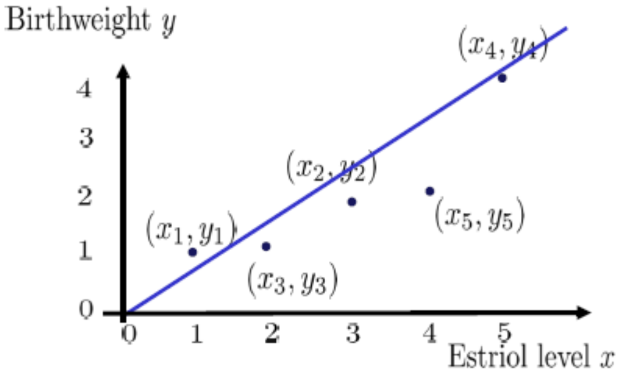
$$\frac{dL(W)}{dW} = 2X^T X W - 2X^T Y = 0$$

$$W^* = (X^T X)^{-1} X^T Y$$

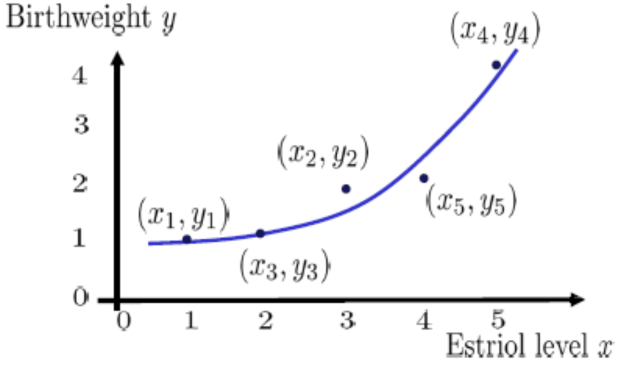
We did this last time!

Polynomial Regression with OLS

- Which **curve** best fits the data?



Linear



Polynomial

What's the optimum W?

$$S_{training} = \{(x_i, y_i), i = 1..n\}$$

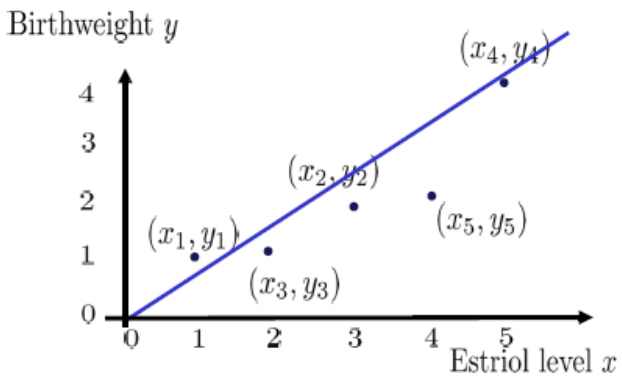
Input: x, $x \in R$

Model parameter: $\mathbf{w} = (w_0, w_1, ..., w_d)$, $w_i \in R$

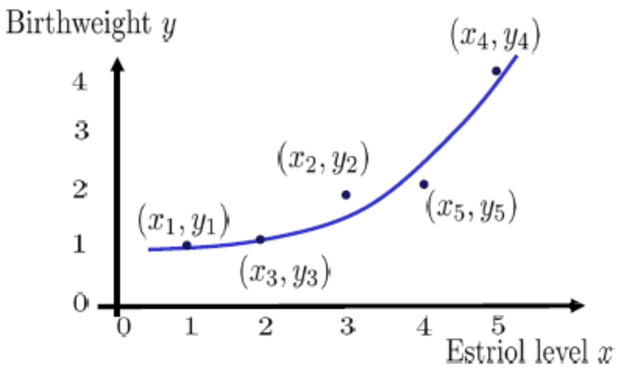
Output: $y = w_0 + w_1 x_1 + w_2 x_1^2 + ... + w_q x_m^q$

Polynomial Regression with OLS

- Which **curve** best fits the data?



Linear



Polynomial

$$S_{training} = \{(x_i, y_i), i = 1..n\}$$

Input: $x,\ x \in R$

Model parameter: $\mathbf{w} = (w_0, w_1, ..., w_d),\ w_i \in R$

Output: $y = w_0 + w_1 x_1 + w_2 x_1^2 + ... + w_q x_m^q$

What's the optimum W?
It is still the same optimization problem except W and X have different elements & shape.

$$W^* = \arg min_W = \arg min_W L(W) = (XW - Y)^T (XW - Y)$$

$$L(W) = W^T X^T X W - W^T X^T Y - Y^T X W + Y^T Y$$

$$\frac{dL(W)}{dW} = 2X^T X W - 2X^T Y = 0$$

$$W^* = (X^T X)^{-1} X^T Y$$

In general, linear regression with least squares estimation

$$Y = X \quad W$$

linear w.r.t. the W!

With an analytical solution:

$$W^* = (X^T X)^{-1} X^T Y$$

Robust Estimation – penalize data

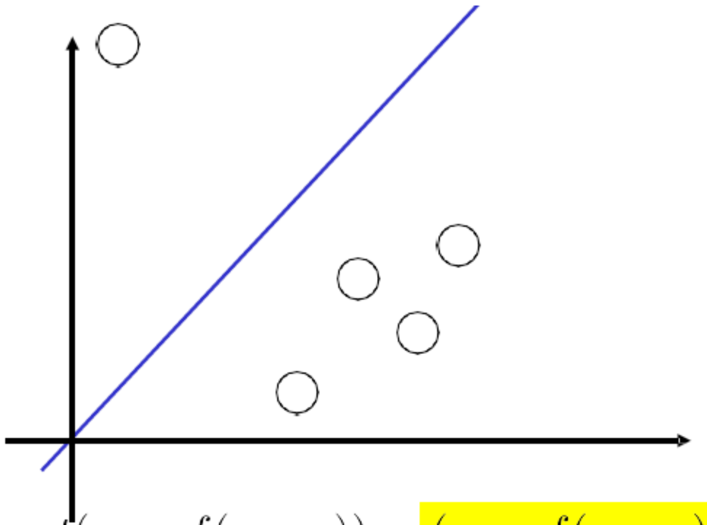
- How much importance is given to the outliers?
- If loss = L2 norm (MSE), outliers have large impact
- If loss = L1 norm (MAE), outliers don't have such a large impact

L2 norm: (squared)

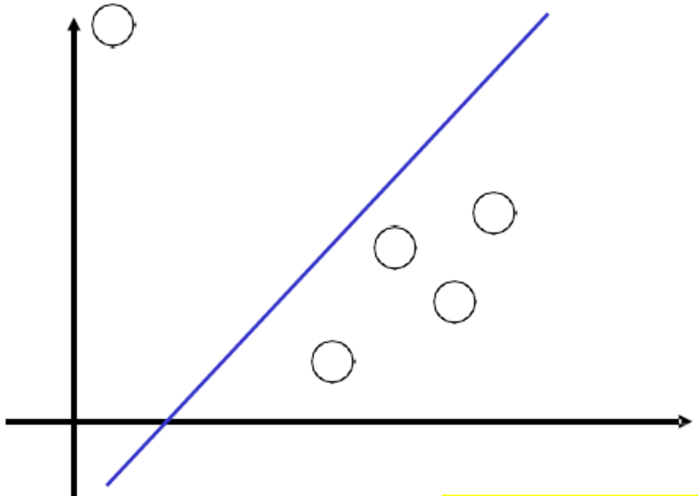
$$e = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \theta))^2$$

L1 norm:

$$e = \sum_{i=1}^{n} |y_i - f(\mathbf{x}_i; \theta)|$$



$cost(y_i - f(\mathbf{x}_i; \mathbf{w})) = (y_i - f(\mathbf{x}_i; \mathbf{w}))^2$



$cost(y_i - f(\mathbf{x}_i; \mathbf{w})) = |y_i - f(\mathbf{x}_i; \mathbf{w})|$

Optimizing L1 norm (need for gradient descent)

1. Loss (Cost) Function

$$L(W) = \sum_{i=1}^{n} |\mathbf{x}_i^T W - y_i|$$

2. Obtain the gradient

$$\frac{\partial L(W)}{\partial W} = \sum_{i=1}^{n} sign(\mathbf{x}_i^T W - y_i) \times \mathbf{x}_i$$

3. Update parameter W

$$W_{t+1} = W_t - \lambda_t \frac{\partial L(W)}{\partial W}$$

L2 loss -> convex & differentiable at all points -> w* is the global optimum and we can get a closed form solution.

L1 loss -> convex but not differentiable at origin -> though there exists a global minima (at origin) it cannot be expressed in closed form -> hence, perform gradient descent to update weights and reach close to w*

Robust estimation – penalize model (regularization)

$$L(\mathbf{w}) = (X\mathbf{w} - \mathbf{y})^T(X\mathbf{w} - \mathbf{y}) + \frac{\lambda}{2}\|\mathbf{w}\|_2$$

Let's try optimizing this loss! Grab a pen & paper ☺

$$L(w) = (X\vec{w} - \vec{y})^T (X\vec{w} - \vec{y}) + \frac{\lambda}{2} \|\vec{w}\|_2$$ just squaring

$$= (X\vec{w} - \vec{y})^T (X\vec{w} - \vec{y}) + \frac{\lambda}{2} \|\vec{w}\|_2^2$$

$$= \vec{w}^T X^T X \vec{w} - 2\vec{w}^T X^T \vec{y} + y^T y + \frac{\lambda}{2} \vec{w}^T \vec{w}$$

$$\frac{\partial L(w)}{\partial w} = 2X^T X \vec{w} - 2X^T \vec{y} + \frac{\lambda}{2} \cdot 2\vec{w}$$

Setting $\frac{\partial L(w)}{\partial w} = 0$,

$$\Rightarrow 2X^T X \vec{w}^* + \lambda \vec{w}^* = 2X^T \vec{y}$$

$$\Rightarrow X^T X \vec{w}^* + \frac{1}{2}\lambda I \vec{w}^* = X^T \vec{y}$$

$$\boxed{\Rightarrow \vec{w}^* = \left(X^T X + \frac{1}{2}\lambda I\right)^{-1} X^T \vec{y}}$$

Error metrics – Confusion matrix

- Accuracy is a good performance metric typically when the dataset is class balanced.
- When there is large imbalance (say, 99% of the data belong to class 0 & 1% belong to class 1), accuracy is biased by the majority class.
  - More likely that your model will bias to the majority class!
- Other metrics – from the confusion matrix
  - Sensitivity/Recall – Ability to identify the + class
  - Specificity - Ability to identify the – class
  - Precision – How many of the predictions are correct?
  - F1-Score – *f*(Precision,Recall)

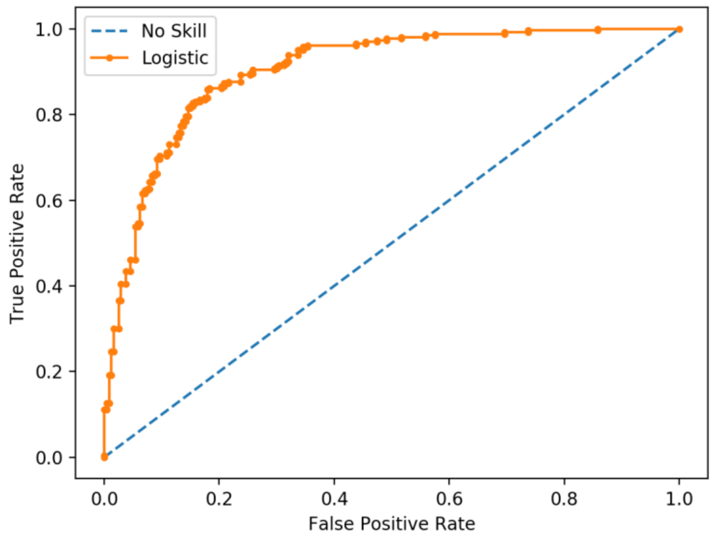
	Predicted +	Predicted -
True +	TP	FN
True -	FP	TN

https://en.wikipedia.org/wiki/Confusion_matrix

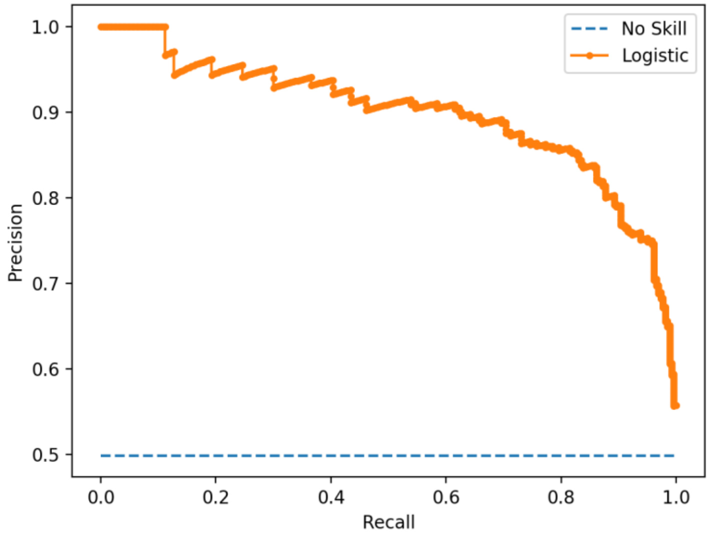
Which metric to use? (in general)

- Event detection (detecting heart beats or an anomaly in a signal) – Sensitivity vs. Specificity (ROC curve)
- Typical classification problems – F1-Score, Precision vs. Recall (Precision-Recall curve)
- Balanced dataset – accuracy

ROC curve

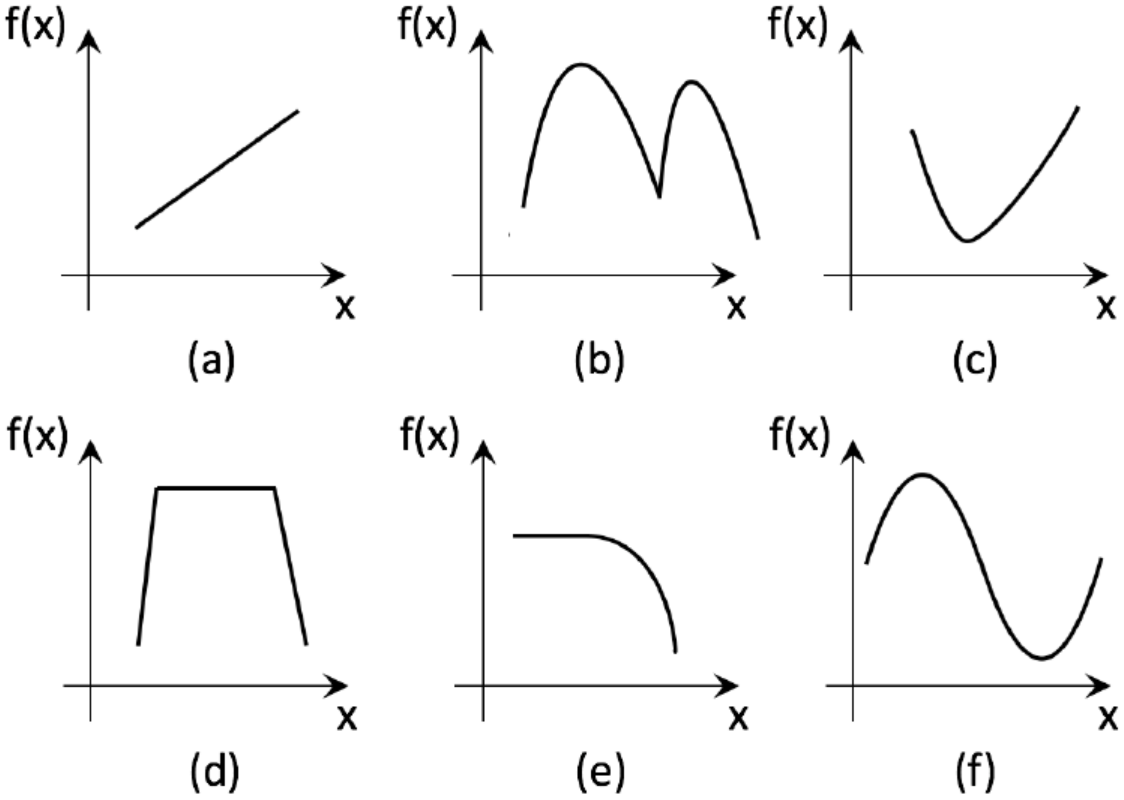


Precision-Recall curve



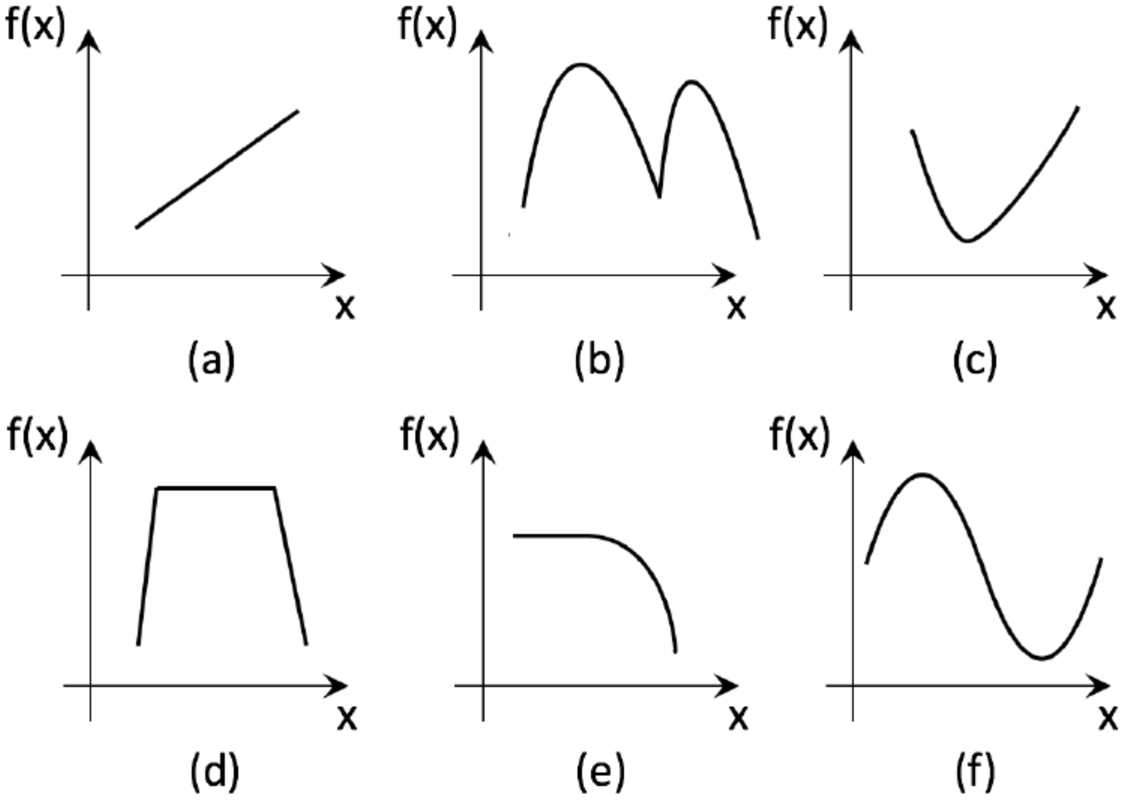
Grab a pen & paper ☺

Identify the convexity for the following six functions (a-f) (Write down whether the function in convex or non-convex).



Grab a pen & paper ☺

Identify the convexity for the following six functions (a-f) (Write down whether the function in convex or non-convex).



Solution:

Convex – a,c

Non-Convex – b,d,e,f

Problem 1

There are 20 samples in a dataset – 15 belong to class A and the remaining 5 belong to class B.

A model f(w) when trained on the dataset correctly classifies 5 samples from class A and 3 samples from class B.

- What is the confusion matrix?
- What is the sensitivity, TPR = TP/(TP+FN) considering B is the + class?

Problem 1

There are 20 samples in a dataset – 15 belong to class A and the remaining 5 belong to class B.

A model f(w) when trained on the dataset correctly classifies 5 samples from class A and 3 samples from class B.

- What is the confusion matrix?
- What is the sensitivity, TPR = TP/(TP+FN) considering B is the + class?

3/(3+2) = 3/5 = 0.6

	Pred A	Pred B
True A	5	10
True B	2	3

Problem 2

A classifier is trying to predict one of the 2 classes {0,1} by returning probabilities for all the 10 samples in the dataset.

We assign the samples to class 1 if the posterior probabilities are greater than a threshold.

X	1	2	3	4	5	6	7	8	9	10
True	1	0	1	1	1	0	1	1	1	1
Post. Prob.	0.8	0.65	0.75	0.98	0.33	0.03	0.44	0.55	0.76	0.43

1. What is the accuracy when the threshold = 0.5?
2. How does the accuracy change when threshold = 0.8 and 0.3?
3. Is accuracy a good measure?
4. Confusion matrix when threshold = 0.5?
5. Find precision & recall when threshold = 0.5. What's the F1 score?

X	1	2	3	4	5	6	7	8	9	10	Accuracy
True	1	0	1	1	1	0	1	1	1	1	
Post. Prob.	0.8	0.65	0.75	0.98	0.33	0.03	0.44	0.55	0.76	0.43	
Pred when th = 0.5	1	1	1	1	0	0	0	1	1	0	6/10
Pred when th = 0.8	1/0	0	0	1	0	0	0	0	0	0	4/10 or 3/10
Pred when th = 0.3	1	1	1	1	1	0	1	1	1	1	9/10

Th = 0.5	Pred 1	Pred 0
True 1	5	3
True 0	1	1

Recall = TP/(TP+FN) = 5/8
Precision = TP/(TP+FP) = 5/6

F1-Score = 2*Precision*Recall/(Precision+Recall) ~= 0.73

$L(w) = \exp(\ y\hat{y})$ where $\hat{y} = \vec{x}^T \vec{w}$

Given: $y = 5$, $x = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$, $w = \begin{bmatrix} e^{10} \\ e^{20} \end{bmatrix}$

Find: $\dfrac{\partial L(w)}{\partial w}$ and $\dfrac{\partial \ln(L(w))}{\partial w}$

What is the relation between them?

By what factor? "Provide a rough estimate".

$$L(w) = \exp\left(y\hat{y}\right) = \exp\left(y\vec{x}^T\vec{w}\right)$$

$$\frac{\partial L(w)}{\partial w} = y\vec{x}^T \exp\left(y\vec{x}^T\vec{w}\right)$$

$$= (5)\begin{pmatrix}1\\4\end{pmatrix} \cdot \exp\left((5)\begin{pmatrix}1 & 4\end{pmatrix}\begin{pmatrix}e^{10}\\e^{20}\end{pmatrix}\right)$$

$$\ln\left(L(w)\right) = y\vec{x}^T\vec{w}$$

$$\frac{\partial \ln\left(L(w)\right)}{\partial w} = y\vec{x}^T = (5)\begin{pmatrix}1\\4\end{pmatrix}$$