CS340 Machine learning
Decision theory

From beliefs to actions

- We have briefly discussed ways to compute p(y|x), where y represents the unknown *state of nature* (eg. does the patient have lung cancer, breast cancer or no cancer), and x are some observable features (eg., symptoms)
- We now discuss: what action a should we take (eg. surgery or no surgery)?
- Define a loss function L(y,a)

y

		None	Lung	Breast
a	Surgery	100	20	10
	No surgery	0	50	50

- Pick the action with minimum expected loss (risk)

$$a^*(x) = \arg\min_a \sum_y p(y|x) L(y,a)$$

Loss/ utility functions, policies

- In statistics, we use loss functions L. In economics, we use utility functions U. Clearly U=-L.
- The principle of maximum expected utility says the optimal (rational) action is

$$a^*(x) = \arg\max_a \sum_y p(y|x) U(y,a)$$

- A decision procedure δ(x) or policy π(x) is a mapping from X to A, which specifies which action to perform for every possible observed feature vector x.

Bayes decision rule

- The conditional risk (expected loss conditioned on x) is

$$R(a|x) = \sum_y p(y|x) L(y,a)$$

- The optimal strategy (Bayes decision rule) is

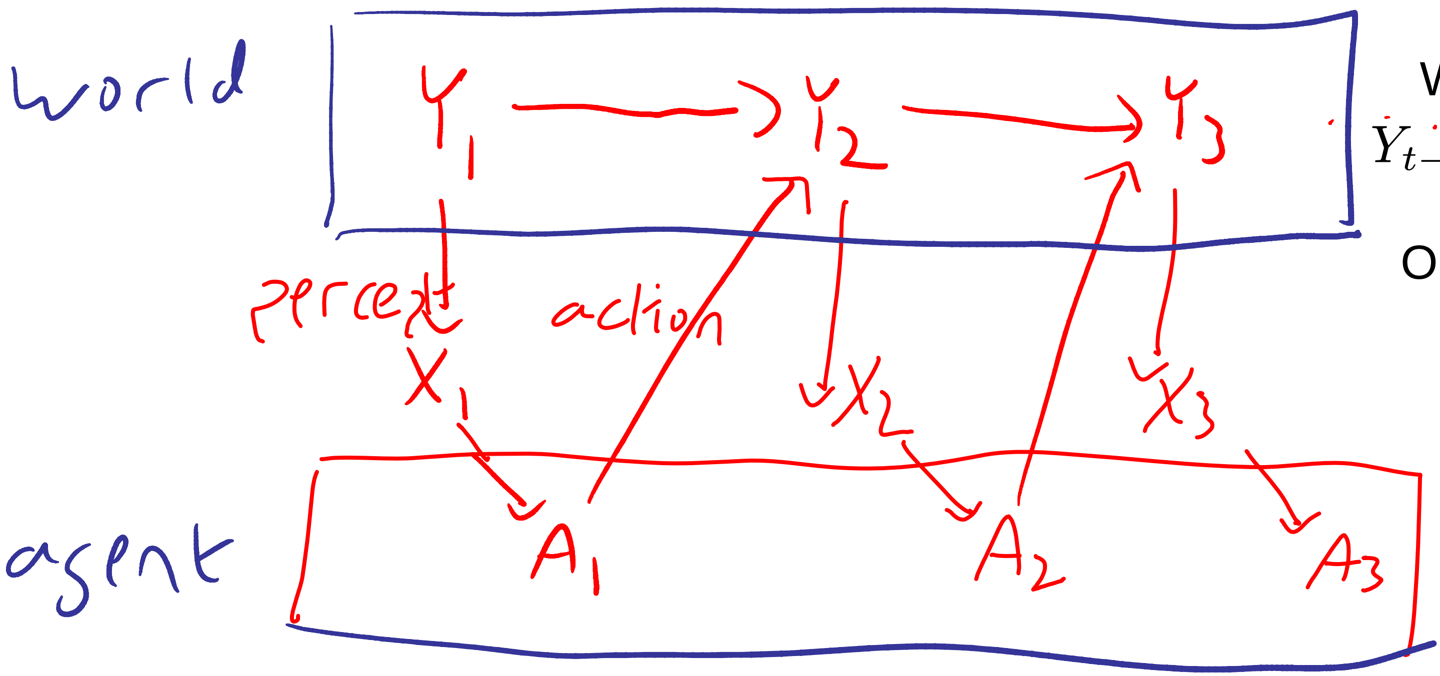
$$\pi(x) = \arg\min_a R(a|x)$$

- The Bayes risk is the expected performance of the optimal strategy

$$r = \int dx \sum_y L(y, \pi(x)) p(x,y)$$

Sequential decision problems

- In general we need to reason about the consequences of our actions.
- This is beyond the scope of this class (see e.g. CS422). We focus on one-shot decision problems.



World model

$Y_{t-1} \times A_{t-1} \to Y_t$

Observation model

$Y_t \to X_t$

Policy

$X_t \to A_{t+1}$

Classification problems

- In classification problems, the action space A is usually taken to be the same as the label space Y.
- We interpret the action a as our best guess about the true label y. The loss matrix defines the penalties for getting the answer wrong.

y

$\hat{y}$

	None	Lung	Breast
None	0	100	100
Lung	50	0	10
Breast	50	10	0

Binary classification problems

- Let Y=1 be ‘positive’ (eg cancer present) and Y=2 be ‘negative’ (eg cancer absent).
- The loss/ cost matrix has 4 numbers:

state y

action $\hat{y}$

	1	2
1	True positive λ_{11}	False positive λ_{12}
2	False negative λ_{21}	True negative λ_{22}

Optimal strategy for binary classification

- We should pick class/ label/ action 1 if

$$\begin{array}{rcl} R(\alpha_2|\mathbf{x}) & > & R(\alpha_1|\mathbf{x}) \\ \lambda_{21} p(Y=1|\mathbf{x}) + \lambda_{22} p(Y=2|\mathbf{x}) & > & \lambda_{11} p(Y=1|\mathbf{x}) + \lambda_{12} p(Y=2|\mathbf{x}) \\ (\lambda_{21}-\lambda_{11}) p(Y=1|\mathbf{x}) & > & (\lambda_{12}-\lambda_{22}) p(Y=2|\mathbf{x}) \\ \frac{p(Y=1|\mathbf{x})}{p(Y=2|\mathbf{x})} & > & \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \end{array}$$

  where we have assumed λ_{21} (FN) >λ_{11} (TP)

- As we vary our loss function, we simply change the optimal threshold θ on the decision rule

$$\pi(x) = 1 \text{ iff } \frac{p(Y=1|x)}{p(Y=2|x)} > \theta$$

0-1 loss

- If the loss function penalizes misclassification errors equally

state y

action $\hat{y}$

	1	2
1	0 λ_{11}	1 λ_{12}
2	1 λ_{21}	0 λ_{22}

- then we should pick the most probable class

$$\pi(x) = 1 \iff \frac{p(Y=1|\mathbf{x})}{p(Y=2|\mathbf{x})} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} = \frac{1-0}{1-0} = 1$$

- In general, for 0-1 loss and multiple classes,

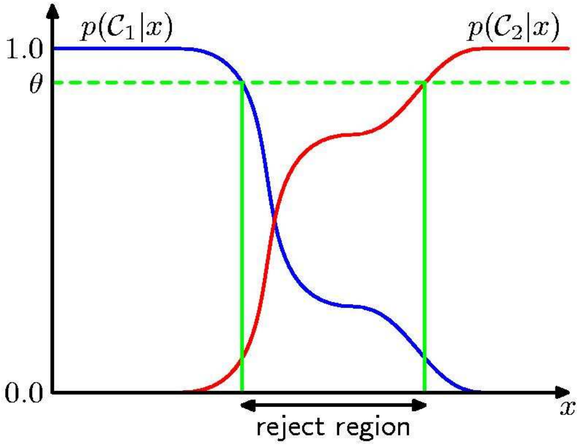
$$\pi(x) = \arg\max_j p(Y=j|x)$$

Reject option

- Suppose we can choose between incurring loss λ_s if we make a misclassification (label substitution) error and loss λ_r if we declare the action “don’t know”

$$\lambda(\alpha_i|Y=j) = \begin{cases} 0 & \text{if } i=j \text{ and } i,j \in \{1,\ldots,C\} \\ \lambda_r & \text{if } i = C+1 \\ \lambda_s & \text{otherwise} \end{cases}$$

- In HW2, you will show that the optimal action is to pick “don’t know” if the most probable class is below a threshold 1-λ_r/λ_s



Bishop 1.26

Discriminant functions

- The optimal strategy π(x) partitions X into decision regions R_i, defined by discriminant functions $g_i(x)$

$$\pi(x) = \arg\max_i g_i(x)$$

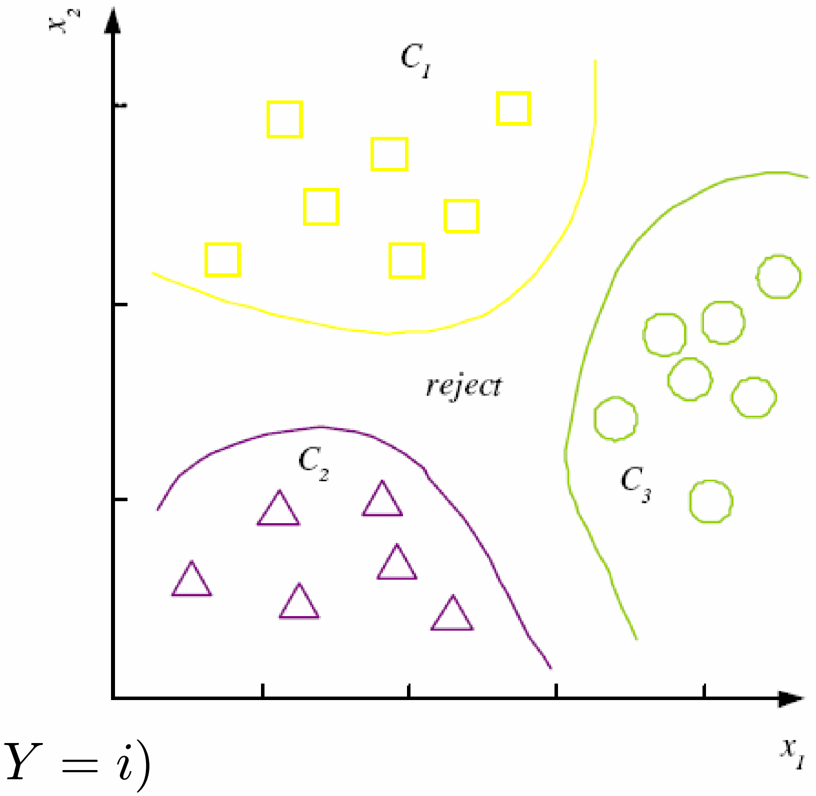
$$R_i = \{x : g_i(x) = \max_k g_k(x)\}$$

In general

$$g_i(x) = -R(a = i|x)$$

But for 0-1 loss we have

$$\begin{aligned} g_i(x) &= p(Y = i|x) \\ &= \log p(Y = i|x) \\ &= \log p(x|Y = i) + \log p(Y = i) \end{aligned}$$

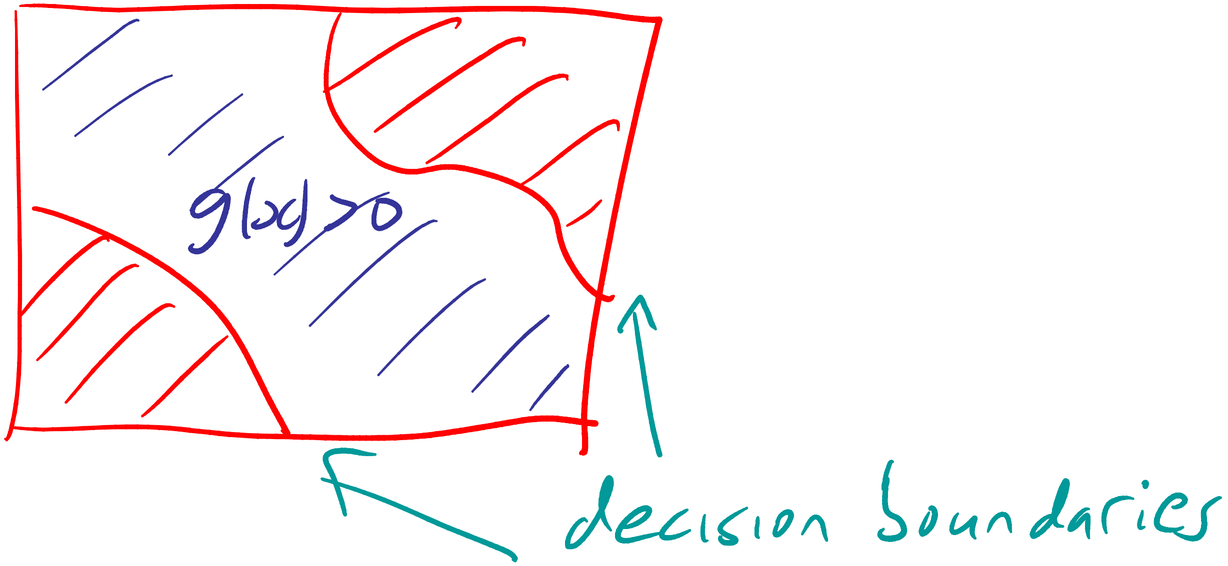


Class prior merely shifts decision boundary by a constant

Binary discriminant functions

- In the 2 class case, we define the discriminant in terms of the log-odds ratio

$$
\begin{aligned}
g(x) &= g_1(x) - g_2(x) \\
&= \log p(Y=1|x) - \log p(Y=2|x) \\
&= \log \frac{p(Y=1|x)}{p(Y=2|x)}
\end{aligned}
$$



Do we need probabilistic classifiers?

- One popular approach to ML is to learn the classification function $\pi(x) = f(x,w)$ directly, bypassing the need to estimate $p(y|x)$

$$w^* = \arg\min_w \sum_n L(y_n, f(x_n, w))$$

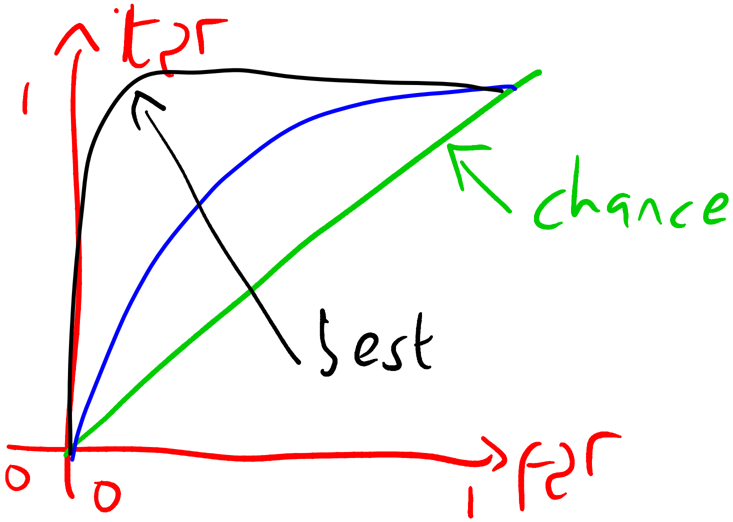
- However, having access to $p(y|x)$ is useful because
  - Modular – no need to relearn if change L
  - Can use reject option
  - Can combine different $p(y|x)$'s
  - Can compensate for different class priors $p(y)$
  - Scientific discovery (inference) often involves examining typical samples from $p(y|x)$, rather than decision making.

ROC curves

- The optimal threshold for a binary detection problem depends on the loss function

$$\pi(x) = 1 \quad \Longleftrightarrow \quad \frac{p(Y=1|\mathbf{x})}{p(Y=2|\mathbf{x})} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}}$$

- Low threshold will give rise to many false positives (Y=1) and high threshold to many false negatives.
- A receive operating characteristic (ROC) curves plots the true positive rate vs false positive rate as we vary θ



Definitions

- Declare x_n to be a positive if $p(y=1|x_n)>\theta$, otherwise declare it to be negative (y=2)

$$\hat{y}_n = 1 \iff p(y=1|x_n) > \theta$$

- Define the number of true positives as

$$TP = \sum_n I(\hat{y}_n = 1 \wedge y_n = 1)$$

- Similarly for FP, TN, FN – all functions of θ

$$\begin{array}{c|c|c|c|c}
 & & & y & \\
 & & 1 & 2 & \\
\hline
 & 1 & TP & FP & \hat{P} \\
\hline
\hat{y} & 2 & FN & TN & \hat{N} \\
\hline
 & & P & N &
\end{array}$$

$$\begin{aligned}\hat{P} &= TP + FP \\ \hat{N} &= FN + TN\end{aligned}$$

$$P = TP + FN, \;\; N = FP + TN$$

Performance measures

		y		
		1	2	
$\hat{y}$	1	TP	FP	$\hat{P}$
	2	FN	TN	$\hat{N}$
		P	N	

precision = positive predictive value (PPV) = TP / P-hat

Sensitivity = recall =
True pos rate = hit rate
= TP / P = 1-FNR

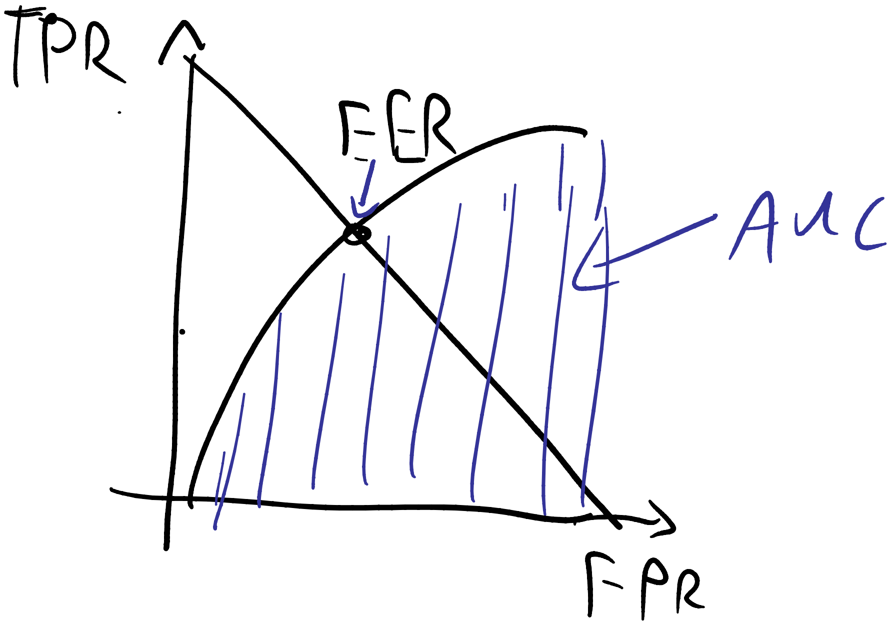
False pos rate = false acceptance =
= type I error rate = FP / N = 1-spec

False neg rate = false rejection =
type II error rate = FN / P = 1-TPR

Specificity = TN / N = 1-FPR

Performance measures

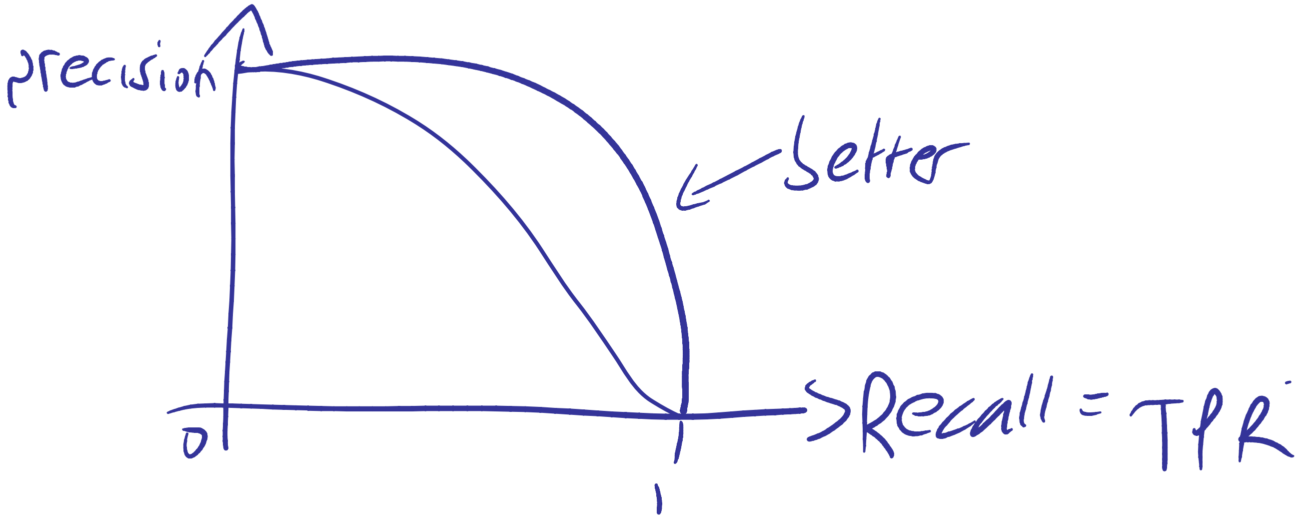
- EER- Equal error rate/ cross over error rate (false pos rate = false neg rate), smaller is better
- AUC - Area under curve, larger is better
- Accuracy = (TP+TN)/(P+N)



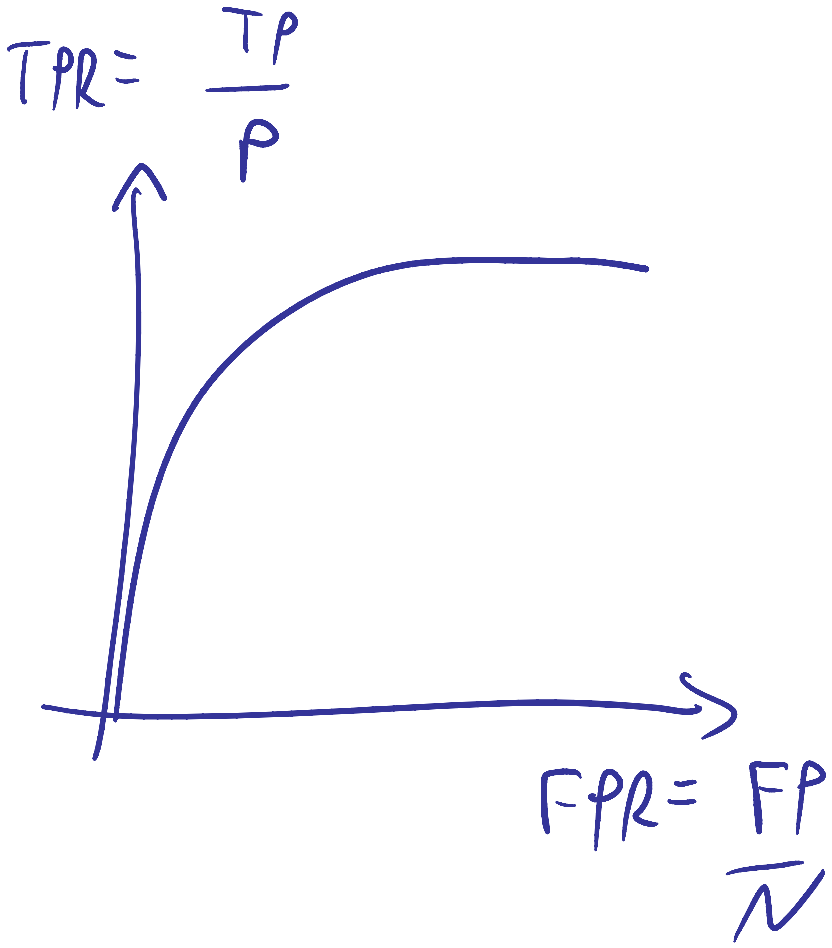
Precision-recall curves

- Useful when notion of “negative” (and hence FPR) is not defined
- Used to evaluate retrieval engines
- Recall = of those that exist, how many did you find?
- Precision = of those that you found, how many correct?
- F-score is harmonic mean

$$F = \frac{2}{1/P + 1/R} = \frac{2PR}{R + P}$$

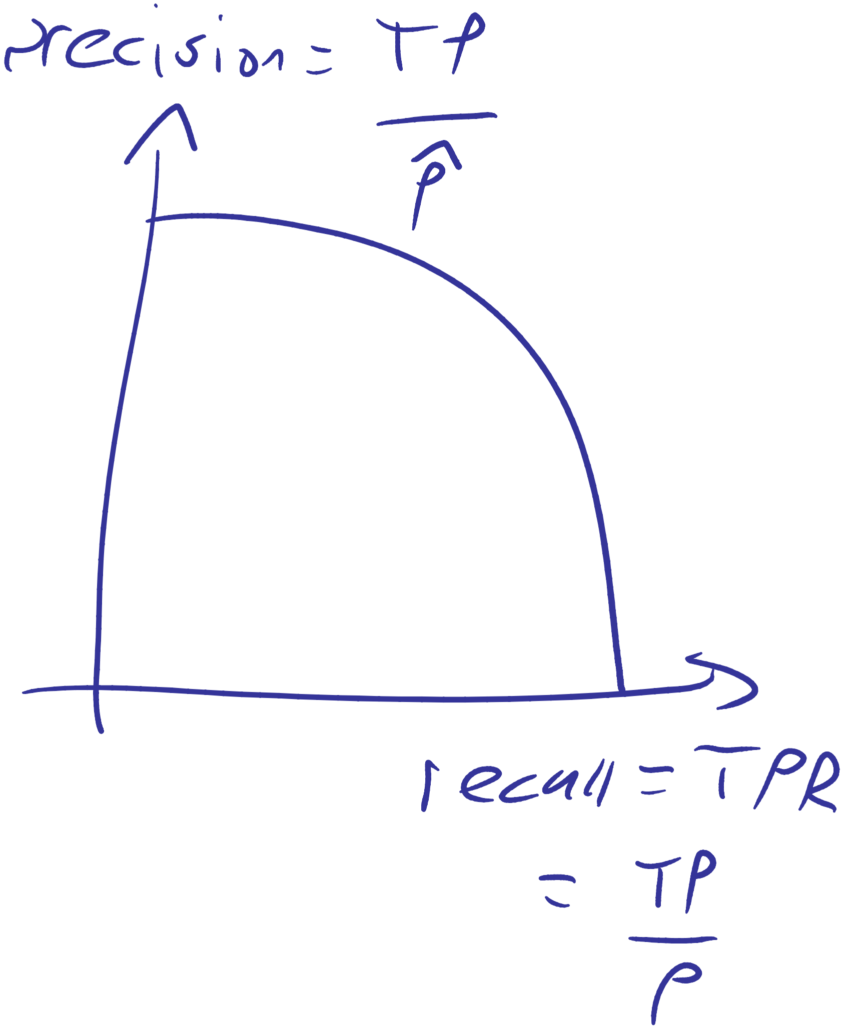


ROC vs PR curves



$TPR = \frac{TP}{P}$

$FPR = \frac{FP}{N}$



$precision = \frac{TP}{\hat{P}}$

$recall = TPR$

$= \frac{TP}{P}$

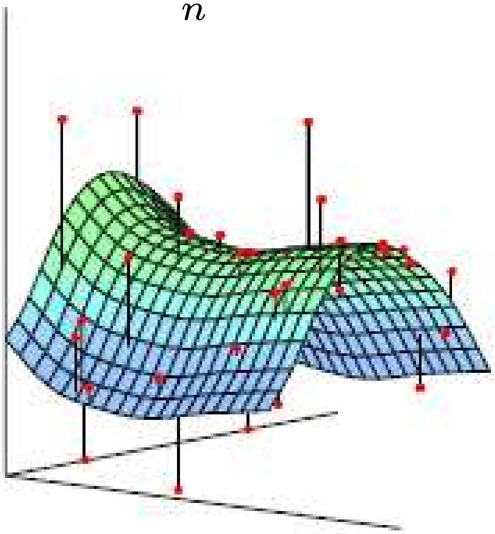
Loss functions for regression

- Regression means predicting $y \in \mathbb{R}$; classification means predicting a discrete output $y \in \{1, 2, \ldots, C\}$
- The most common loss is squared error

$$L(y, f(x|\theta)) = (y - f(x|\theta))^2$$

- The residual sum of squares is

$$RSS(\theta) = \sum_n (y_n - f(x_n|\theta))^2$$



HTF 2.10

Minimizing squared error

- The expected loss is

$$EL = \int\int (y - f(x))^2 p(x,y) dx dy$$

- Let us discretize x and optimize this wrt f_x

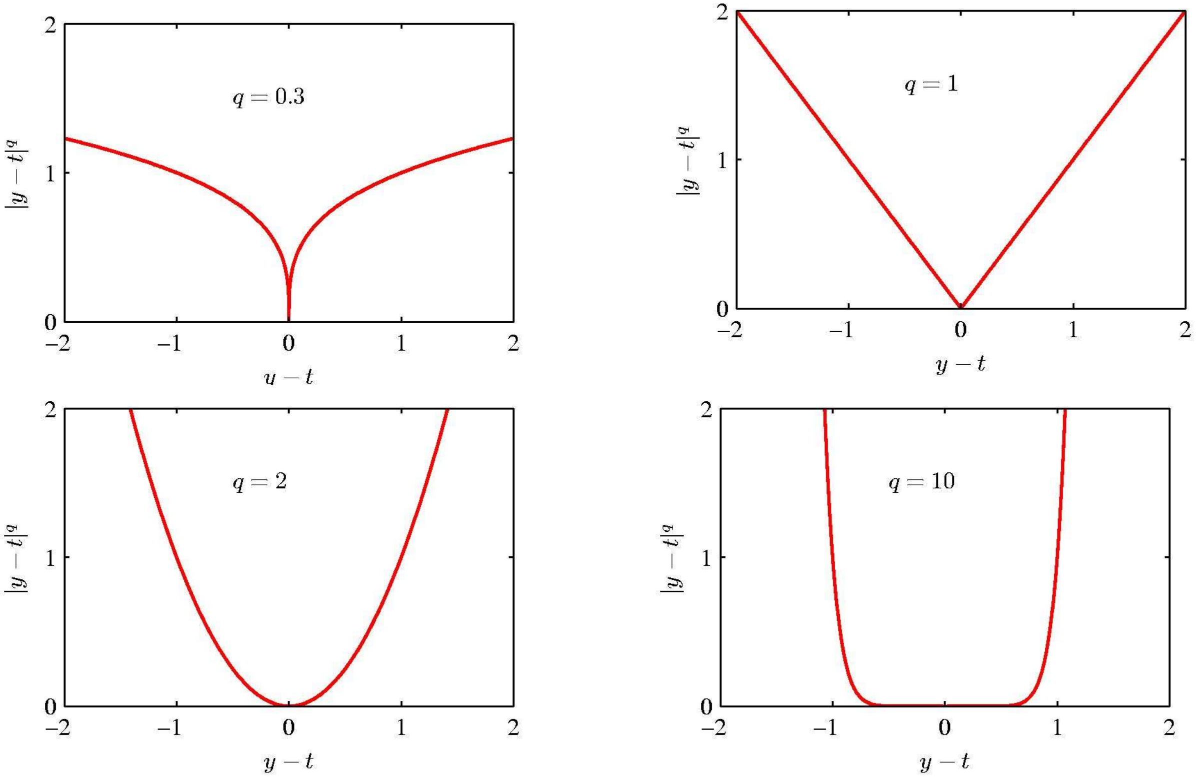
$$\begin{aligned}
\frac{\partial}{\partial f_x} E[L] &= \frac{\partial}{\partial f_x} \int dy \sum_x (y - f_x)^2 p(x,y) \\
&= \int dy \; 2(y - f_x) p(x,y) \\
&= 0 \Rightarrow \\
f_x p(x) &= \int dy \; y \; p(x,y) \\
f_x &= E[y|x]
\end{aligned}$$

- **Hence to minimize squared error, we should compute the posterior mean E[y|x]**

Robust loss functions

- Square error (L2) is sensitive to outliers
- It is common to use L1 instead.
- In general, Lp loss is defined as

$$L_p(y, \hat{y}) = |y - \hat{y}|^p$$



Minimizing robust loss functions

- For L2 loss, mean p(y|x)
- For L1 loss, median p(y|x)
- For L0 loss, mode p(y|x)