EE5907/EE5027 Week 6: Bayesian Statistics

BT Thomas Yeo

ECE, CSC, CIRC, N.1, HMS

Last Week Recap

- Non-parametric approaches do not mean no parameters, but instead parameters grow with more data
  - Do not assume data is from specific distributions, such as Gaussian
  - Less assumptions imply non-parametric approaches need more data
- Two problems: density estimation and classification
- Two approaches
  - Parzen's window: Count number of neighbors inside fixed window size
  - KNN: Expand window until K neighbors are captured

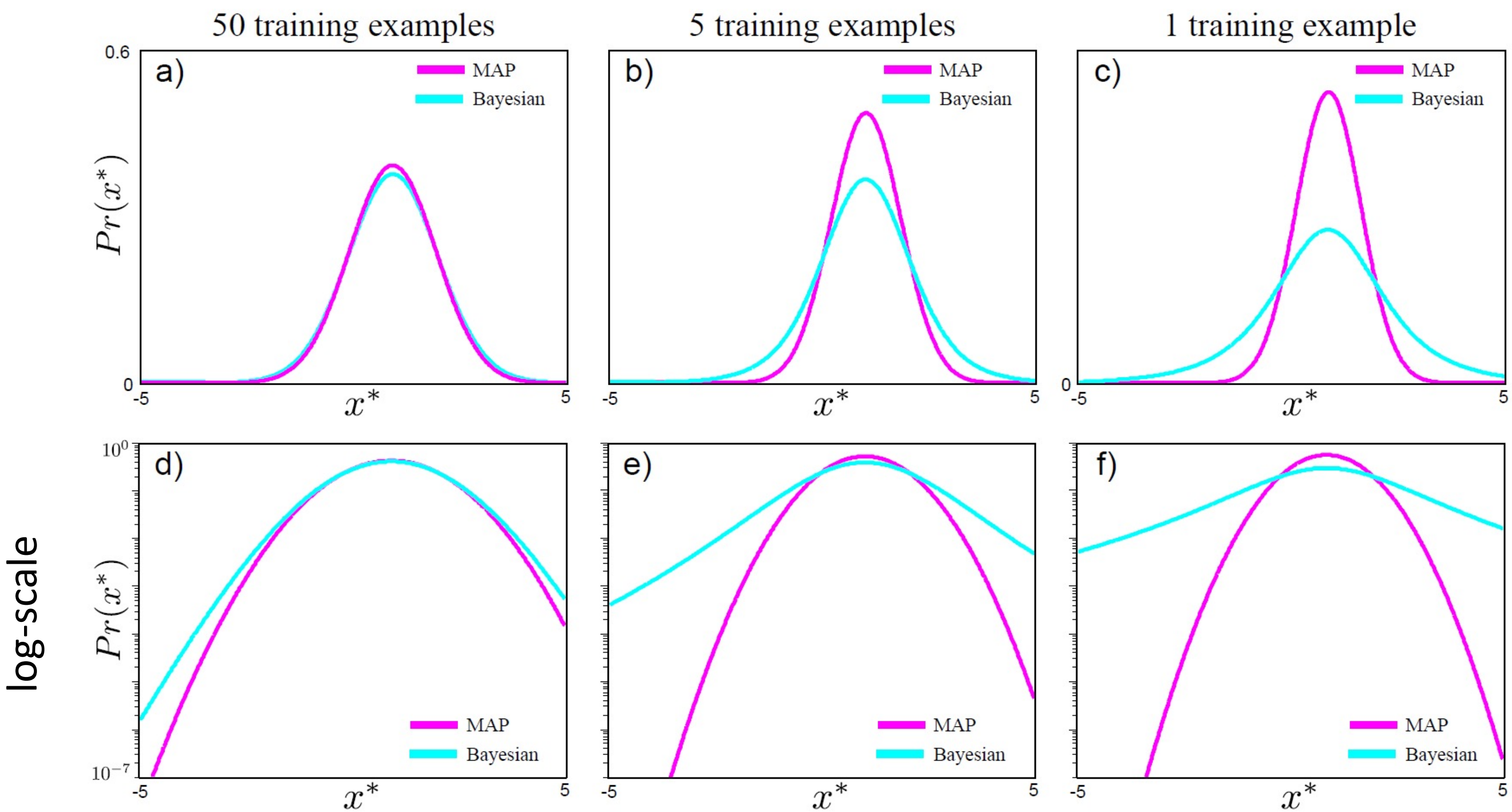
This week

- Problems With MAP estimation
- Bayesian model selection
- Bayesian decision theory

Problems With MAP Estimation

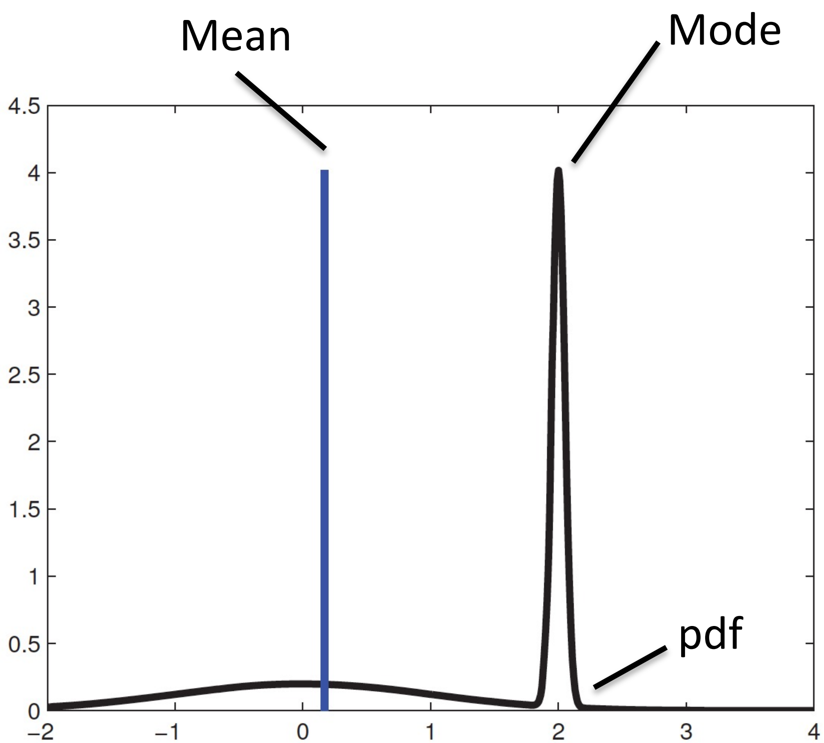
MAP Problems

- Plug-in approximation can overfit (black swan paradox)

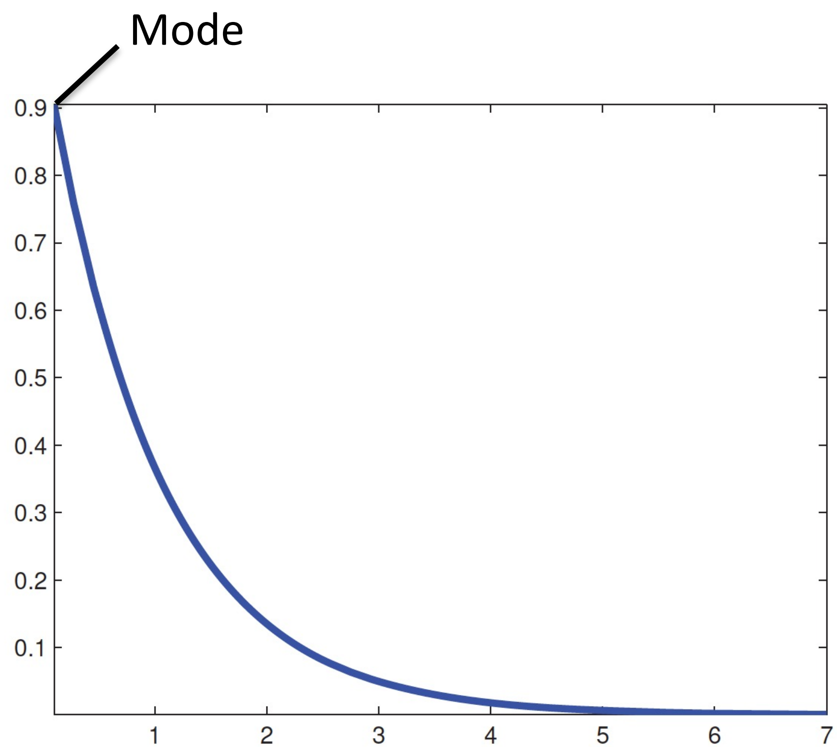


MAP Problems

- Mode can be atypical unlike mean and median which account for "volume" of pdf



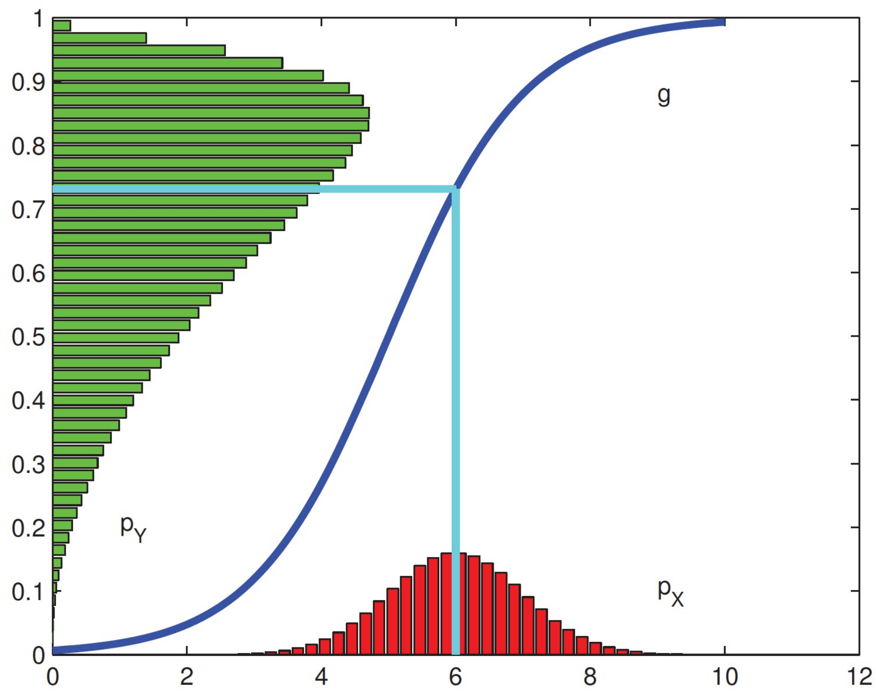
Bimodal Distribution (black)



Skewed Distribution (blue)

MAP Problems

- MAP (unlike ML and posterior predictive estimation) is sensitive to parameterization
- $y = f(x) \implies p_Y(y) = p_X(x)|\frac{dx}{dy}|$ (for f monotonic, invertible)
  - $y_{MODE} \neq f(x_{MODE})$



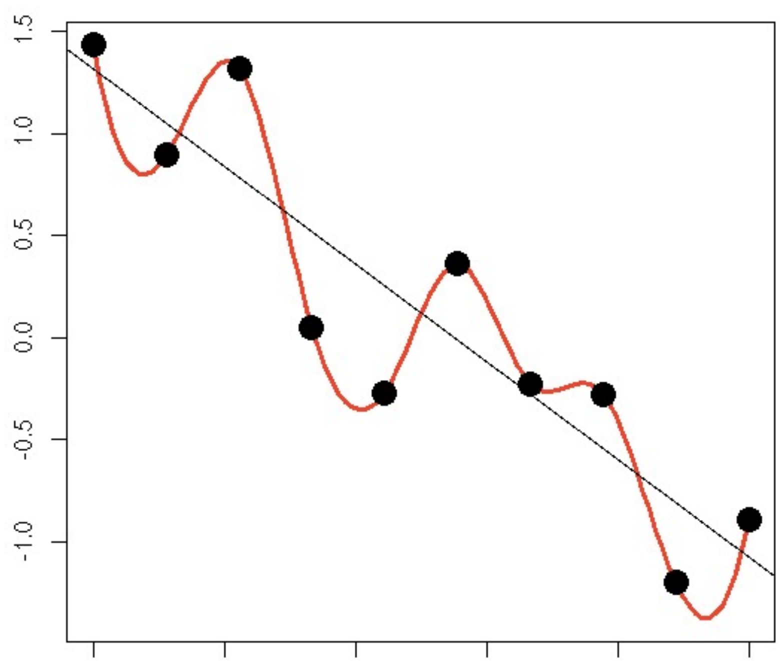
$$x \sim \mathcal{N}(6, 1)$$

$$y = \frac{1}{1 + e^{-x+5}}$$

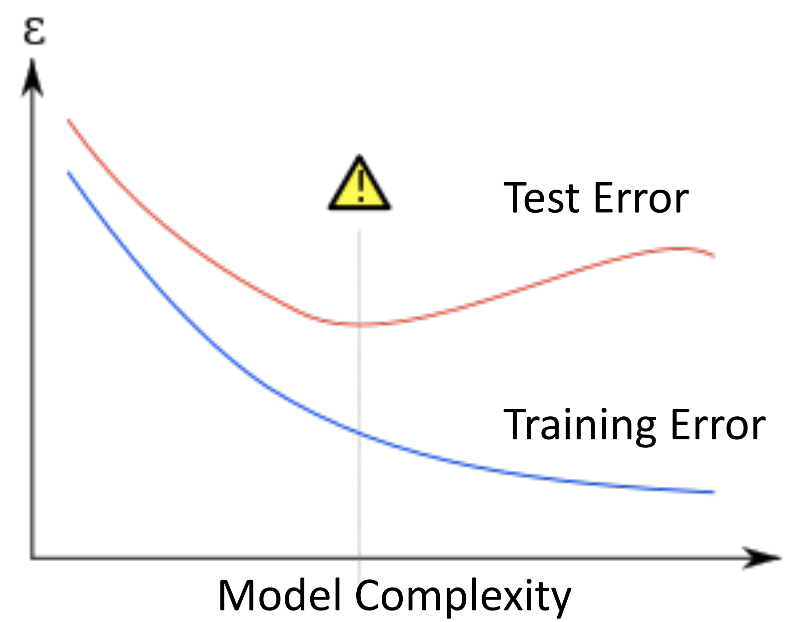
Bayesian Model Selection

Overfitting

- Very complex models can explain training data well, but test data poorly. If models too simple, may explain both training and test data poorly
  - Regularization (explicit or implicit)
  - Cross Validation
- Bayesian model selection tradeoffs complexity with training error



Can exactly fit N+1 data points with N-degree polynomial



Images from google images

Bayesian Occam's Razor

- Bayesian model selection:

$$\hat{m} = \operatorname*{argmax}_m p(m|D) = \operatorname*{argmax}_m p(D|m)p(m)$$
$$= \operatorname*{argmax}_m p(D|m) \quad \text{assuming } p(m) \propto 1$$

  - $p(D|m)$ is called marginal likelihood

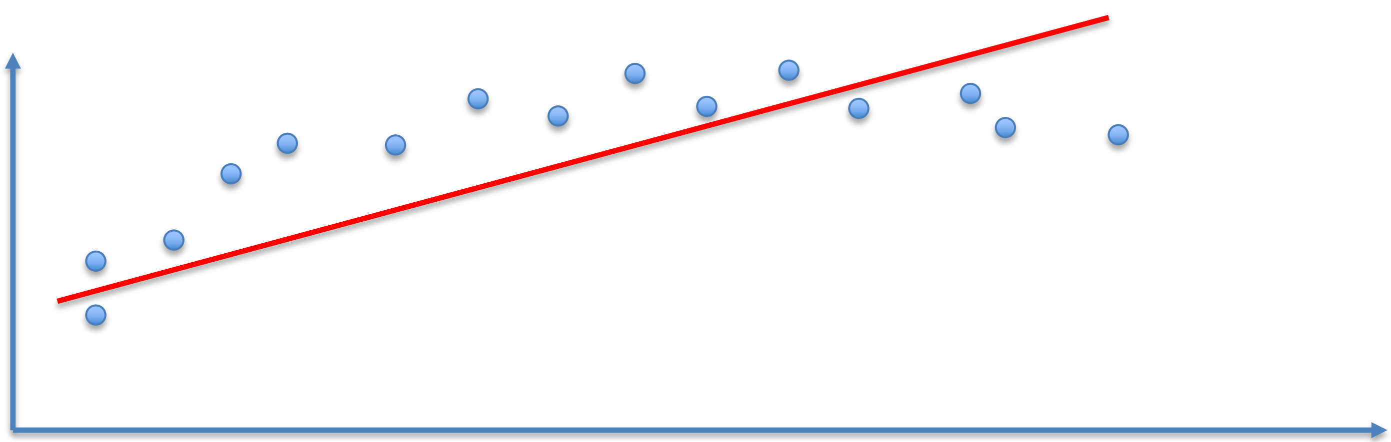
- Recall: suppose we observe D Gaussian samples and want to estimate $\theta = (\mu, \sigma^2)$

$$\theta_{MAP} = \operatorname*{argmax}_\theta p(\theta|D) = \operatorname*{argmax}_\theta \frac{p(\theta)p(D|\theta)}{p(D)}$$
$$= \text{we drop } p(D) \text{ because does not depend on } \theta$$

  - Above implicitly depends on modeling assumptions (e.g., Normal inverse Gamma prior with specific hyperparameters), $\frac{p(\theta)p(D|\theta)}{p(D)}$ can be more explicitly written as $\frac{p(\theta|m)p(D|\theta,m)}{p(D|m)}$ Therefore, the "evidence" term $p(D)$ we throw away for MAP estimation is actually "marginal likelihood" $p(D|m)$

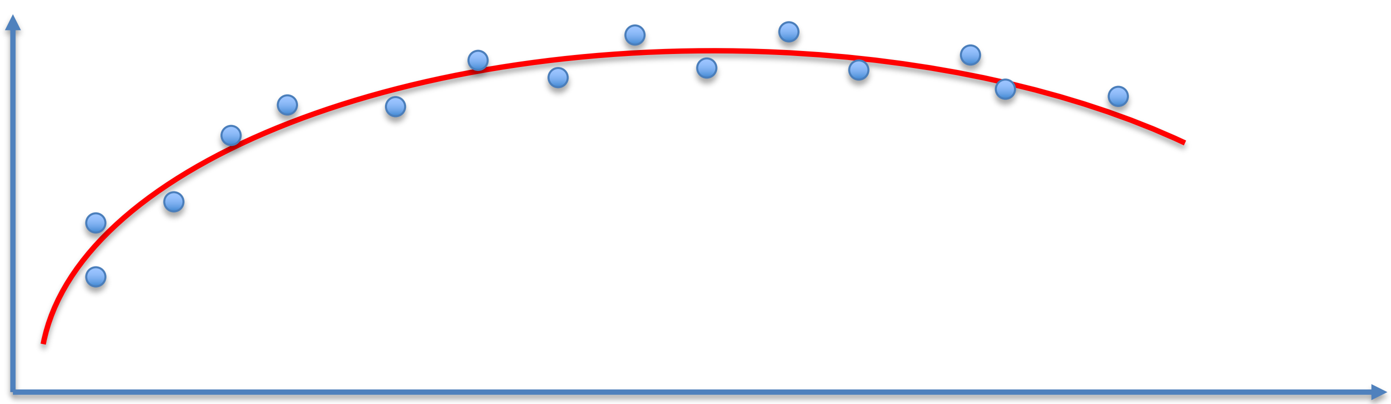
Why does it work?

- Complex models can explain many things $\implies p(D'|m)$ is non-zero for many different and complex D'.
- However, $\sum_{D'} p(D'|m) = 1$ and many $p(D'|m)$ are non-zero, which means $p(D'|m)$ cannot be very big either
- Example: Dots come from quadratic curve.
  - M_1 (linear curves): cannot fit dots well, so $p(\text{dots}|M_1)$ is small



Why does it work?

- Complex models can explain many things $\implies p(D'|m)$ is non-zero for many different and complex D'.
- However, $\sum_{D'} p(D'|m) = 1$ and many $p(D'|m)$ are non-zero, which means $p(D'|m)$ cannot be very big either
- Example: Dots come from quadratic curve.
  - M_1 (linear curves): cannot fit dots well, so $p(\text{dots}|M_1)$ is small
  - M_3 (linear, quadratic, cubic curves): can fit dots well, but "waste" non-zero probability on cubic curves, so $p(\text{dots}|M_3)$ cannot be too big
  - M_2 (linear, quadratic curves): can fit dots well, non-zero probability only for linear, quadratic curves, so $p(\text{dots}|M_2)$ highest ✓



Computing Marginal Likelihood / Evidence

- Previous class: $p(\theta|D,m) \propto p(\theta|m)p(D|\theta,m) \implies$ ignore denominator $p(D|m)$ as "constant" because only consider one model

- Now need to compare models, so $p(D|m)$ is quantity of interest!

- For conjugate distributions, posterior $p(\theta|D)$ easy to compute, so can just solve for $p(D)$ using Bayes' rule: $p(D) = \frac{p(D|\theta)p(\theta)}{p(\theta|D)}$

- Example: $p(D|\theta) = \mathrm{Bin}(N_0, N_1|\theta), p(\theta) = \mathrm{Beta}(a,b), p(\theta|D) = \mathrm{Beta}(a + N_1, b + N_0)$, then

prior likelihood

evidence $$p(D) = \frac{\left(\frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}\right)\left(\binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}\right)}{\frac{1}{B(a+N_1,b+N_0)}\theta^{a+N_1-1}(1-\theta)^{b+N_0-1}}$$

posterior

Computing Marginal Likelihood / Evidence

- Previous class: $p(\theta|D,m) \propto p(\theta|m)p(D|\theta,m) \implies$ ignore denominator $p(D|m)$ as "constant" because only consider one model

- Now need to compare models, so $p(D|m)$ is quantity of interest!

- For conjugate distributions, posterior $p(\theta|D)$ easy to compute, so can just solve for $p(D)$ using Bayes' rule: $p(D) = \frac{p(D|\theta)p(\theta)}{p(\theta|D)}$

- Example: $p(D|\theta) = \text{Bin}(N_0, N_1|\theta), p(\theta) = \text{Beta}(a,b), p(\theta|D) = \text{Beta}(a + N_1, b + N_0)$, then

$$
\begin{aligned}
p(D) &= \frac{\left(\frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}\right)\left(\binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}\right)}{\frac{1}{B(a+N_1,b+N_0)}\theta^{a+N_1-1}(1-\theta)^{b+N_0-1}} \\
&= \binom{N}{N_1}\frac{B(a+N_1,b+N_0)}{B(a,b)}
\end{aligned}
$$

Does not depend on Θ → "Beta-binomial compound distribution" parameterized by a and b

- Bayesian Information Criteria (BIC): $\log p(D) \approx \log p(D|\theta_{ML}) - (\text{dof}(\theta)/2)\log N$
BIC cost: $-2\log p(D) \approx -2\log p(D|\theta_{ML}) + \text{dof}(\theta)\log N$

- θ_{MAP} may work better than θ_{ML}

Bayes Factor

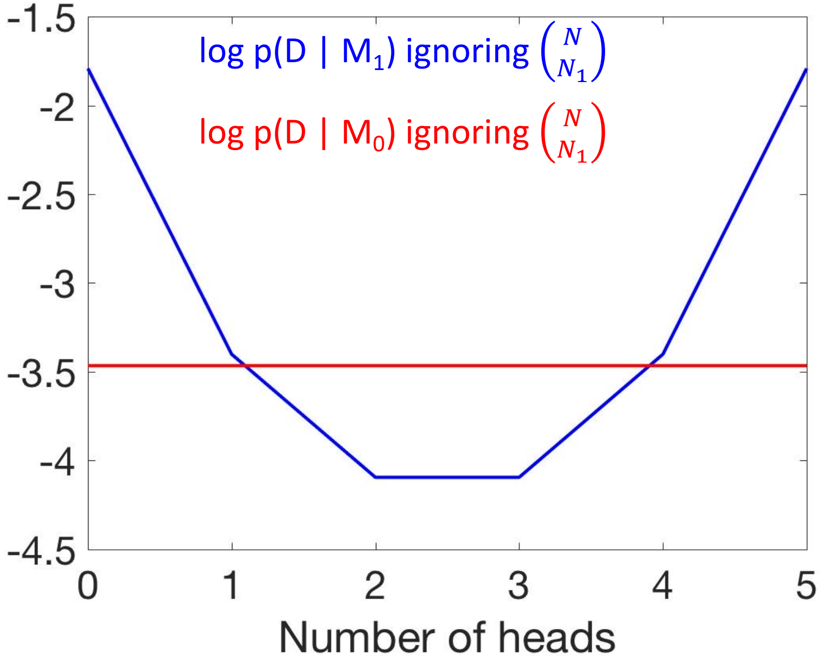
- Suppose two models M_0 and M_1. Bayes factor $BF_{1,0} = \frac{p(D|M_1)}{p(D|M_0)}$
  - Bayes alternative to hypothesis testing in frequentist statistics

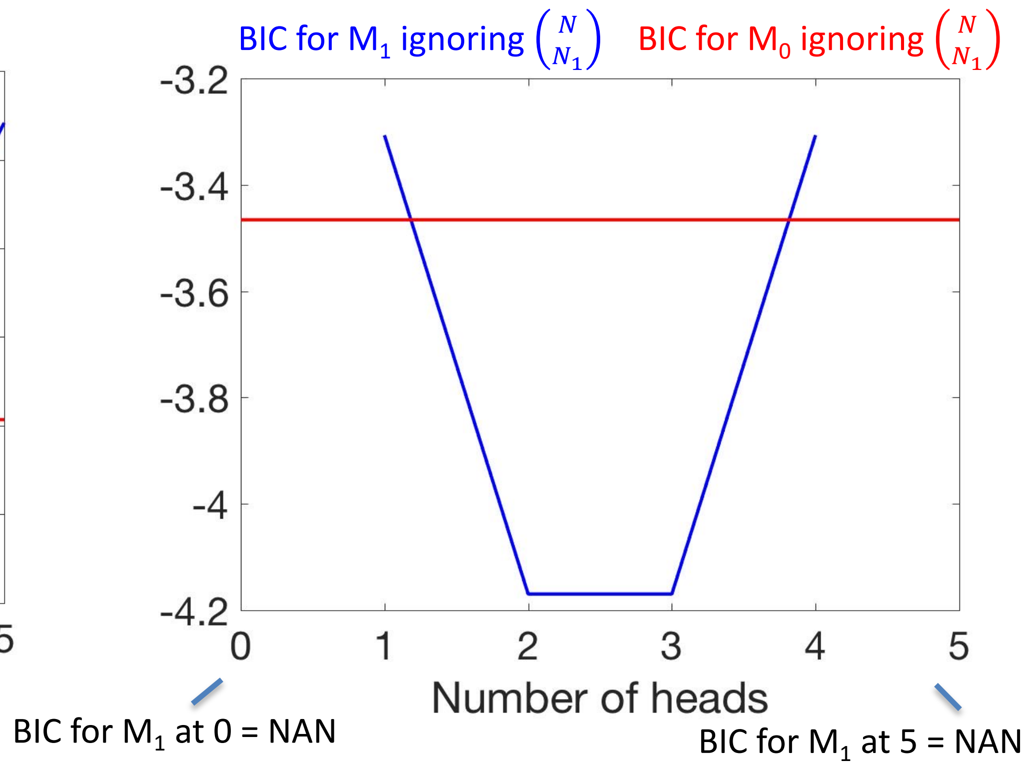
Bayes factor $BF(1,0)$	Interpretation
$BF < \frac{1}{100}$	Decisive evidence for M_0
$BF < \frac{1}{10}$	Strong evidence for M_0
$\frac{1}{10} < BF < \frac{1}{3}$	Moderate evidence for M_0
$\frac{1}{3} < BF < 1$	Weak evidence for M_0
$1 < BF < 3$	Weak evidence for M_1
$3 < BF < 10$	Moderate evidence for M_1
$BF > 10$	Strong evidence for M_1
$BF > 100$	Decisive evidence for M_1

- Assuming $p(M_1) = p(M_0) = 0.5$, then $p(M_0|D) = \frac{1}{BF_{1,0}+1}$

Example: Is a coin fair?

- M_0 : fair coin, $M_1 : \theta \sim \text{Beta}(1,1)$, observe N_1 heads in N tosses
  - $p(D|M_0) = \binom{N}{N_1} \frac{1}{2^N}$
  - $p(D|M_1) = \binom{N}{N_1} \frac{B(1+N_1, 1+N_0)}{B(1,1)}$
- Ignore $\binom{N}{N_1}$ which appears in both models (so $p(D|M_0)$ is a constant):





Bayesian Decision Theory

Posterior Expected Loss

- Posterior probability is nice but need to convert into real world action
- Game against nature
  - Nature picks $y \in \mathcal{Y}$ and then generates observation $x \in \mathcal{X}$
  - We then choose action $a \in \mathcal{A}$, resulting in some loss $L(y, a)$ based on compatibility between y and a. Example: $L(y, a) = (y - a)^2$
  - Pick action by minimizing posterior expected loss:

$$\delta(x) = \operatorname*{argmin}_{a \in \mathcal{A}} E[L(y, a)] \stackrel{\triangle}{=} \operatorname*{argmin}_{a \in \mathcal{A}} \sum_{y} L(y, a) p(y|x)$$

  - Bayes' estimator or decision rule
- In economics, instead of loss L, we have utility $U(y, a)$, so $\delta(x) = \operatorname{argmax}_{a \in \mathcal{A}} E[U(y, a)]$

MAP Estimate Minimizes 0-1 Loss

- $0 - 1$ loss: $L(y, a) = \mathbb{I}(y \neq a) = \begin{cases} 0 & \text{if } a = y \\ 1 & \text{if } a \neq y \end{cases}$
  - In classification, y is true label, $a = \hat{y}$ is estimated label
- For two classes, can visualize:

	$\hat{y} = 1$	$\hat{y} = 0$
$y = 1$	0	1
$y = 0$	1	0

- Posterior expected loss:

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

$$= 1 - p(y = a|x)$$

- Minimizing $1 - p(y = a|x)$ equivalent to maximize $p(y = a|x)$, i.e., MAP estimate

Posterior Mean Minimizes l_2 Loss

- Quadratic (or l_2) loss: $L(y, a) = (y - a)^2$
- Posterior expected loss:

$$\rho(a|x) = \sum_y L(y, a)p(y|x) = E[(y - a)^2|x]$$
$$= E(y^2|x) - 2aE(y|x) + a^2$$

- Differentiating with respect to a:

$$\frac{\partial}{\partial a}\rho(a|x) = -2E(y|x) + 2a = 0$$
$$\implies a = E(y|x)$$

  – Minimum Mean Squared Error (MMSE) estimate corresponds to posterior mean

Posterior Median Minimizes l_1 Loss

- Absolute (or l_1) loss: L(y, a) = |y – a|
  - Posterior median minimizes l_1 loss
  - See ungraded homework assignment

False Positive – False Negative Tradeoff

- General loss matrix for binary classification:

	$\hat{y}=1$	$\hat{y}=0$
$y=1$	0	L_{FN}
$y=0$	L_{FP}	0

  - L_{FN} = false negative cost, L_{FP} = false positive cost

- Posterior expected loss are

$$L(\hat{y}=0|x) = L_{FN}p(y=1|x)$$
$$L(\hat{y}=1|x) = L_{FP}p(y=0|x)$$

- Therefore, should pick $\hat{y}=1$ iff

$$L(\hat{y}=0|x) > L(\hat{y}=1|x)$$
$$\frac{p(y=1|x)}{p(y=0|x)} > \frac{L_{FP}}{L_{FN}}$$

Different Metrics

- From previous slide, threshold $f(x) = \frac{p(y=1|x)}{p(y=0|x)}$ at $\tau = \frac{L_{FP}}{L_{FN}}$ to make classification decision
- When comparing two algorithms, we often do not know (or do not want) to define L_{FP} and L_{FN} because for the same dataset (e.g., facebook photos), loss might be application specific
- Instead of thresholding $f(x)$ at $\tau = \frac{L_{FP}}{L_{FN}}$, we threshold at different τ, and compute for each τ
- TP (true positives)
  - N_{TP} = # data points in test set whose true label = 1 & classified correctly as 1
  - N_1 = # data points in test set whose true label = 1
  - TPR (true positive rate or sensitivity or recall) = N_{TP} / N_1

	ŷ = 0	ŷ = 1	
y = 0	N_{TN}	N_{FP}	$N_0 = N_{TN} + N_{FP}$
y = 1	N_{FN}	N_{TP}	$N_1 = N_{FN} + N_{TP}$

Different Metrics

- From previous slide, threshold $f(x) = \frac{p(y=1|x)}{p(y=0|x)}$ at $\tau = \frac{L_{FP}}{L_{FN}}$ to make classification decision
- When comparing two algorithms, we often do not know (or do not want) to define L_{FP} and L_{FN} because for the same dataset (e.g., facebook photos), loss might be application specific
- Instead of thresholding $f(x)$ at $\tau = \frac{L_{FP}}{L_{FN}}$, we threshold at different τ, and compute for each τ
- TP (true positives)
  - N_{TP} = # data points in test set whose true label = 1 & classified correctly as 1
  - N_1 = # data points in test set whose true label = 1
  - TPR (true positive rate or sensitivity or recall) = N_{TP} / N_1
- TN (true negatives)
  - N_{TN} = # data points in test set whose true label = 0 and classified correctly as 0
  - N_0 = # data points in test set whose true label = 0
  - TNR (true negative rate or specificity) = N_{TN} / N_0

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	N_{TN}	N_{FP}	$N_0 = N_{TN} + N_{FP}$
$y = 1$	N_{FN}	N_{TP}	$N_1 = N_{FN} + N_{TP}$

Different Metrics

- From previous slide, threshold $f(x) = \frac{p(y=1|x)}{p(y=0|x)}$ at $\tau = \frac{L_{FP}}{L_{FN}}$ to make classification decision
- When comparing two algorithms, we often do not know (or do not want) to define L_{FP} and L_{FN} because for the same dataset (e.g., facebook photos), loss might be application specific
- Instead of thresholding $f(x)$ at $\tau = \frac{L_{FP}}{L_{FN}}$, we threshold at different τ, and compute for each τ
- TP (true positives)
  - N_{TP} = # data points in test set whose true label = 1 & classified correctly as 1
  - N_1 = # data points in test set whose true label = 1
  - TPR (true positive rate or sensitivity or recall) = N_{TP} / N_1

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	N_{TN}	N_{FP}	$N_0 = N_{TN} + N_{FP}$
$y = 1$	N_{FN}	N_{TP}	$N_1 = N_{FN} + N_{TP}$

- TN (true negatives)
  - N_{TN} = # data points in test set whose true label = 0 and classified correctly as 0
  - N_0 = # data points in test set whose true label = 0
  - TNR (true negative rate or specificity) = N_{TN} / N_0
- FP (false positives)
  - N_{FP} = # data points in test set whose true label = 0 and classified wrongly as 1
  - FPR (false positive rate or type 1 error) = N_{FP} / N_0

Different Metrics

- From previous slide, threshold $f(x) = \frac{p(y=1|x)}{p(y=0|x)}$ at $\tau = \frac{L_{FP}}{L_{FN}}$ to make classification decision
- When comparing two algorithms, we often do not know (or do not want) to define L_{FP} and L_{FN} because for the same dataset (e.g., facebook photos), loss might be application specific
- Instead of thresholding $f(x)$ at $\tau = \frac{L_{FP}}{L_{FN}}$, we threshold at different τ, and compute for each τ
- TP (true positives)
  - N_{TP} = # data points in test set whose true label = 1 & classified correctly as 1
  - N_1 = # data points in test set whose true label = 1
  - TPR (true positive rate or sensitivity or recall) = N_{TP} / N_1

	ŷ = 0	ŷ = 1	
y = 0	N_{TN}	N_{FP}	$N_0 = N_{TN} + N_{FP}$
y = 1	N_{FN}	N_{TP}	$N_1 = N_{FN} + N_{TP}$

- TN (true negatives)
  - N_{TN} = # data points in test set whose true label = 0 and classified correctly as 0
  - N_0 = # data points in test set whose true label = 0
  - TNR (true negative rate or specificity) = N_{TN} / N_0
- FP (false positives)
  - N_{FP} = # data points in test set whose true label = 0 and classified wrongly as 1
  - FPR (false positive rate or type 1 error) = N_{FP} / N_0
- FN (false negatives)
  - N_{FN} = # data points in test set whose true label = 1 and classified wrongly as 0
  - FNR (false negative rate or type 2 error) = N_{FN} / N_1

Different Metrics

- From previous slide, threshold $f(x) = \frac{p(y=1|x)}{p(y=0|x)}$ at $\tau = \frac{L_{FP}}{L_{FN}}$ to make classification decision
- When comparing two algorithms, we often do not know (or do not want) to define L_{FP} and L_{FN} because for the same dataset (e.g., facebook photos), loss might be application specific
- Instead of thresholding $f(x)$ at $\tau = \frac{L_{FP}}{L_{FN}}$, we threshold at different τ, and compute for each τ

	$\hat{y} = 0$	$\hat{y} = 1$	
y = 0	N_{TN}	N_{FP}	$N_0 = N_{TN} + N_{FP}$
y = 1	N_{FN}	N_{TP}	$N_1 = N_{FN} + N_{TP}$

- TP (true positives)
  - N_{TP} = # data points in test set whose true label = 1 & classified correctly as 1
  - N_1 = # data points in test set whose true label = 1
  - TPR (true positive rate or sensitivity or recall) = N_{TP} / N_1
- TN (true negatives)
  - N_{TN} = # data points in test set whose true label = 0 and classified correctly as 0
  - N_0 = # data points in test set whose true label = 0
  - TNR (true negative rate or specificity) = N_{TN} / N_0
- FP (false positives)
  - N_{FP} = # data points in test set whose true label = 0 and classified wrongly as 1
  - FPR (false positive rate or type 1 error) = N_{FP} / N_0
- FN (false negatives)
  - N_{FN} = # data points in test set whose true label = 1 and classified wrongly as 0
  - FNR (false negative rate or type 2 error) = N_{FN} / N_1
- Many other possibilities: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

ROC (1)

- Previously, we threshold $f(x) = \frac{p(y=1|x)}{p(y=0|x)}$ at different τ and compute different metrics, e.g., TPR(τ) and FPR(τ)
- But f(x) does not need to be $\frac{p(y=1|x)}{p(y=0|x)}$
- All classifiers (even non-probabilistic classifiers, e.g., support vector machines) will output a number (e.g., between 0 and 1), which can then be thresholded to give a final classification output.
- Thus, for any classifier f(x), we can threshold at different τ and compute TPR(τ) and FPR(τ)
- Plot of TPR(τ) against FPR(τ) is called receiver operating characteristic (ROC) curve: strange name because it was invented during World War II for analyzing radar signals

ROC Curve (2)

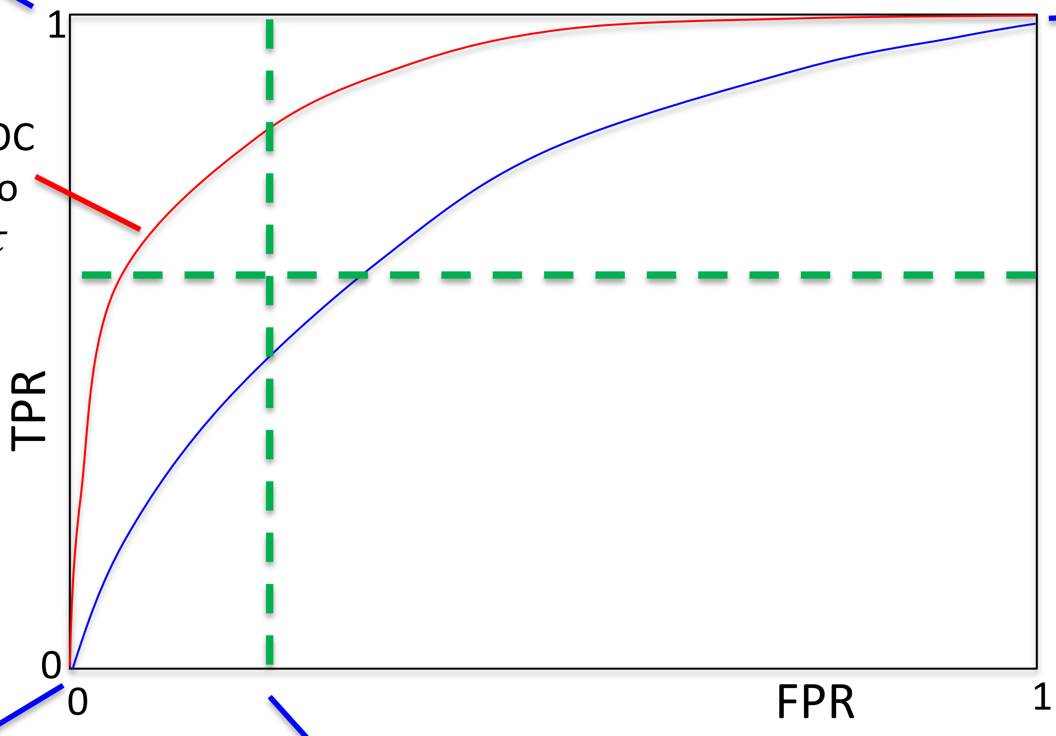
- Red and blue curves are ROC curves for classifiers A and B

100% accuracy
(TPR = 1, FPR = 0)

Every data point
classified as 1, so
TPR = 1 & FPR = 1

Every location on ROC
curve corresponds to
different threshold τ

"A" better than "B"
because for any TPR,
"A" achieves lower FPR
than B

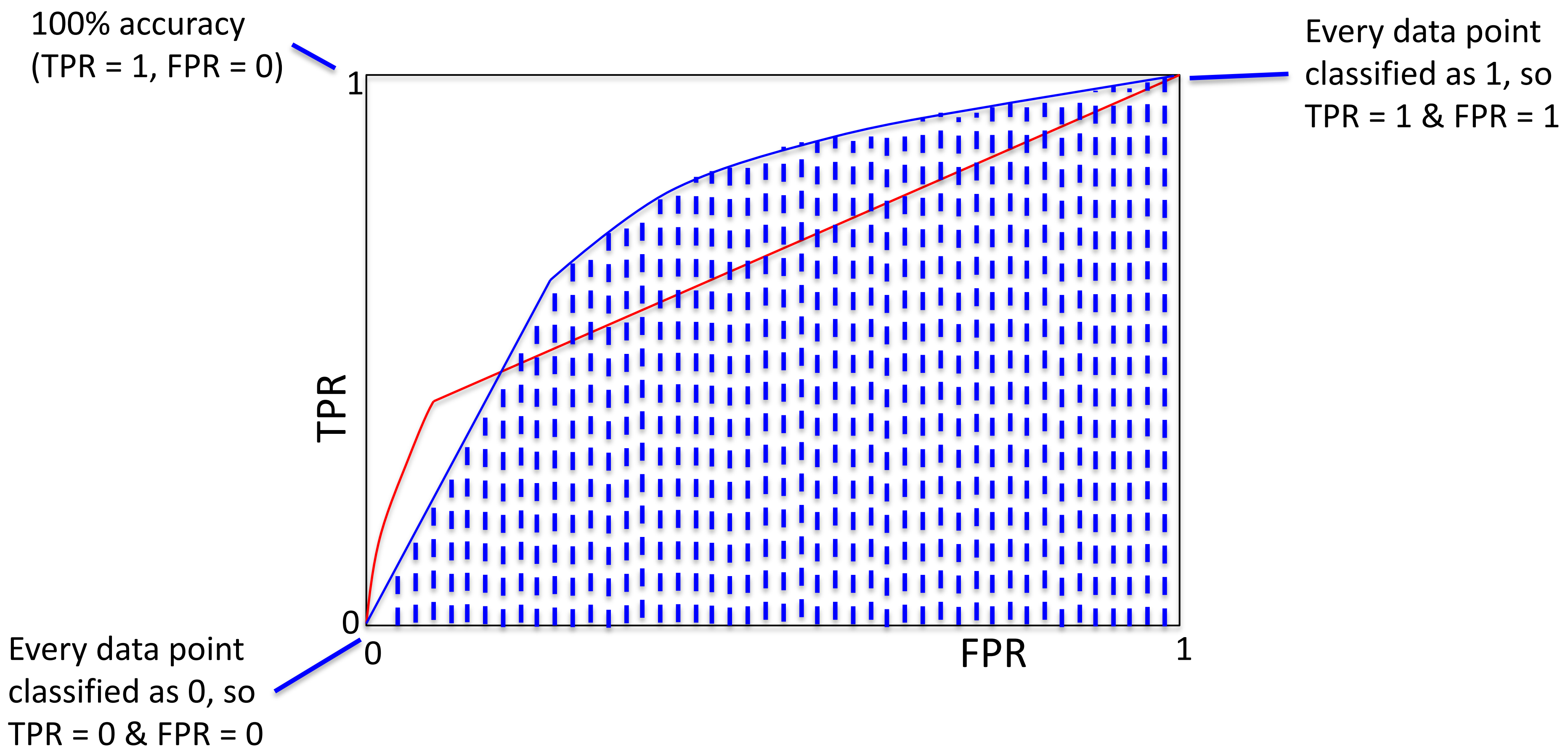


Every data point
classified as 0, so
TPR = 0 & FPR = 0

"A" better than "B" because for any FPR,
"A" achieves higher TPR than B

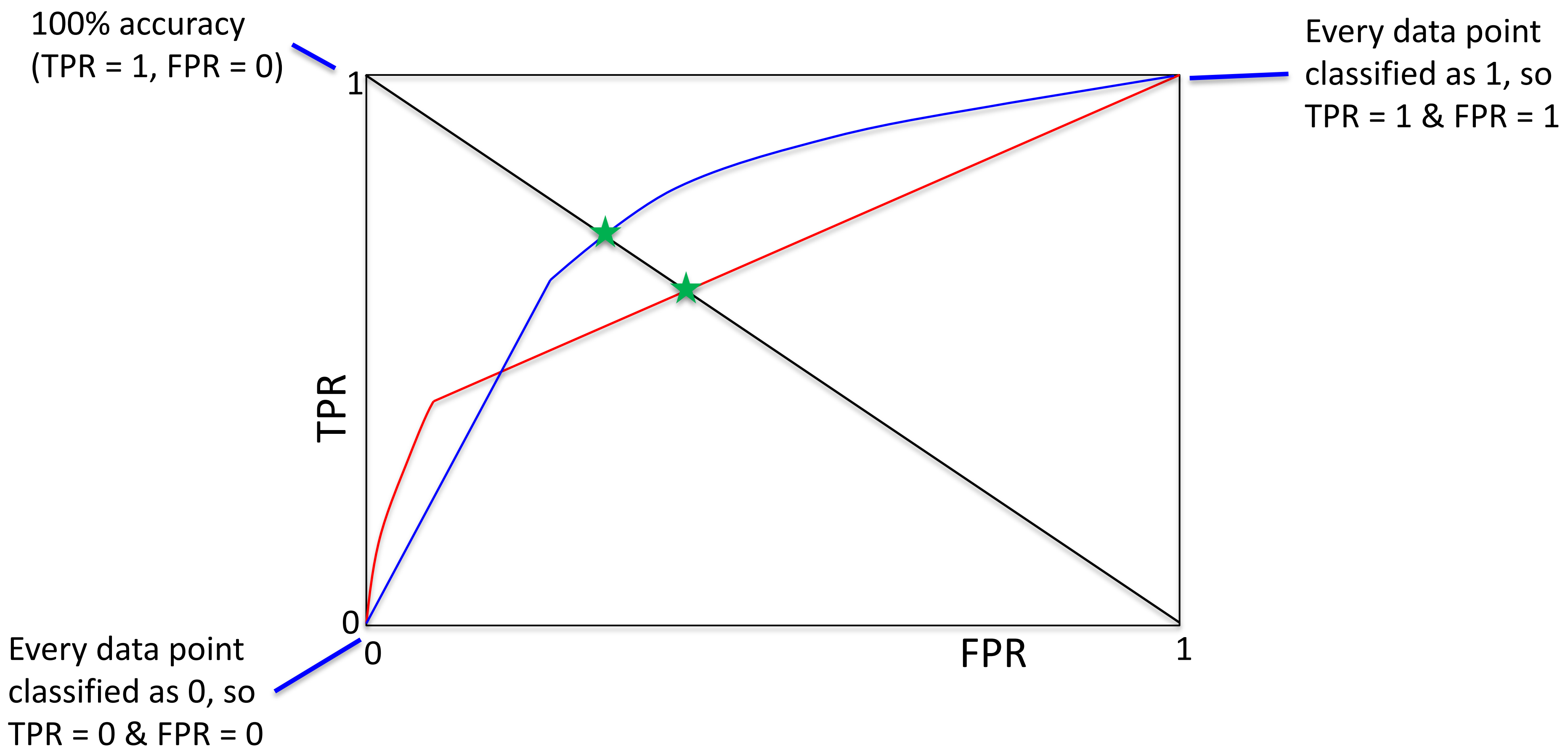
ROC Curve (2)

- Red and blue curves are ROC curves for classifiers A and B
- Sometimes two curves intersect, so useful to summarize ROC curve with one number
  - Blue area = area under the curve (AUC) for classifier B



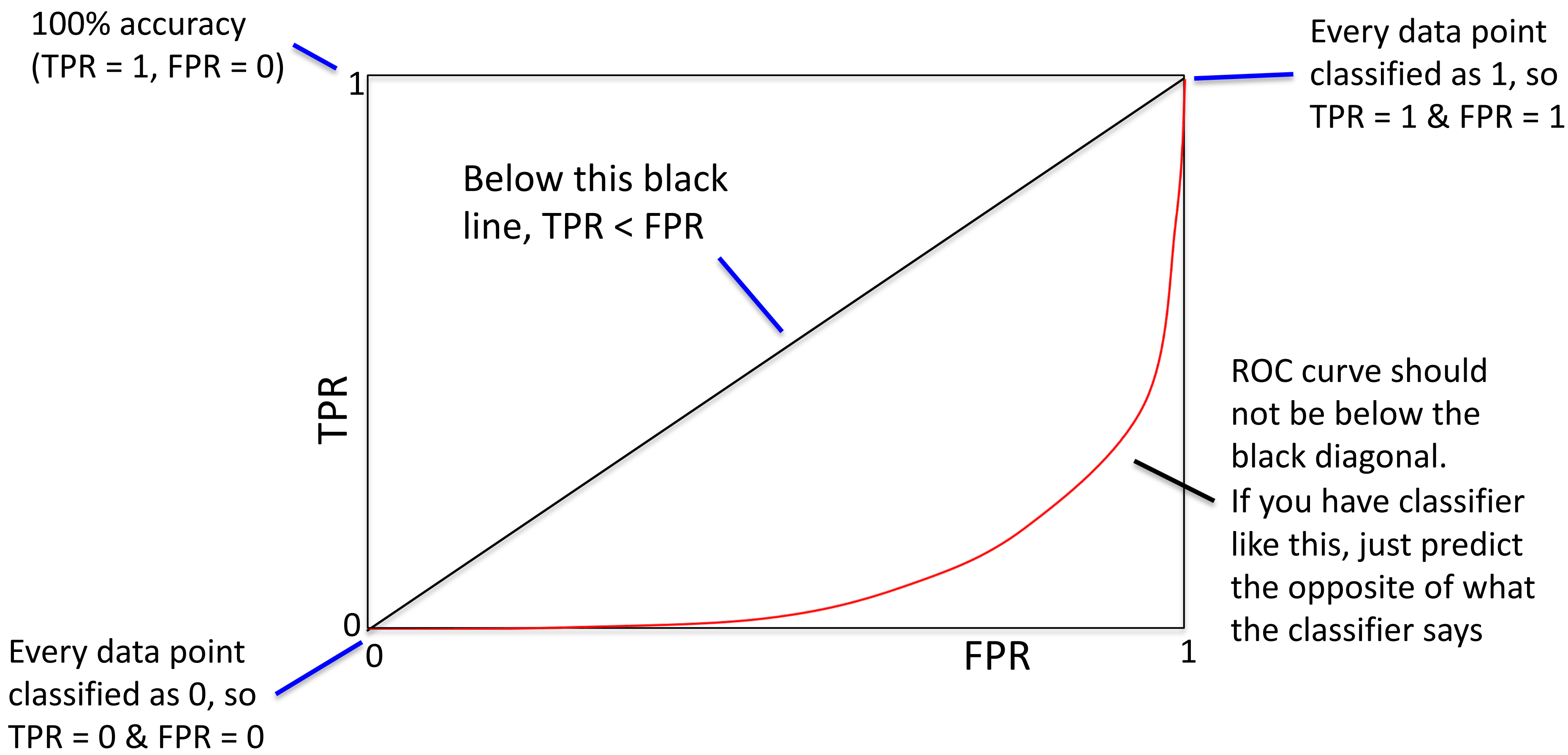
ROC Curve (2)

- Red and blue curves are ROC curves for classifiers A and B
- Sometimes two curves intersect, so useful to summarize ROC curve with one number
  - Blue area = area under the curve (AUC) for classifier B
  - Intersection between black diagonal line and ROC: equal error rate (EER)



ROC Curve (2)

- Red and blue curves are ROC curves for classifiers A and B
- Sometimes two curves intersect, so useful to summarize ROC curve with one number
  - Blue area = area under the curve (AUC) for classifier B
  - Intersection between black diagonal line and ROC: equal error rate (ERR)



Summary

- Problems With MAP estimation
  - Mode (MAP) of a distribution might be atypical
  - MAP sensitive to parameterization
- Bayesian model selection
  - Automatically select for sufficiently complex (but not too complex) model that can explain data well
  - Approximations often needed, e.g., BIC
- Bayesian decision theory
  - Minimize posterior expected loss when making decisions
  - ROC curves

Optional Reading

- Notes based on
  - KM Chapter 5 (beware of typos)

Additional Material

Laplace Approximation

- Let $p(\theta|D) = \frac{e^{-E(\theta)}}{p(D)}$, where $E(\theta) = -\log p(\theta, D)$ and $\theta \in \mathbb{R}^M$.
- Let θ^* be mode of $\log p(\theta, D)$, then by Taylor expansion:

$$E(\theta) \approx E(\theta^*) + (\theta - \theta^*)^T g + 0.5(\theta - \theta^*)^T H(\theta - \theta^*),$$

where $g = \nabla E(\theta)|_{\theta^*}, H = \frac{\partial^2 E(\theta)}{\partial\theta\partial\theta^T}\Big|_{\theta^*}$

- $g = 0$ since θ^* is mode, so

$$\begin{aligned} p(\theta|D) &\approx \frac{1}{p(D)} e^{-E(\theta^*)} \exp\left[-\frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*) \right] \\ &= \mathcal{N}(\theta|\theta^*, H^{-1}) \end{aligned}$$

- Since the normalization constant of MVN is $\frac{1}{(2\pi)^{M/2}|H|^{1/2}}$, we have

$$\frac{1}{(2\pi)^{M/2}|H|^{1/2}} = \frac{1}{p(D)} e^{-E(\theta^*)} \implies p(D) = e^{-E(\theta^*)}(2\pi)^{M/2}|H|^{-1/2}$$

Proof of BIC

- From previous slide: $p(D) \approx e^{-E(\theta^*)}(2\pi)^{M/2}|H|^{-1/2}$
- Taking log, we have

$$\log p(D) \approx \log p(D|\theta^*) + \log p(\theta^*) + \frac{M}{2}\log 2\pi - \frac{1}{2}\log|H|$$

- Now $H = \sum_{n=1}^{N} H_i$, where $H_i = \nabla\nabla \log p(D_i|\theta)$. Approximating each H_i by $\hat{H}$, we have

$$\log|H| = \log|N\hat{H}| = \log(N^M|\hat{H}|) = M\log N + \log|\hat{H}|$$

- Therefore

$$\begin{aligned}\log p(D) &\approx \log p(D|\theta^*) + \log p(\theta^*) + \frac{M}{2}\log 2\pi - \frac{M}{2}\log N - \frac{1}{2}\log|\hat{H}| \\ &\approx \log p(D|\theta_{ML}) - \frac{M}{2}\log N,\end{aligned}$$

  where we assumed

  - $p(\theta) \propto 1$, so drop off $p(\theta^*)$ and substitute θ^* with θ_{ML}
  - $\log N$ dominates $\log 2\pi$ and $\log|\hat{H}|$, so the two terms can be dropped