

# Classifiers: Naïve Bayes, evaluation

Tomáš Svoboda and Matěj Hoffmann

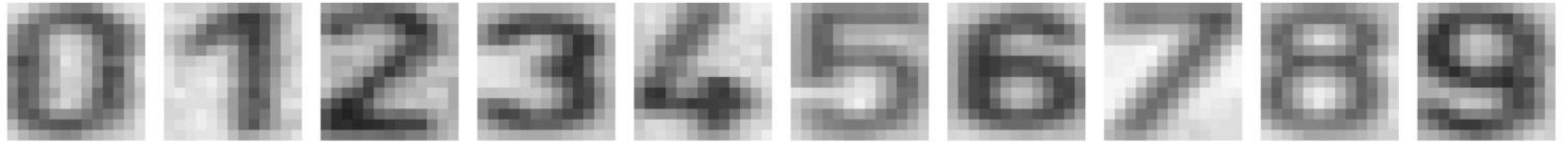
thanks to Daniel Novák and Filip Železný, Ondřej Drbohlav

Vision for Robots and Autonomous Systems, Center for Machine Perception  
Department of Cybernetics  
Faculty of Electrical Engineering, Czech Technical University in Prague

May 13, 2020

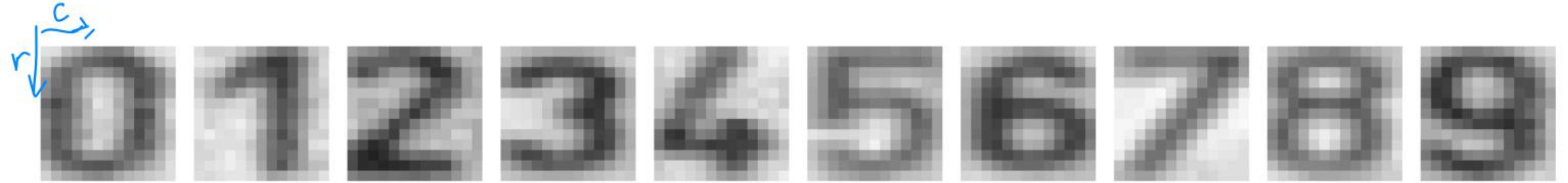
ok		13	100%
slabe	0		0%
nula	0		0%

## Example: Digit recognition/classification



- ▶ **Input:** 8-bit image  $13 \times 13$ , pixel intensities 0 – 255. (0 means black, 255 means white)
- ▶ **Output:** Digit 0 – 9. Decision about the class, classification.
- ▶ **Features:** Pixel intensities ...

## Example: Digit recognition/classification



- ▶ **Input:** 8-bit image  $13 \times 13$ , pixel intensities 0 – 255. (0 means black, 255 means white)
- ▶ **Output:** Digit 0 – 9. Decision about the class, classification.
- ▶ **Features:** Pixel intensities ...

# Classification as a special case of statistical decision theory

- ▶ Attribute vector  $\vec{x} = (x_1, x_2, \dots)$ : pixels 1, 2, .....
- ▶ **State set  $\mathcal{S}$  = decision set  $\mathcal{D} = \{0, 1, \dots, 9\}$ .**
- ▶ State = actual class, Decision = recognized class
- ▶ Loss function:

$$l(s, d) = \begin{cases} 0, & d = s \\ 1, & d \neq s \end{cases}$$

$$\delta^*(\vec{x}) = \arg \min_d \sum_s \underbrace{l(s, d)}_{0 \text{ if } d=s} P(s|\vec{x}) = \arg \min_d \sum_{s \neq d} P(s|\vec{x})$$

Obviously  $\sum_s P(s|\vec{x}) = 1$ , then:

$$P(d|\vec{x}) + \sum_{s \neq d} P(s|\vec{x}) = 1$$

Inserting into above:

$$\delta^*(\vec{x}) = \arg \min_d [1 - P(d|\vec{x})] = \arg \max_d P(d|\vec{x})$$

# Classification as a special case of statistical decision theory

- ▶ Attribute vector  $\vec{x} = (x_1, x_2, \dots)$ : pixels 1, 2, <sup>132</sup>...
- ▶ **State set  $\mathcal{S}$  = decision set  $\mathcal{D} = \{0, 1, \dots, 9\}$ .**
- ▶ State = actual class, Decision = recognized class
- ▶ Loss function:

$$l(s, d) = \begin{cases} 0, & d = s \\ 1, & d \neq s \end{cases}$$

$$\delta^*(\vec{x}) = \arg \min_d \sum_s \underbrace{l(s, d)}_{0 \text{ if } d=s} P(s|\vec{x}) = \arg \min_d \sum_{s \neq d} P(s|\vec{x})$$

Obviously  $\sum_s P(s|\vec{x}) = 1$ , then:

$$P(d|\vec{x}) + \sum_{s \neq d} P(s|\vec{x}) = 1$$

$$s^* = \arg \max_{s \in \mathcal{S}} P(s|\vec{x})$$

Inserting into above:

$$\delta^*(\vec{x}) = \arg \min_d [1 - P(d|\vec{x})] = \arg \max_d P(d|\vec{x})$$

# Bayes classification in practice

- ▶ Usually we are not given  $P(s|\vec{x})$
- ▶ It has to be estimated from already classified examples – training data
- ▶ For discrete  $\vec{x}$ , training examples  $(\vec{x}_1, s_1), (\vec{x}_2, s_2), \dots, (\vec{x}_l, s_l)$ 
  - ▶ so-called i.i.d (independent, identically distributed) multiset
  - ▶ every  $(\vec{x}_i, s)$  is drawn independently from  $P(\vec{x}, s)$
- ▶ Without knowing anything about the distribution, a non-parametric estimate:

$$P(s|\vec{x}) = \frac{P(\vec{x}, s)}{P(\vec{x})} \approx \frac{\# \text{ examples where } \vec{x}_i = \vec{x} \text{ and } s_i = s}{\# \text{ examples where } \vec{x}_i = \vec{x}}$$

- ▶ Hard in practice:
  - ▶ To reliably estimate  $P(s|\vec{x})$ , the number of examples grows exponentially with the number of elements of  $\vec{x}$ .
    - ▶ e.g. with the number of pixels in images
    - ▶ curse of dimensionality
    - ▶ denominator often 0

# Bayes classification in practice

- ▶ Usually we are not given  $P(s|\vec{x})$
- ▶ It has to be estimated from already classified examples – training data
- ▶ For discrete  $\vec{x}$ , training examples  $(\vec{x}_1, s_1), (\vec{x}_2, s_2), \dots, (\vec{x}_l, s_l)$ 
  - ▶ so-called i.i.d (independent, identically distributed) multiset
  - ▶ every  $(\vec{x}_i, s)$  is drawn independently from  $P(\vec{x}, s)$
- ▶ Without knowing anything about the distribution, a non-parametric estimate:

$$P(s|\vec{x}) = \frac{P(\vec{x}, s)}{P(\vec{x})} \approx \frac{\# \text{ examples where } \vec{x}_i = \vec{x} \text{ and } s_i = s}{\# \text{ examples where } \vec{x}_i = \vec{x}}$$

- ▶ Hard in practice:
  - ▶ To reliably estimate  $P(s|\vec{x})$ , the number of examples grows exponentially with the number of elements of  $\vec{x}$ .
    - ▶ e.g. with the number of pixels in images
    - ▶ curse of dimensionality
    - ▶ denominator often 0

$\dim(\vec{x}) = 13^2 \quad 256$   
 $(13^2)^{256}$  možemj & obratit

# Naïve Bayes classification

- ▶ For efficient classification we must thus rely on additional assumptions.
- ▶ In the exceptional case of **statistical independence** between components of  $\vec{x}$  for each class  $s$  it holds

$$P(\vec{x}|s) = P(x[1]|s) \cdot P(x[2]|s) \cdot \dots$$

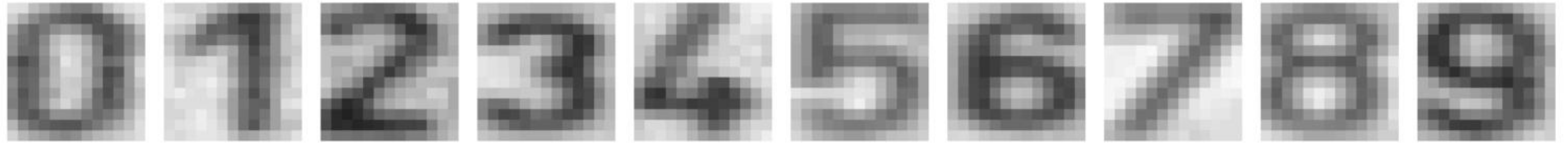
- ▶ Use simple Bayes law and maximize:

$$P(s|\vec{x}) = \frac{P(\vec{x}|s)P(s)}{P(\vec{x})} = \frac{P(s)}{P(\vec{x})} P(x[1]|s) \cdot P(x[2]|s) \cdot \dots =$$

- ▶ No combinatorial curse in estimating  $P(s)$  and  $P(x[i]|s)$  separately for each  $i$  and  $s$ .
- ▶ No need to estimate  $P(\vec{x})$ . (Why?)
- ▶  $P(s)$  may be provided apriori.
- ▶ **naïve** = when used despite statistical dependence



## Example: Digit recognition/classification

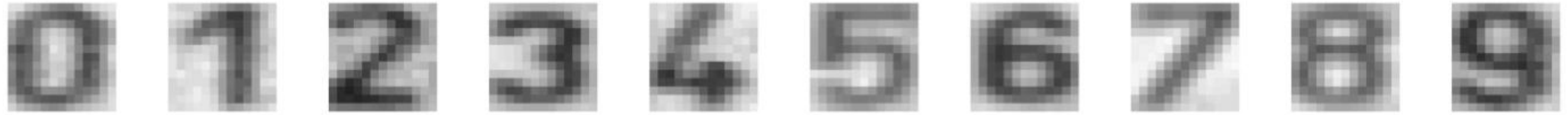


- ▶ **Input:** 8-bit image  $13 \times 13$ , pixel intensities 0 – 255. (0 means black, 255 means white)
- ▶ **Output:** Digit 0 – 9. Decision about the class, classification.
- ▶ **Features:** Pixel intensities ...

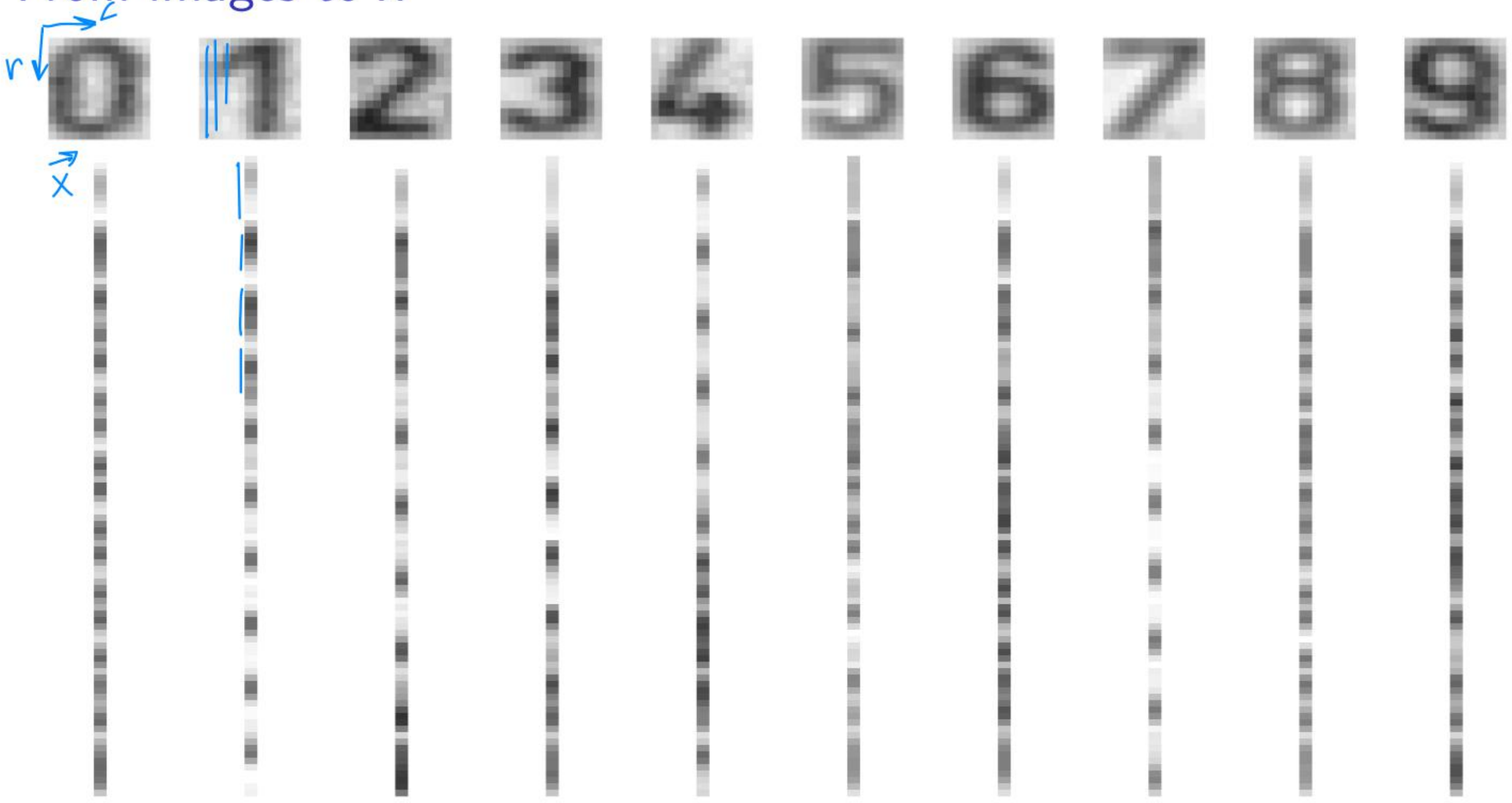
Collect **data** , ...

- ▶  $P(\vec{x})$ . What is the dimension of  $\vec{x}$ ? How many possible images?
- ▶ Learn  $P(\vec{x}|s)$  per each class (digit).
- ▶ Classify  $s^* = \operatorname{argmax}_s P(s|\vec{x})$ .

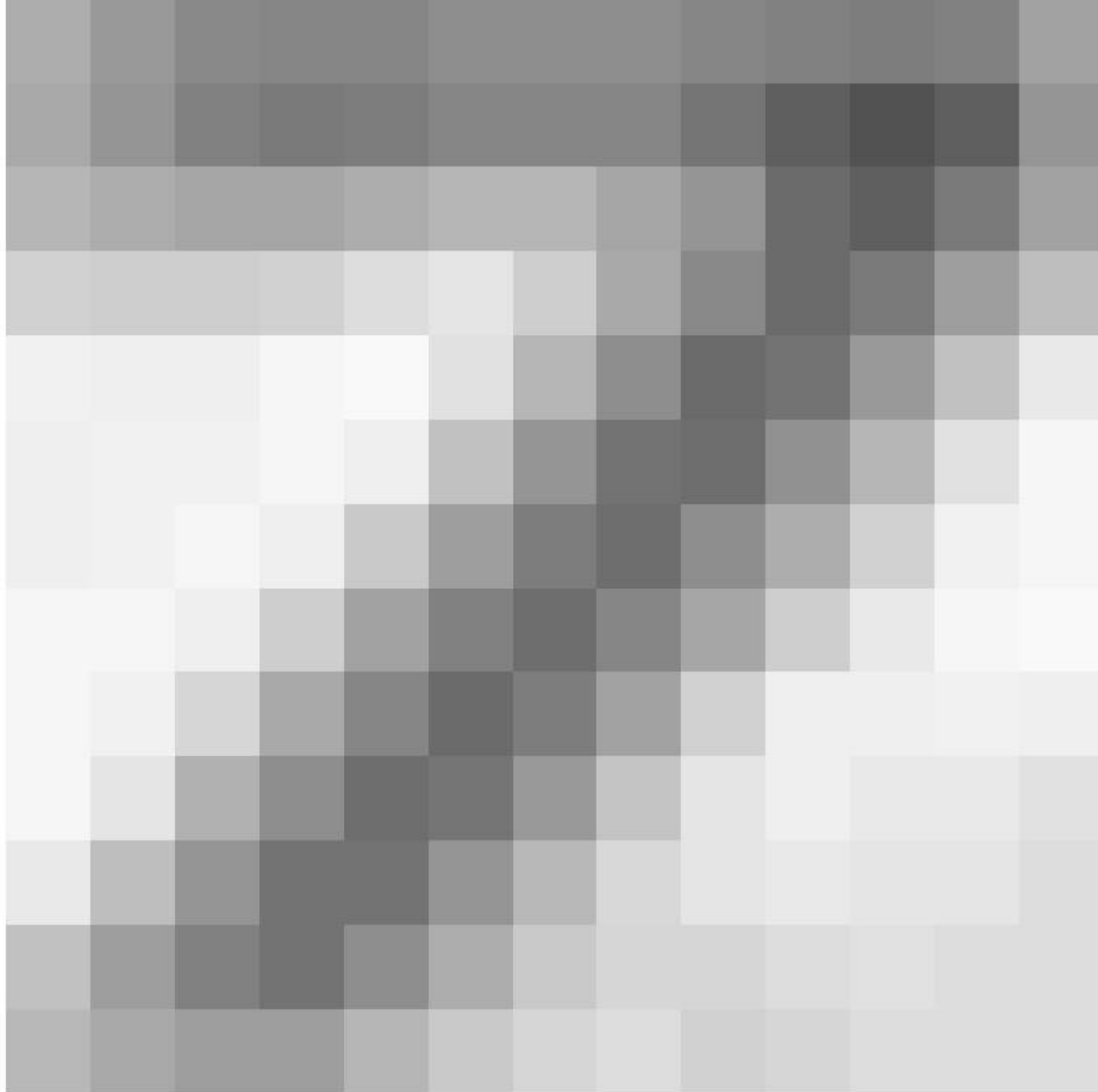
From images to  $\vec{x}$



From images to  $\vec{x}$

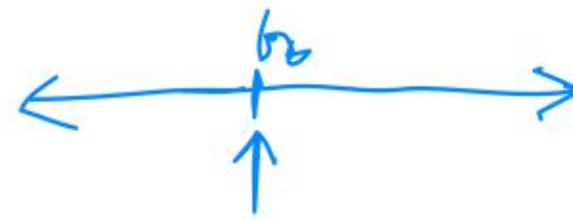
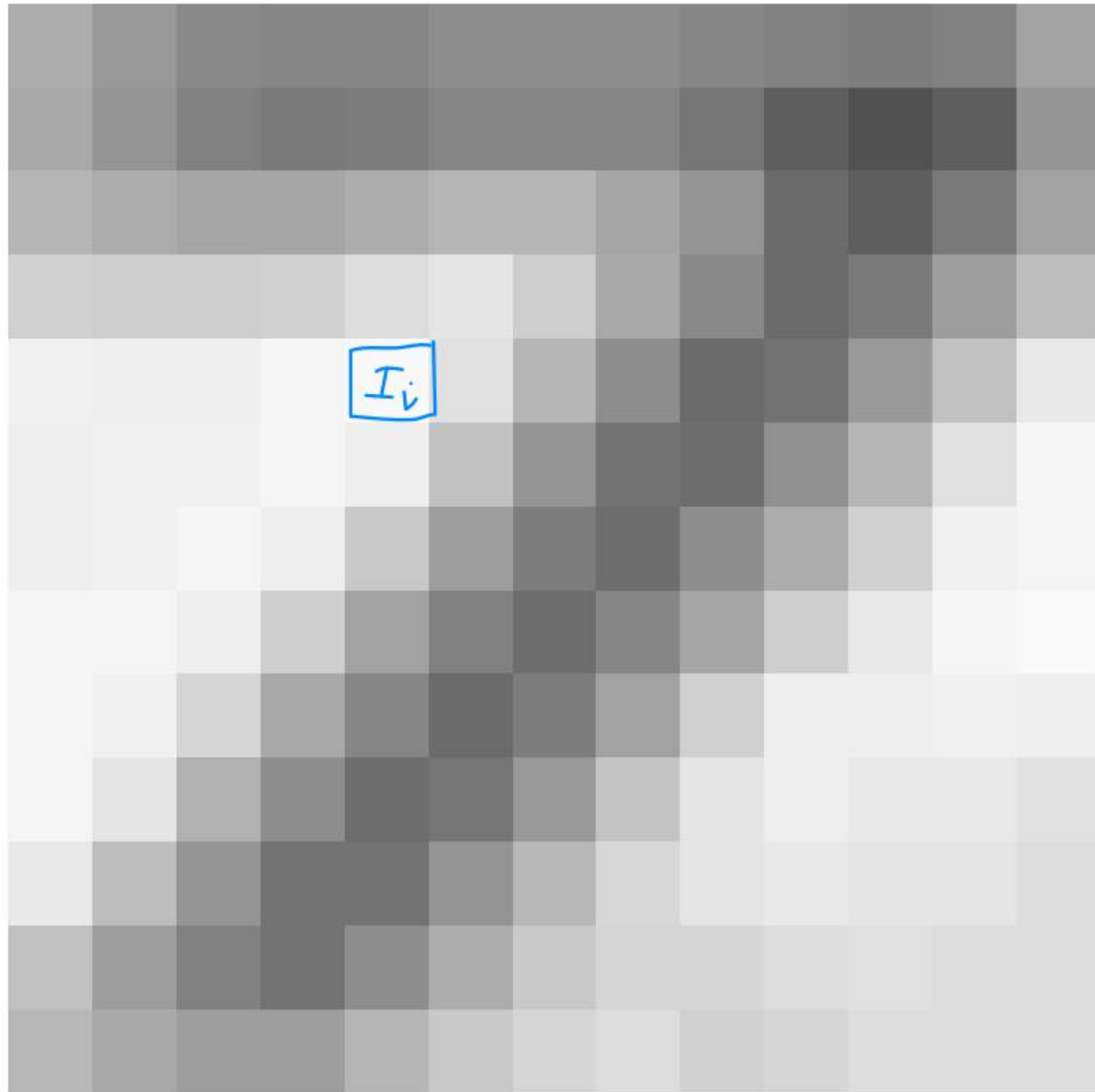


# Conditional probabilities, likelihoods



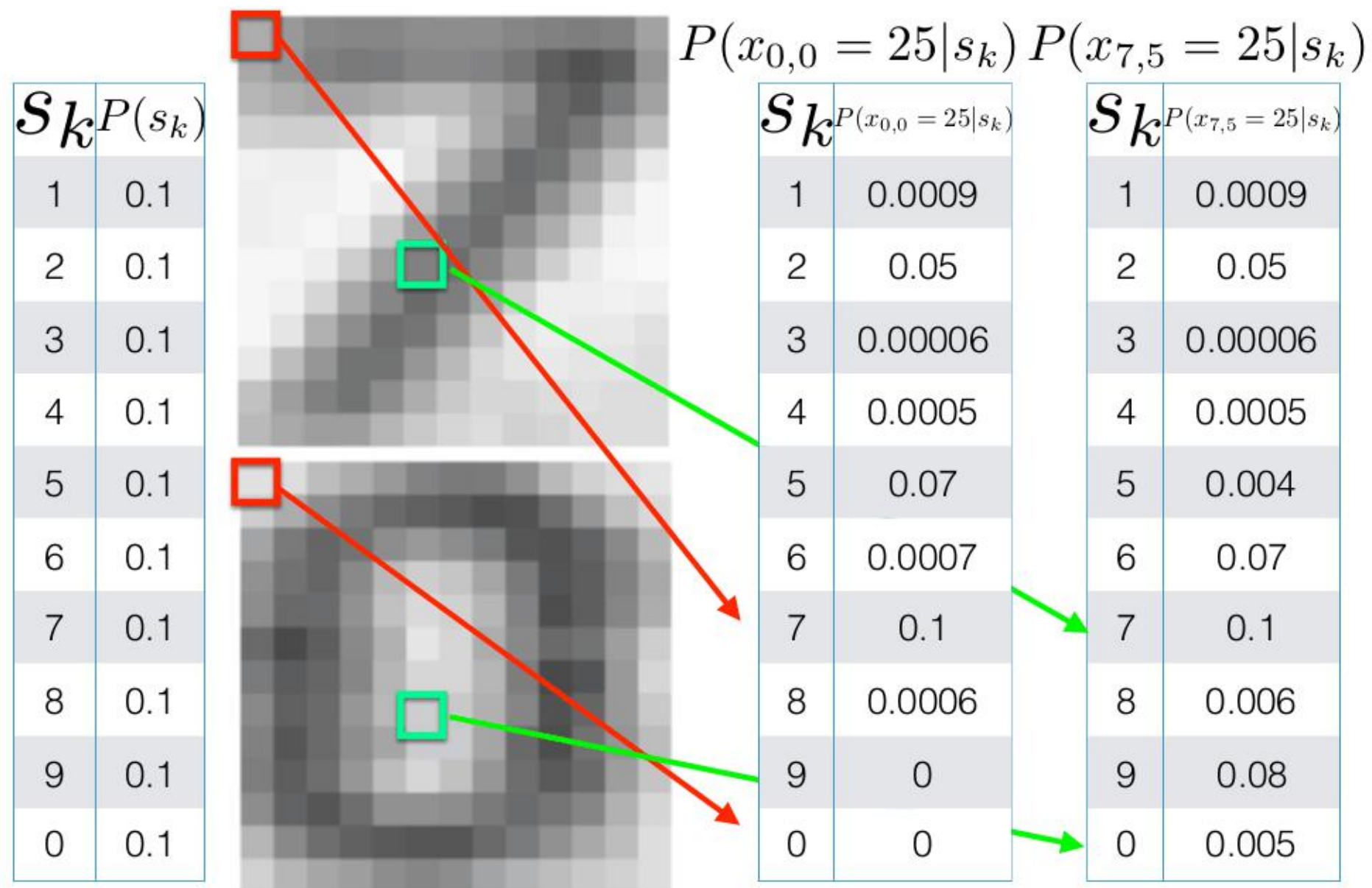
- ▶ Apriori digit probabilities  $P(s_k)$
- ▶ Likelihoods for pixels.  $P(x_{r,c} = l_i | s_k)$

# Conditional probabilities, likelihoods

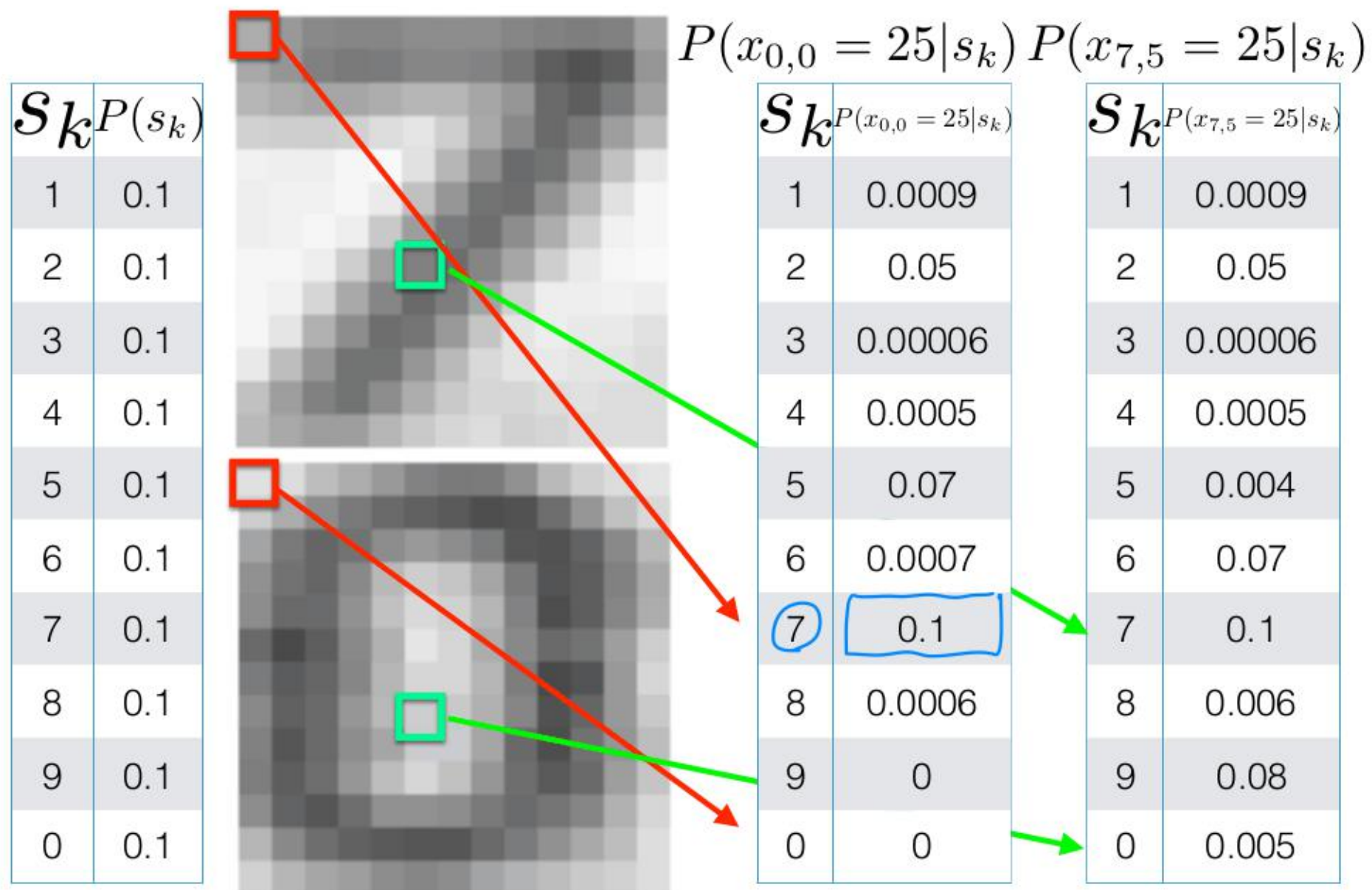


- ▶ Apriori digit probabilities  $P(s_k)$
- ▶ Likelihoods for pixels.  $P(x_{r,c} = I_i | s_k)$

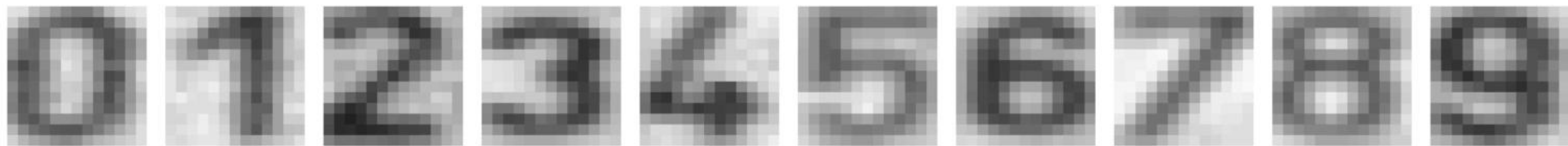
# Conditional likelihoods



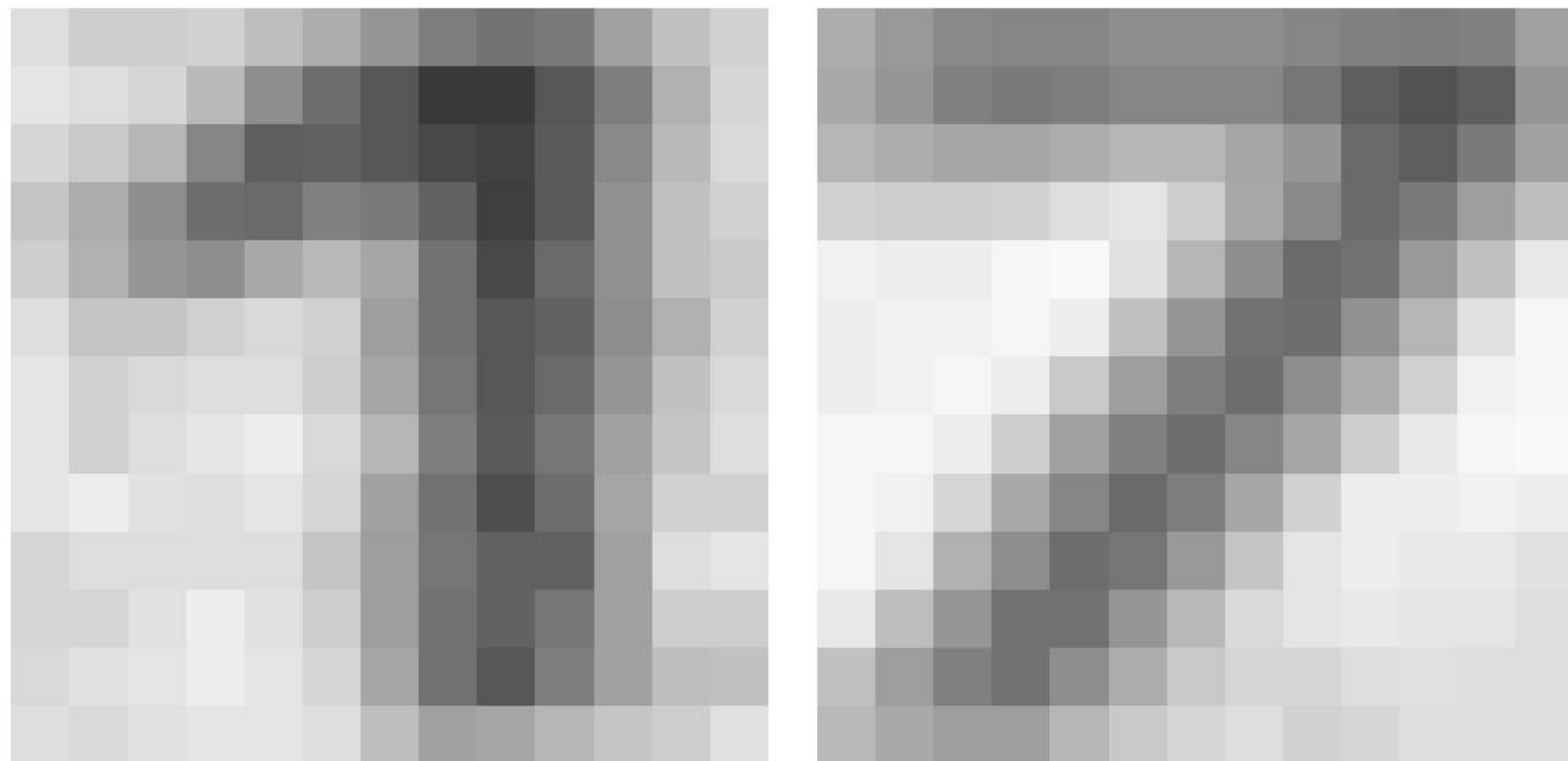
# Conditional likelihoods



## Unseen events



Images  $13 \times 13$ , intensities 0 – 255, 100 exemplars per each class.

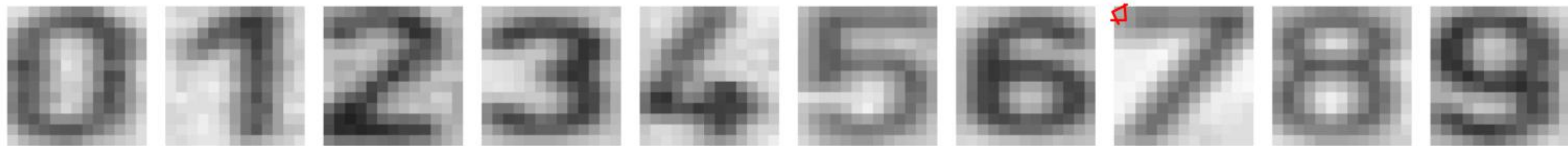




# Unseen events

$$P(\vec{x} | S) = P(x_{1,1} | S) P(x_{1,2} | S) \dots$$

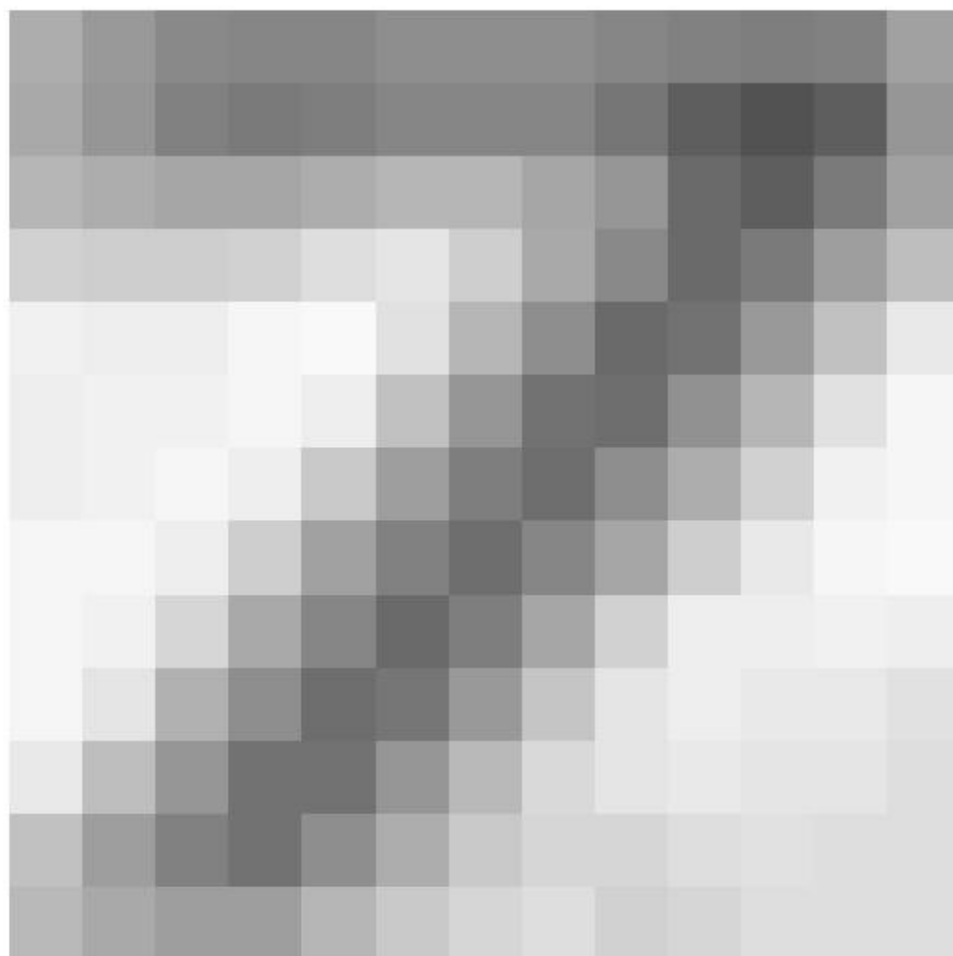
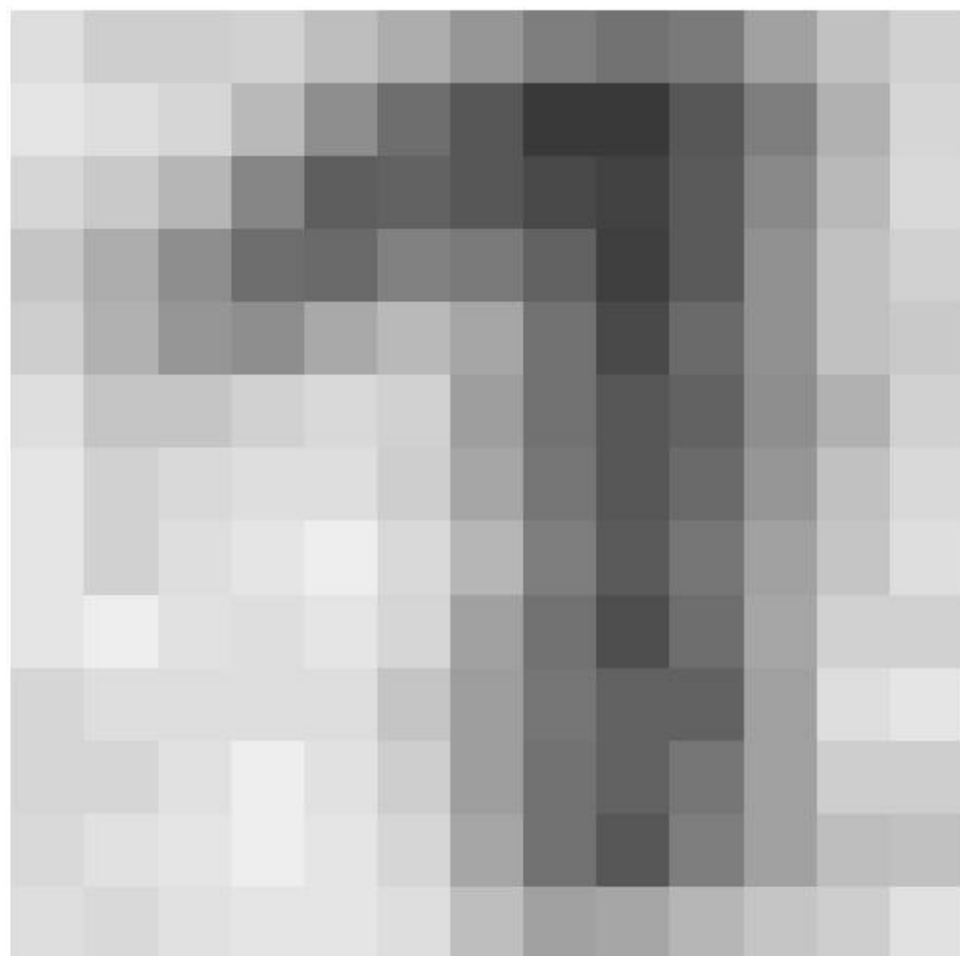
$$P(x_{0,0} = 25 | \overline{7}) = \emptyset$$



Images  $13 \times 13$ , intensities 0 – 255, 100 exemplars per each class.

$$P(x_{0,0} = 24 | \overline{7}) = 0,001$$
$$P(x_{0,0} = 26 | \overline{7}) = 0,002$$

$$x_{0,0} = 25$$



# Laplace smoothing (“additive smoothing”)

$$P(x) = \frac{\text{count}(x)}{\text{total samples}}$$

Problem:  $\text{count}(x) = 0$

Pretend you see the (any) sample one more time.

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{N + |X|}$$

where  $N$  is the number of (total) observations;  $|X|$  is the number of possible values  $X$  can take (cardinality).

# Laplace smoothing (“additive smoothing”)

$$x \sim x_{\text{opt}} = 25$$

$$P(x) = \frac{\text{count}(x)}{\text{total samples}}$$

Problem:  $\text{count}(x) = \underline{0}$

Pretend you see the (any) sample one more time.

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{N + |X|}$$

where  $N$  is the number of (total) observations;  $|X|$  is the number of possible values  $X$  can take (cardinality).

## Laplace smoothing ( “additive smoothing” )

$$P(x) = \frac{\text{count}(x)}{\text{total samples}}$$

Problem:  $\text{count}(x) = 0$

Pretend you see the (any) sample one more time.

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{N + |X|}$$

where  $N$  is the number of (total) observations;  $|X|$  is the number of possible values  $X$  can take (cardinality).

# Laplace smoothing (“additive smoothing”)

$$P(x) = \frac{\text{count}(x)}{\text{total samples}}$$

$$x \in \{0, \dots, 255\}$$

Problem:  $\text{count}(x) = 0$

Pretend you see the (any) sample one more time.

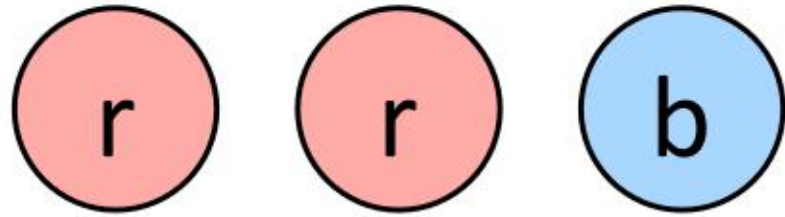
$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + 1}{N + |X|}$$

where  $N$  is the number of (total) observations;  $|X|$  is the number of possible values  $X$  can take (cardinality).

$$P_{\text{LAP}}(X) = ?$$

Observation:



What is  $P_{\text{LAP}}(X = \text{red})$  and  $P_{\text{LAP}}(X = \text{blue})$ ?

A:  $P_{\text{LAP}}(X = \text{red}) = 7/10$ ,  $P_{\text{LAP}}(X = \text{blue}) = 3/10$

B:  $P_{\text{LAP}}(X = \text{red}) = 2/3$ ,  $P_{\text{LAP}}(X = \text{blue}) = 1/3$

C:  $P_{\text{LAP}}(X = \text{red}) = 3/5$ ,  $P_{\text{LAP}}(X = \text{blue}) = 2/5$

D: None of the above.

$$P_{LAP}(X) = ?$$

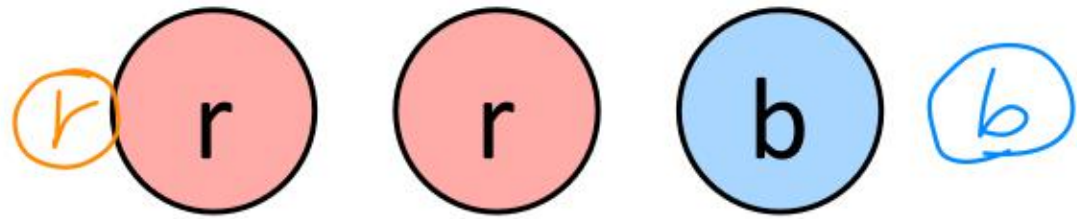
$$P(X=r) = \frac{2}{3}$$

$$P(X=b) = \frac{1}{3}$$

$$P_{LAP}(X=r) = \frac{3}{5}$$

$$|X|=2$$

Observation:



What is  $P_{LAP}(X = \text{red})$  and  $P_{LAP}(X = \text{blue})$ ?

A:  $P_{LAP}(X = \text{red}) = 7/10$ ,  $P_{LAP}(X = \text{blue}) = 3/10$

B:  $P_{LAP}(X = \text{red}) = 2/3$ ,  $P_{LAP}(X = \text{blue}) = 1/3$

**C:**  $P_{LAP}(X = \text{red}) = 3/5$ ,  $P_{LAP}(X = \text{blue}) = 2/5$

D: None of the above.



# Laplace smoothing - as a hyperparameter $k$

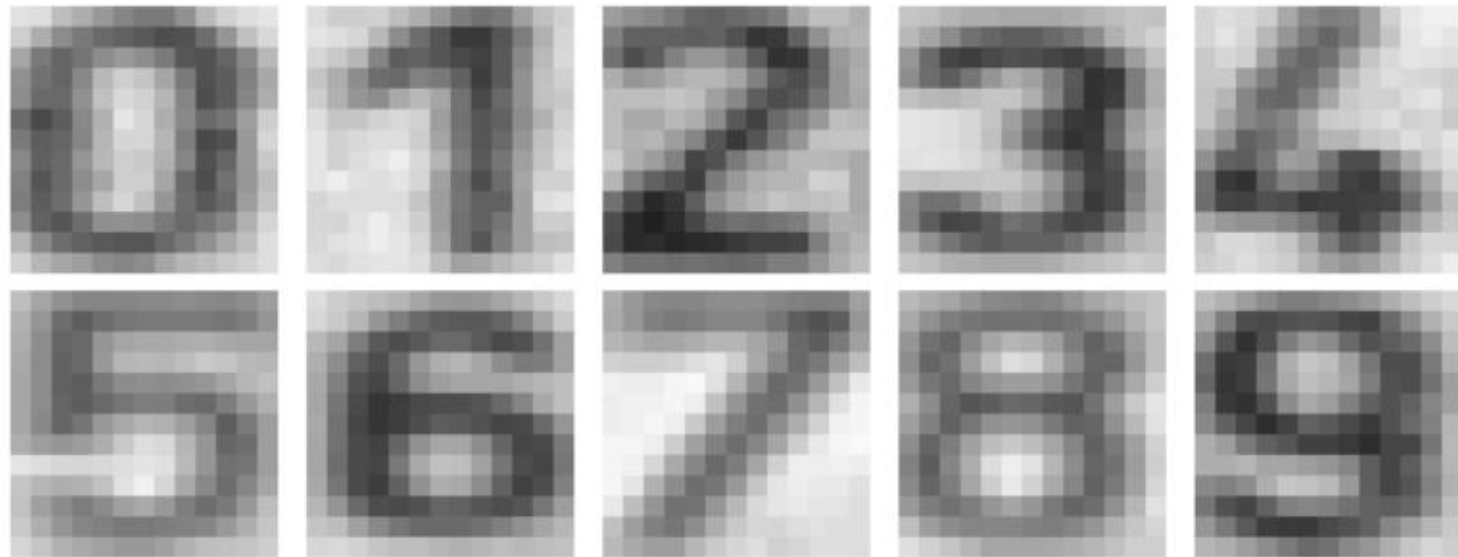
Pretend you see every sample  $k$  extra times:

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{\sum_x [c(x) + k]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{N + k|X|}$$

For conditional, smooth each condition independently

$$P_{\text{LAP}}(x|s) = \frac{c(x, s) + k}{c(s) + k|X|}$$



What is  $|X|$  equal to?

A: 10

B: 2

C: 256

D: None of the above



# Laplace smoothing - as a hyperparameter $k \approx ?$

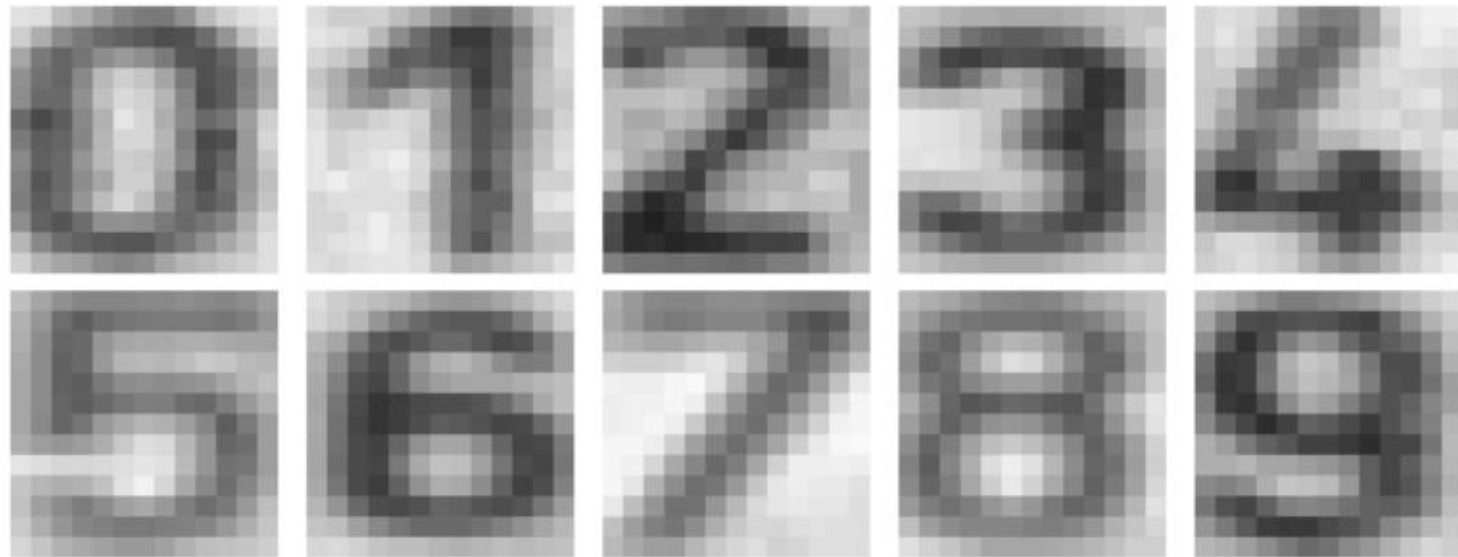
Pretend you see every sample  $k$  extra times:

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{\sum_x [c(x) + k]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{N + k|X|}$$

For conditional, smooth each condition independently

$$P_{\text{LAP}}(x|s) = \frac{c(x, s) + k}{c(s) + k|X|}$$



What is  $|X|$  equal to?

A: 10

B: 2

C: 256

D: None of the above

# Laplace smoothing - as a hyperparameter $k$

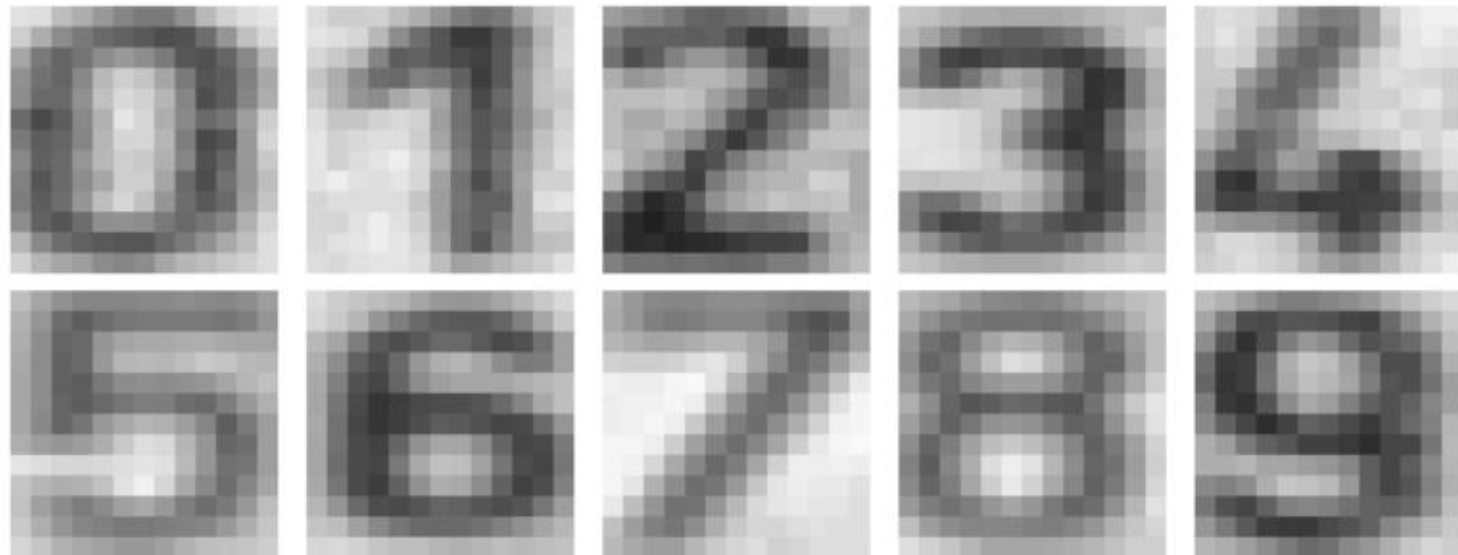
Pretend you see every sample  $k$  extra times:

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{\sum_x [c(x) + k]}$$

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{N + k|X|}$$

For conditional, smooth each condition independently

$$P_{\text{LAP}}(x|s) = \frac{c(x, s) + k}{c(s) + k|X|}$$



What is  $|X|$  equal to?

- A: 10
- B: 2
- C: 256
- D: None of the above

# Laplace smoothing - as a hyperparameter $k$

Pretend you see every sample  $k$  extra times:

$$P_{\text{LAP}}(x) = \frac{c(x) + k}{\sum_x [c(x) + k]}$$

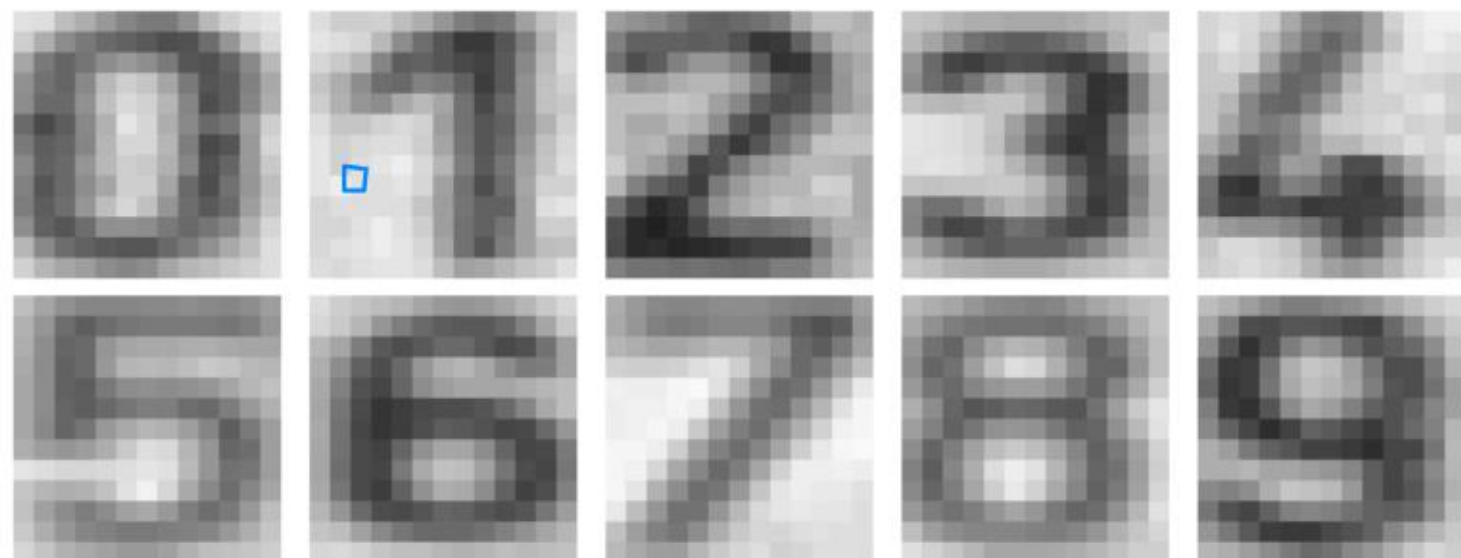
$$P_{\text{LAP}}(x) = \frac{c(x) + k}{N + k|X|}$$

*X = 0, 1, 2, ..., 255*

For conditional, smooth each condition independently

$$P_{\text{LAP}}(x|s) = \frac{c(x, s) + k}{c(s) + k|X|}$$

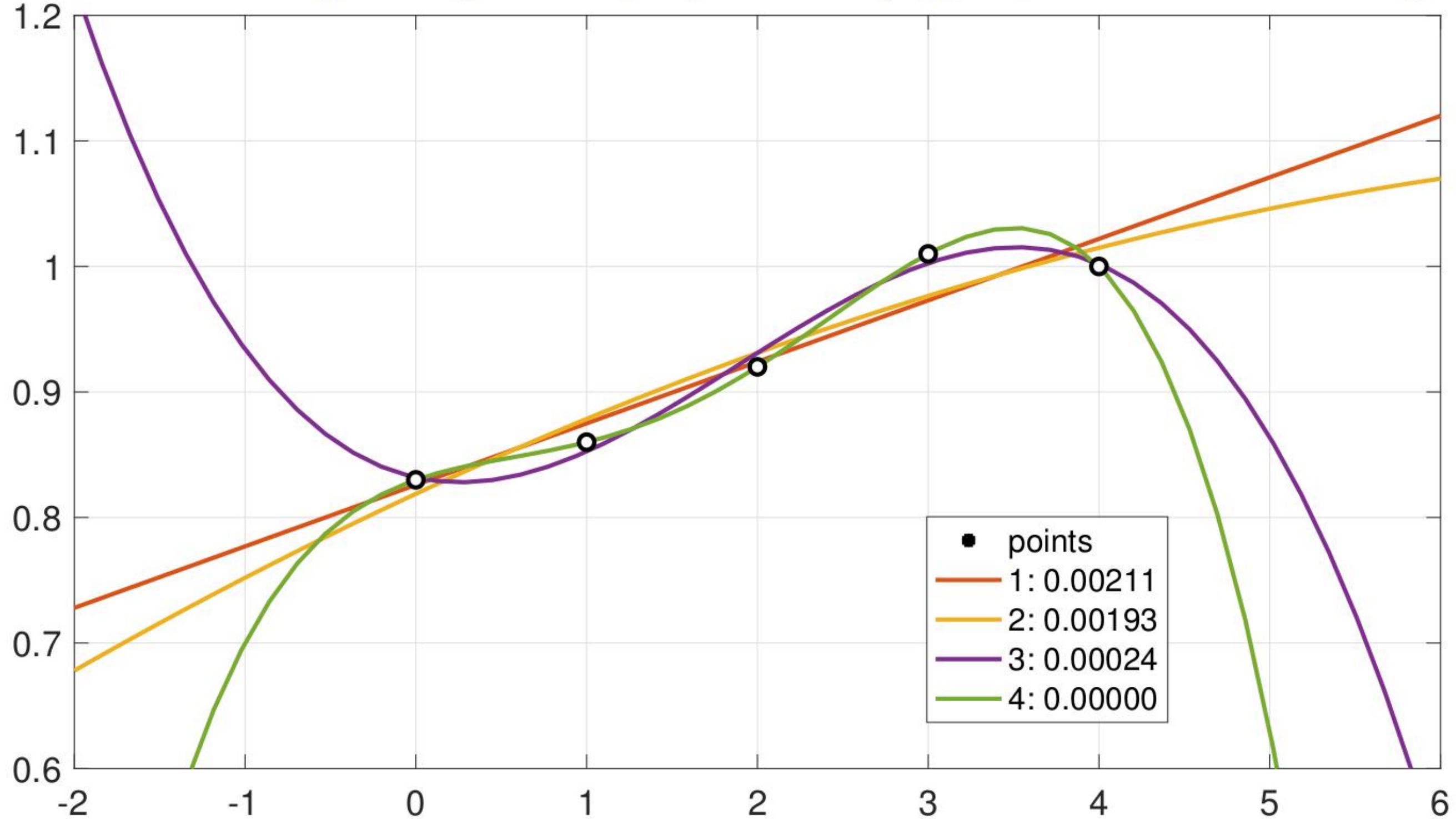
*X = I ∈ {0 ... 255}*



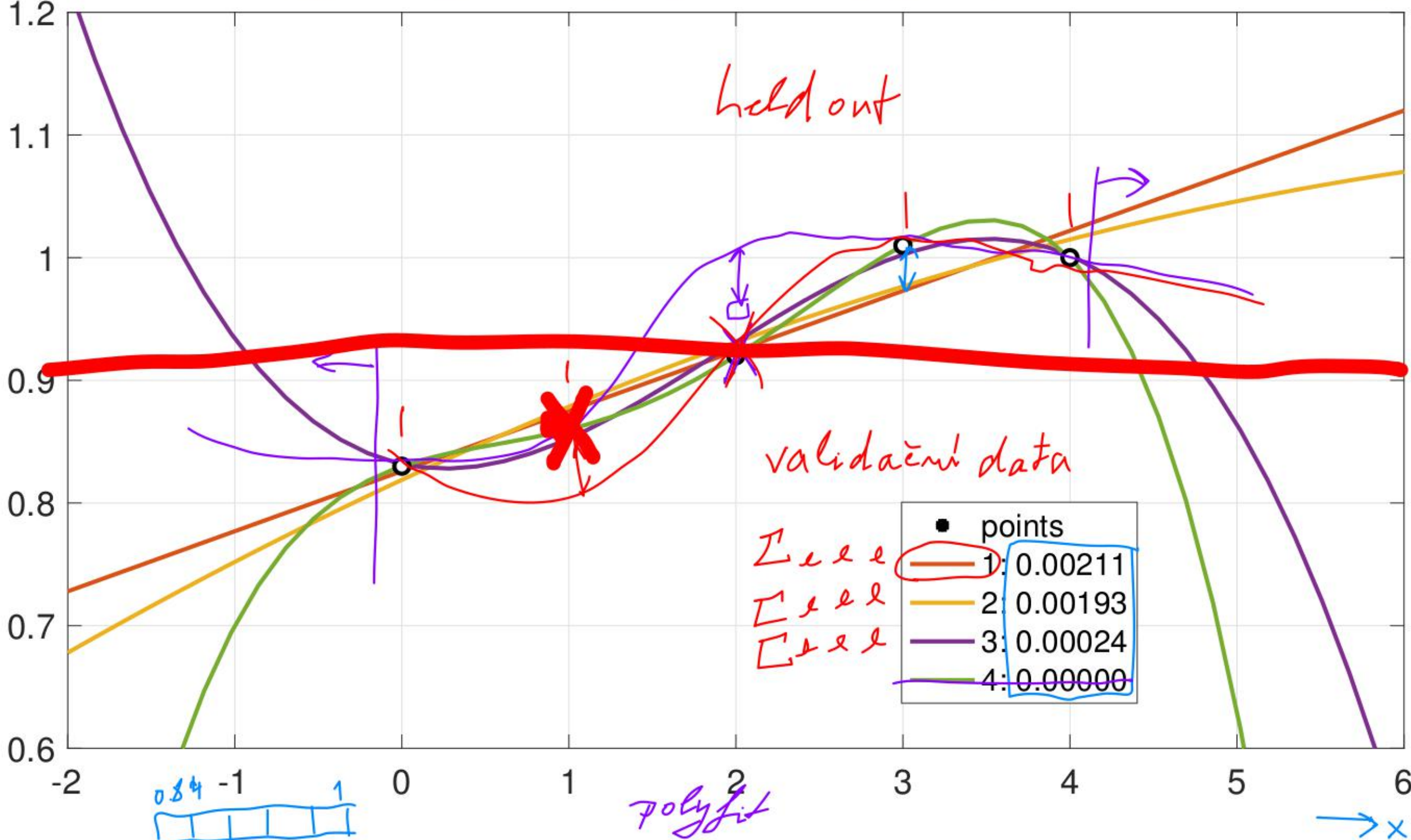
What is  $|X|$  equal to?

- A: 10
  - B: 2
  - C: 256
  - D: None of the above
- |   |    |     |
|---|----|-----|
| A | 17 | 45% |
| B | 2  | 5%  |
| C | 16 | 43% |
| D | 2  | 5%  |

# What is the right degree of polynomial (hyperparameter of a regressor)



# What is the right degree of polynomial (hyperparameter of a regressor)



# Generalization and overfitting

- ▶ Data: training, validating, testing . Wanted classifier performs well on what data?
- ▶ Overfitting: too close to training, poor on testing.

# Training and testing

**Data** labeled instances.

- ▶ Training set
- ▶ Held-out (validation) set
- ▶ Testing set.

**Features** : Attribute-value pairs.

**Learning cycle:**

- ▶ **Learn** parameters (e.g. probabilities) on training set.
- ▶ **Tune** hyperparameters on held-out (validation) set.
- ▶ **Evaluate** performance on testing set.



# Training and testing

**Data** labeled instances.

- ▶ Training set
- ▶ Held-out (validation) set
- ▶ Testing set.

**Features** : Attribute-value pairs.

**Learning cycle:**

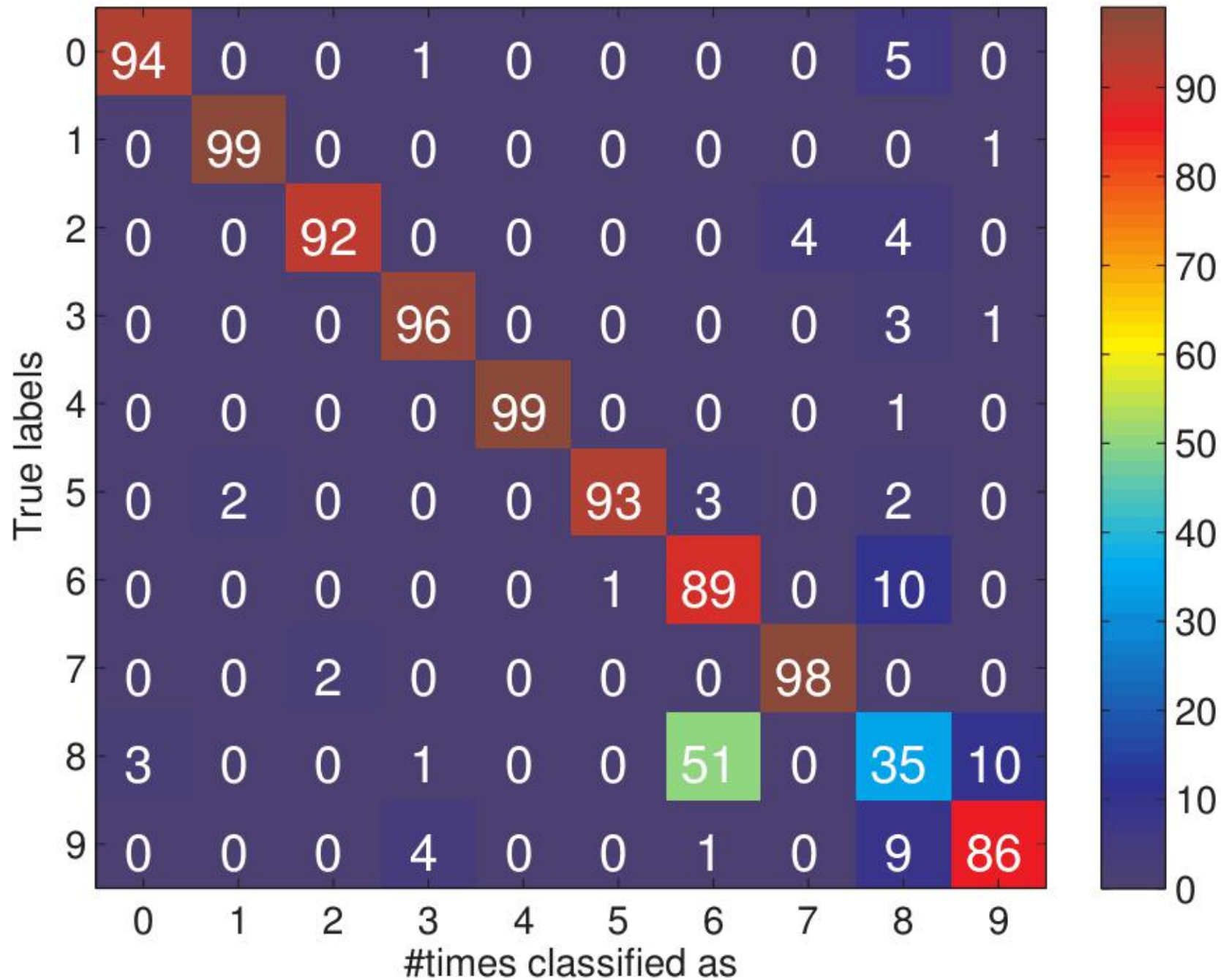
- ▶ **Learn** parameters (e.g. probabilities) on training set.  $1) P(\vec{x}|s) \sim P(x_{00}|s)$
- ▶ **Tune** <sup>meta</sup> hyperparameters on held-out (validation) set.  $k \sim \text{vyhledávání}$
- ▶ **Evaluate** performance on testing set.  $2) I_{thr}$  7





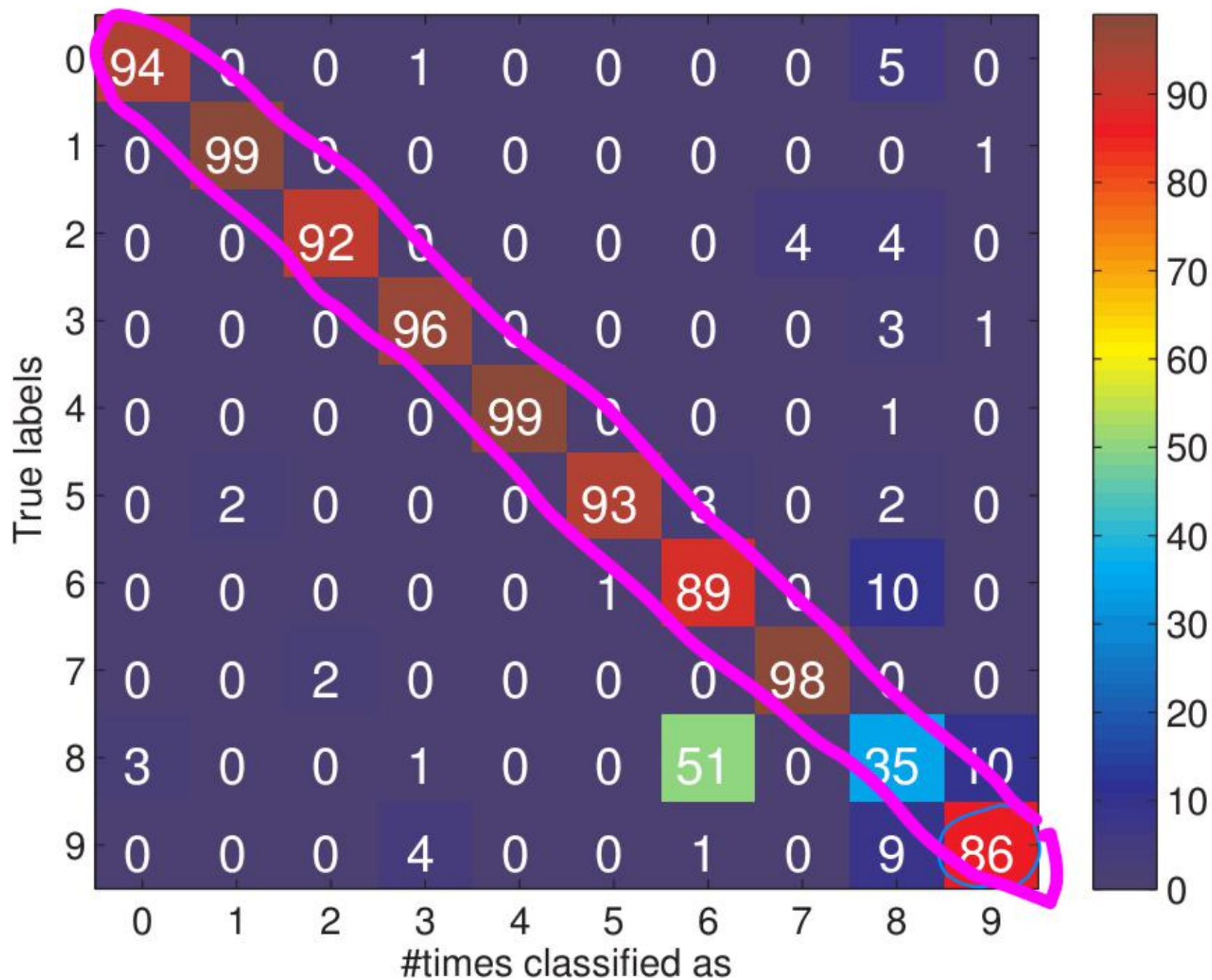
# How to evaluate a classifier? Confusion table

Matching table for test set



# How to evaluate a classifier? Confusion table

Matching table for test set



# Precision and Recall, and ...

Consider digit **detection** (is there a digit?) or SPAM/HAM classification.

**Recall** :

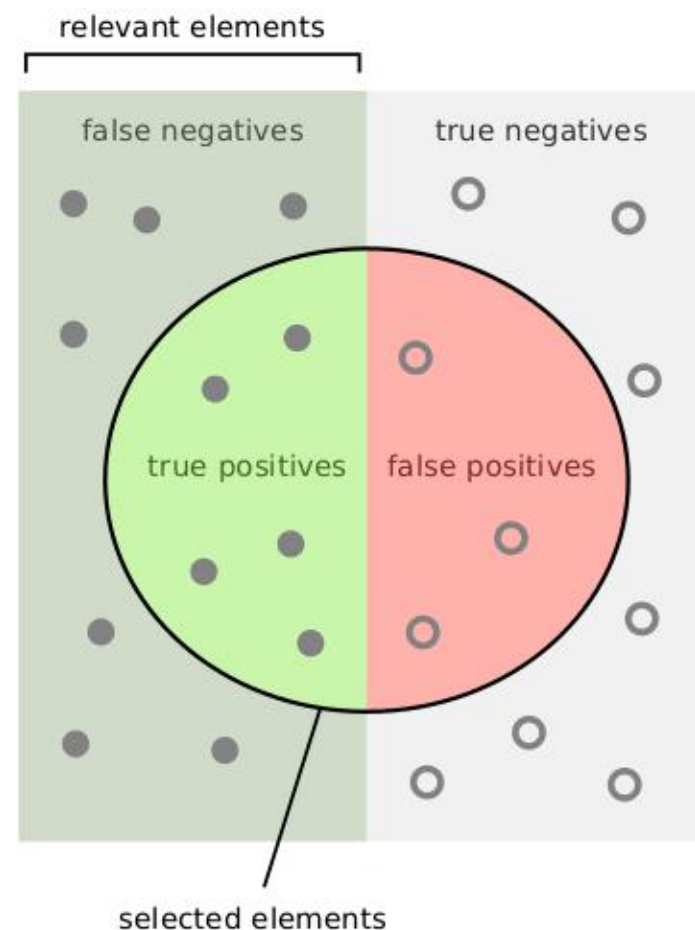
- ▶ How many relevant items are selected?
- ▶ Are we missing some items?
- ▶ Also called: **True positive rate** (TPR), sensitivity, hit rate ...

**Precision**

- ▶ How many selected items are relevant?
- ▶ Also called: Positive predictive value

**False positive rate** (FPR)

- ▶ Probability of false alarm



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

# Precision and Recall, and ...

Consider digit **detection** (is there a digit?) or SPAM/HAM classification.

**Recall** :

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

- ▶ How many relevant items are selected?
- ▶ Are we missing some items?
- ▶ Also called: **True positive rate** (TPR), sensitivity, hit rate ...

**Precision**

$$\frac{TP}{TP+FP}$$

- ▶ How many selected items are relevant?
- ▶ Also called: Positive predictive value

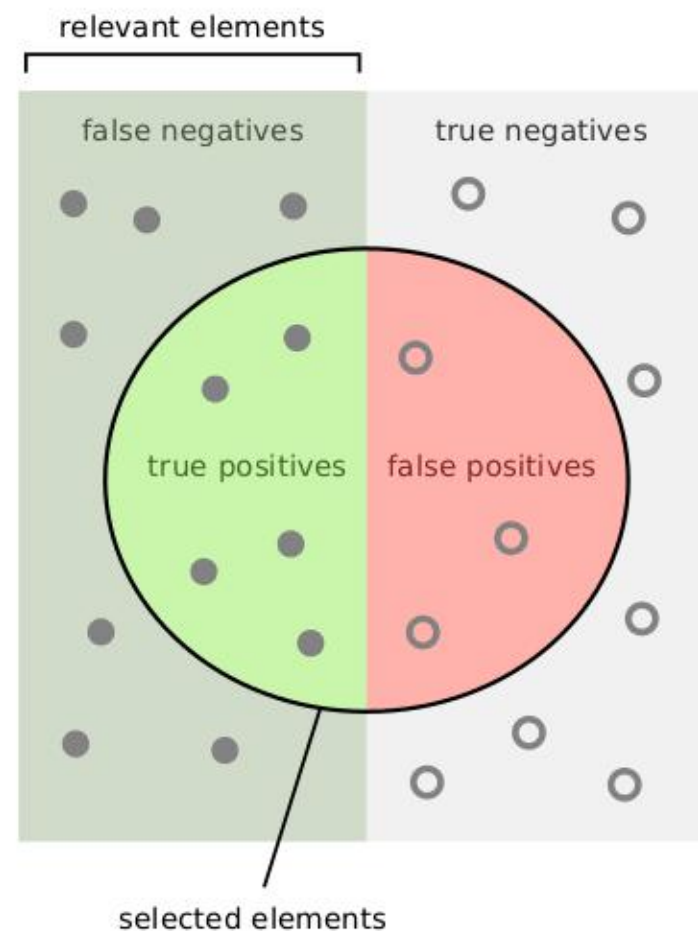
**False positive rate** (FPR)

- ▶ Probability of false alarm

$$FPR = \frac{FP}{FP+TN}$$

classifier

	S	H	
S	TP	FN	P
H	FP	TN	N



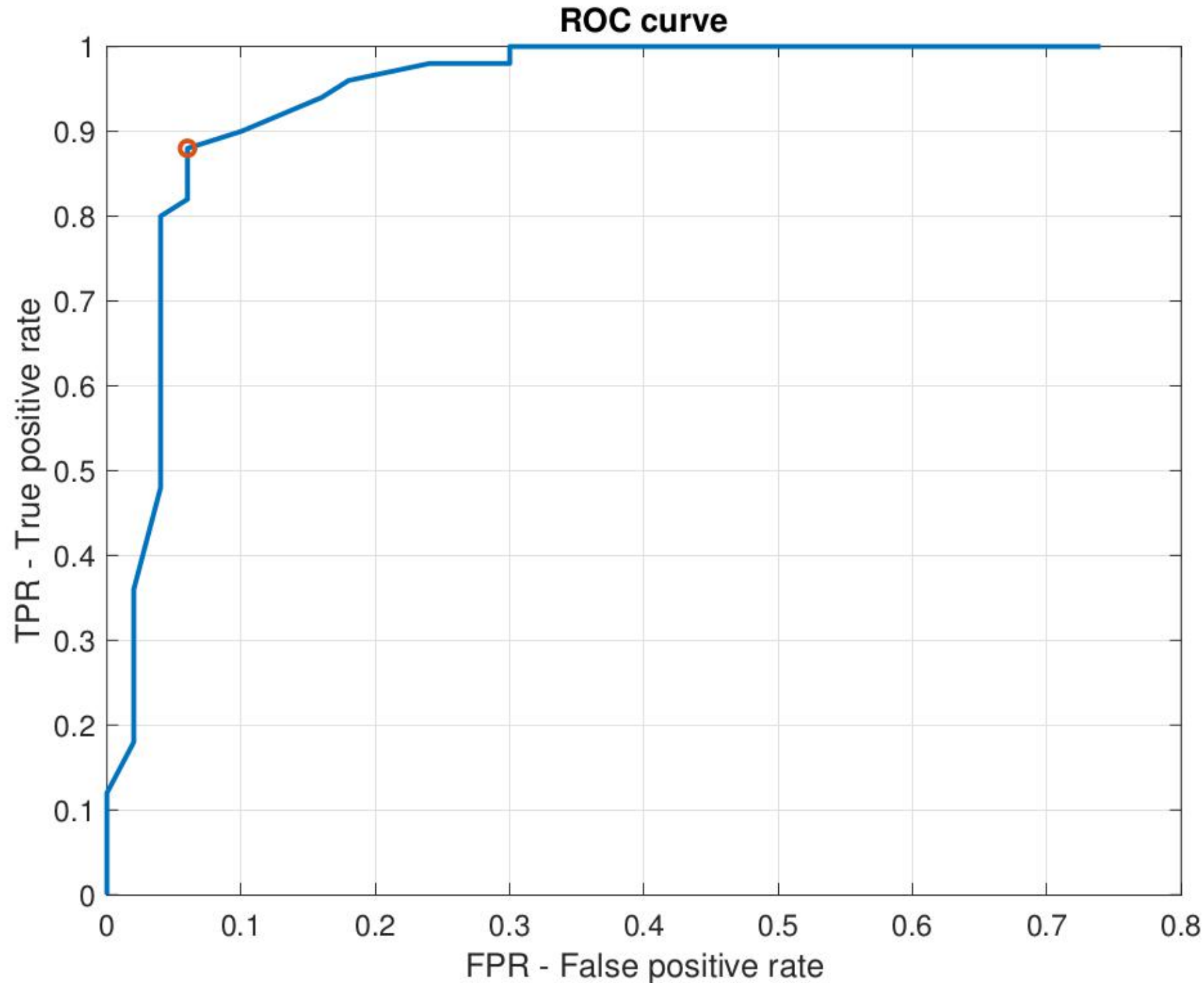
How many selected items are relevant?

$$\text{Precision} = \frac{\text{green}}{\text{green} + \text{red}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{grey}}$$

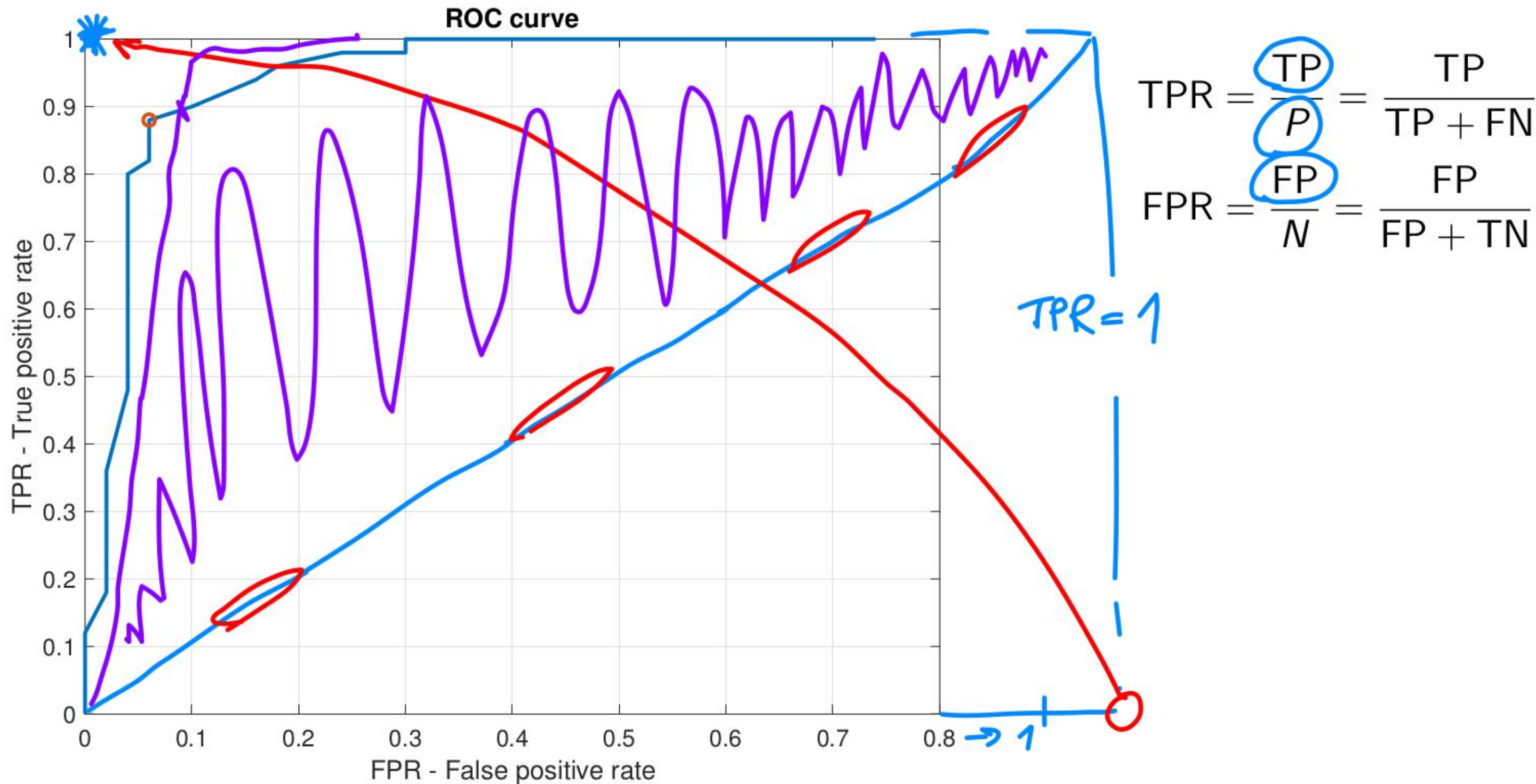
# ROC – Receiver operating characteristics curve



$$\text{TPR} = \frac{\text{TP}}{P} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{N} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

# ROC – Receiver operating characteristics curve



# Discriminant functions $f(\vec{x}, s)$

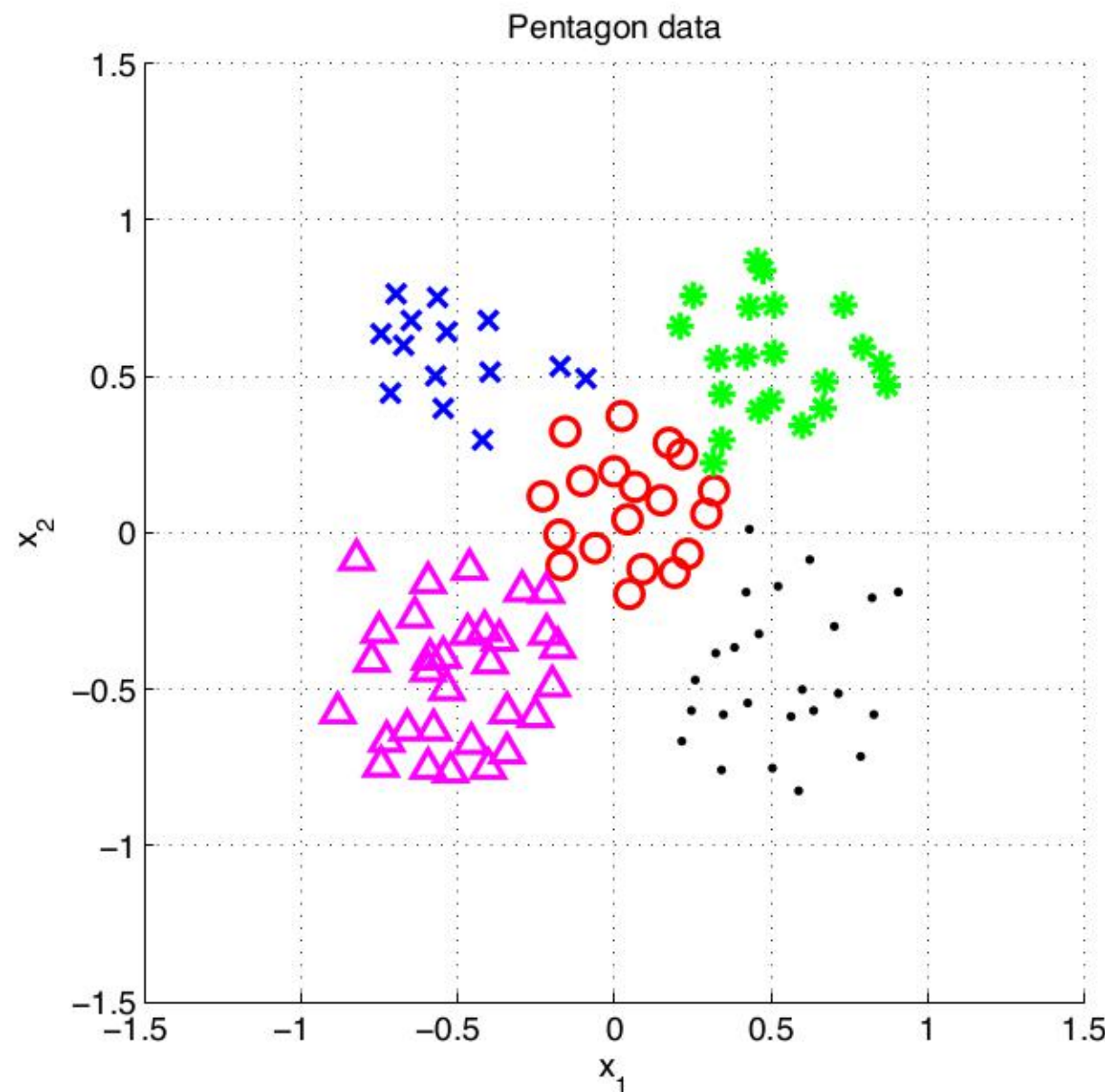
$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} f(\vec{x}, s)$$

Conditional likelihoods:  $\mathcal{N}(\vec{x} | \vec{\mu}_s, \Sigma_s)$

$$\frac{1}{2\pi |\Sigma_s|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_s)^\top \Sigma_s^{-1}(\vec{x} - \vec{\mu}_s)\right\}$$

Bayes:

$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} P(s | \vec{x}) = \frac{P(\vec{x} | s)P(s)}{P(\vec{x})}$$



$$f(\vec{x}, s) = P(s) \frac{1}{2\pi |\Sigma_s|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_s)^\top \Sigma_s^{-1}(\vec{x} - \vec{\mu}_s)\right\}$$

# Discriminant functions $f(\vec{x}, s)$

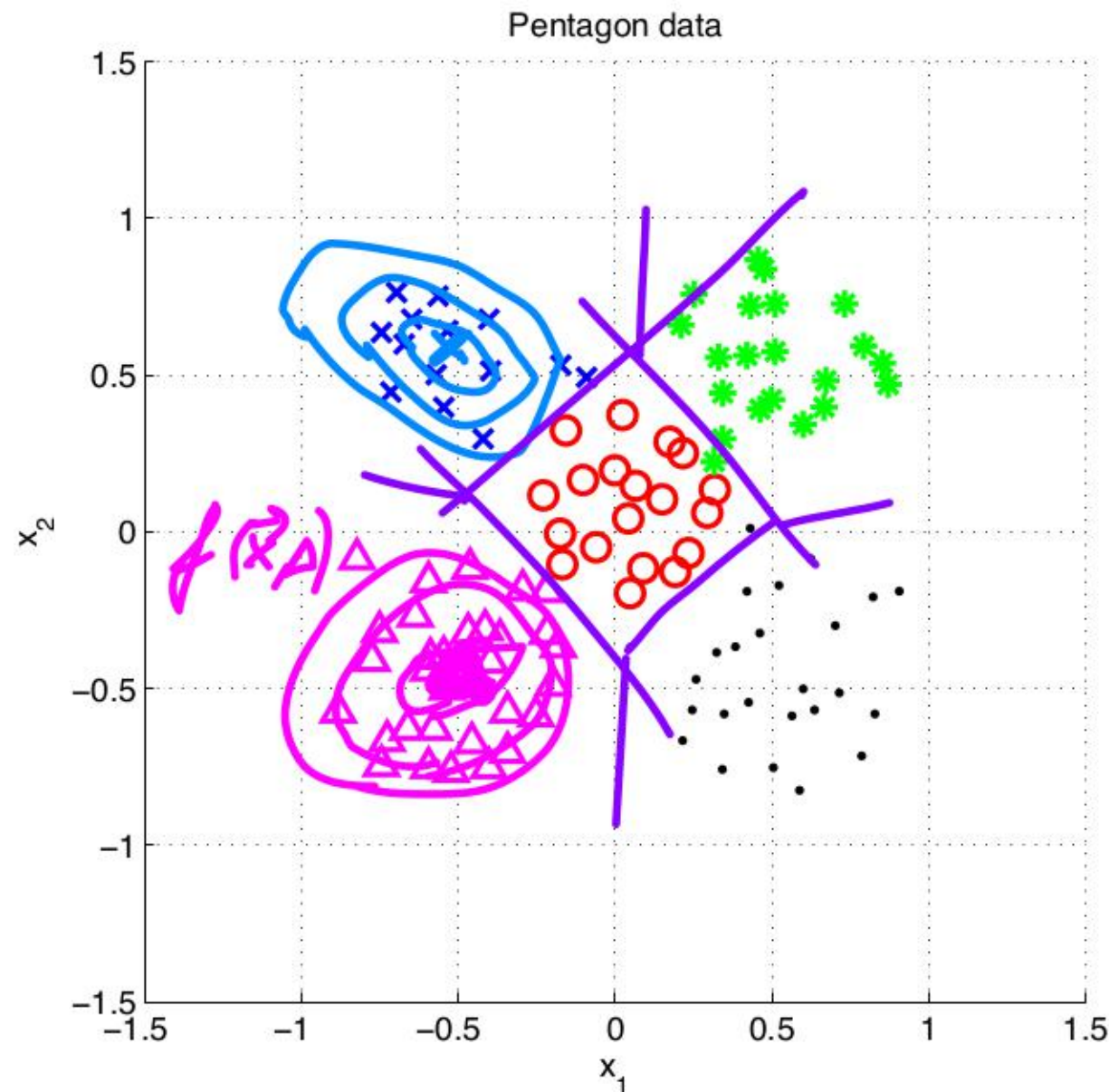
$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} f(\vec{x}, s)$$

Conditional likelihoods:  $\mathcal{N}(\vec{x} | \vec{\mu}_s, \Sigma_s)$

$$p(\vec{x} | s) = \frac{1}{2\pi |\Sigma_s|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_s)^\top \Sigma_s^{-1}(\vec{x} - \vec{\mu}_s)\right\}$$

Bayes:

$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} P(s | \vec{x}) = \frac{P(\vec{x} | s)P(s)}{P(\vec{x})}$$



$$\Sigma_s = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad f(\vec{x}, s) = P(s) \frac{1}{2\pi |\Sigma_s|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_s)^\top \Sigma_s^{-1}(\vec{x} - \vec{\mu}_s)\right\}$$

$-(\vec{x} - \vec{\mu}_s)^\top (\vec{x} - \vec{\mu}_s) = \|\vec{x} - \vec{\mu}_s\|^2$



## Towards linear classifier, geometrical thoughts . . .

$$f(\vec{x}, s) = P(s) \frac{1}{2\pi |\Sigma_s|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_s)^\top \Sigma_s^{-1}(\vec{x} - \vec{\mu}_s)\right\}$$

## Towards linear classifier, geometrical thoughts ...

$$f(\vec{x}, s) = P(s) \frac{1}{2\pi |\Sigma_s|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_s)^T \Sigma_s^{-1}(\vec{x} - \vec{\mu}_s)\right\}$$

$$s(\vec{x}) = \underset{s}{\operatorname{argmin}} \|\vec{x} - \vec{\mu}_s\|$$

$$(\vec{x} - \vec{\mu}_s)^T (\vec{x} - \vec{\mu}_s) = \vec{x}^T \vec{x} - 2\vec{\mu}_s^T \vec{x} + \vec{\mu}_s^T \vec{\mu}_s$$

$$\underset{s}{\operatorname{argmin}} = \vec{x}^T \vec{x} - [2\vec{\mu}_s^T \vec{x} - \vec{\mu}_s^T \vec{\mu}_s]$$

$$\underset{s}{\operatorname{argmax}} = 2\vec{\mu}_s^T \vec{x} - \vec{\mu}_s^T \vec{\mu}_s$$

$$f(\vec{x}, s) = \vec{\mu}_s^T \vec{x} + \frac{1}{2} \vec{\mu}_s^T \vec{\mu}_s$$

? best linear classifier

$$f(\vec{x}, s) = \vec{w}_s^T \vec{x} + w_{b,s} \quad [w_{0s}, w_{1s}, w_{2s}]$$

## Product of many small numbers ...

$$P(s|\vec{x}) = \frac{P(\vec{x}|s)P(s)}{P(\vec{x})} = \frac{P(s)}{P(\vec{x})} P(x[1]|s) \cdot P(x[2]|s) \cdot \dots$$

$P(\vec{x})$  not needed, .....

$$\log(P(x[1]|s)P(x[2]|s)\dots) = \log(P(x[1]|s)) + \log(P(x[2]|s)) + \dots$$

## Product of many small numbers ...

$$P(s|\vec{x}) = \frac{P(\vec{x}|s)P(s)}{P(\vec{x})} = \frac{P(s)}{P(\vec{x})} P(x[1]|s) \cdot P(x[2]|s) \cdot \dots$$

$P(\vec{x})$  not needed, .....

$$\log(P(x[1]|s)P(x[2]|s)\dots) = \log(P(x[1]|s)) + \log(P(x[2]|s)) + \dots$$

## Product of many small numbers ...

$$P(s|\vec{x}) = \frac{P(\vec{x}|s)P(s)}{P(\vec{x})} = \frac{P(s)}{P(\vec{x})} P(x[1]|s) \cdot P(x[2]|s) \cdot \dots$$

$P(\vec{x})$  not needed, .....

$$\log(P(x[1]|s)P(x[2]|s)\dots) = \log(P(x[1]|s)) + \log(P(x[2]|s)) + \dots$$