Prediction & Optimal Predictors

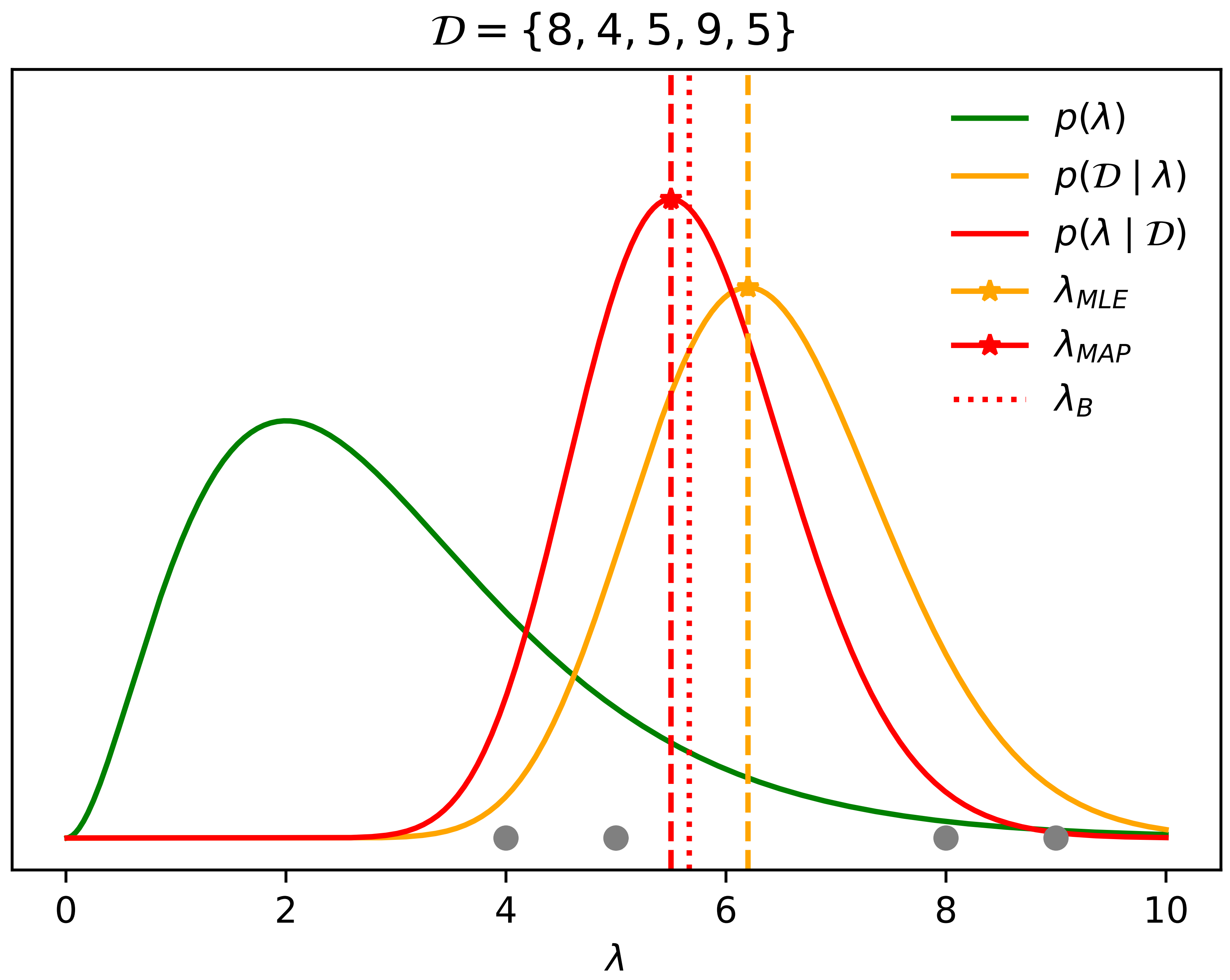
CMPUT 296: Basics of Machine Learning

Textbook §6.1-6.2

Recap: Bayesian Estimation

- **Bayesian estimation:** Estimating models & parameter using the **posterior distribution**
  - **Prior** and **posterior** distributions are over **models**, not over **data**
  - **Conjugate priors** make it possible to perform Bayesian updates **analytically**
    - But many models don't have conjugate priors
- **Point estimates:** MAP, MLE, Bayes estimator
- **Conditional models:** **Predictions** $p(y \mid x)$ can depend on **observations**

$$\overbrace{\boxed{p(f \mid \mathscr{D})}}^{\text{Posterior}} = \frac{\overbrace{\boxed{p(\mathscr{D} \mid f)}}^{\text{Likelihood}}\overbrace{\boxed{p(f)}}^{\text{Prior}}}{\underbrace{\boxed{p(\mathscr{D})}}_{\text{Evidence}}}$$



Outline

1. Recap & Logistics
2. Supervised Prediction
3. Optimal Prediction
4. Irreducible vs. Reducible Error

Types of
Machine Learning Problems

1. *passive* vs. *active* data collection
2. *i.i.d.* vs. *non-i.i.d.*
3. *complete* vs. *incomplete* observations

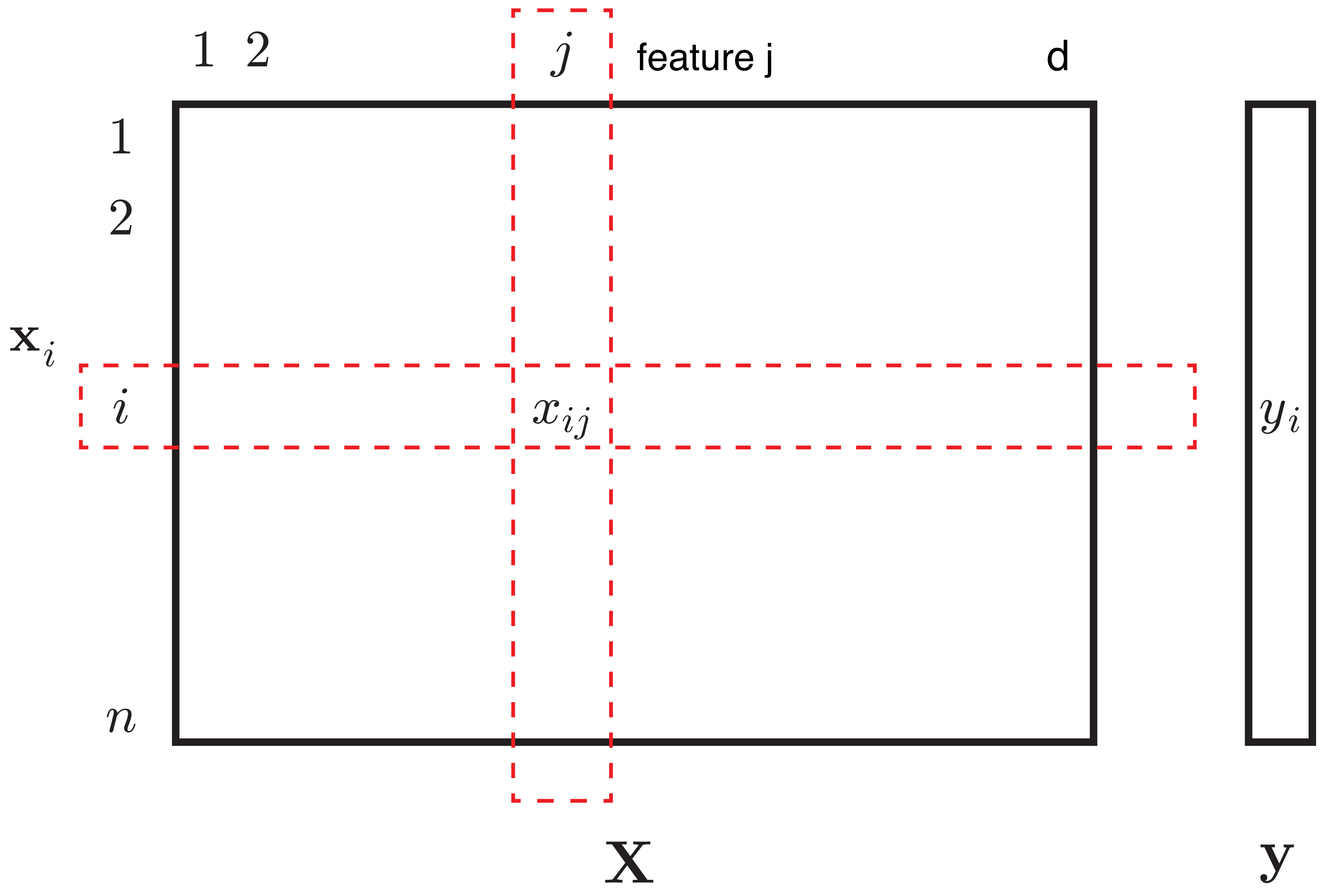
Supervised Prediction

In a supervised prediction problem, we learn a model based on a training dataset of **observations** and their corresponding **targets**, and then use the model to make predictions about new targets based on new observations.

- Dataset: $\mathscr{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
- $\mathbf{x}_i \in \mathscr{X}$ is the i-th **observation** (or input or instance or sample)
- $y_i \in \mathscr{Y}$ is the corresponding **target**
- $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ is a d-dimension vector (i.e., $\mathscr{X} = \mathbb{R}^d$)
- The j-th value of $\mathbf{x}_i$ is the j-th **feature**

Dataset as Matrix

- Typically organize dataset into a $n \times d$ matrix $\mathbf{X}$ and d-vector y
  - One row for each observation
  - One column for each feature



Regression

- A supervised learning problem can typically be classified as either a **regression** problem or a **classification** problem
- **Regression:** Target values are continuous, e.g. $\mathscr{Y} = \mathbb{R}, \mathscr{Y} = [0,\infty)$
- Our house price prediction example is a regression problem; we can extend it to have multiple features:

	size [sqft]	age [yr]	dist [mi]	inc [\$]	dens [ppl/mi^2]	y
$\mathbf{x}_1$	1250	5	2.85	56,650	12.5	2.35
$\mathbf{x}_2$	3200	9	8.21	245,800	3.1	3.95
$\mathbf{x}_3$	825	12	0.34	61,050	112.5	5.10

$\mathbf{X}$ $\quad$ y

Classification

Classification: Predict discrete **class labels**

- Usually not that many labels, e.g. $\mathcal{Y} = \{\text{healthy}, \text{diseased}\}$
- **Multi-label:** A single input may be assigned multiple labels, e.g., categories from $\mathcal{Y} = \{\text{sports}, \text{politics}, \text{travel}, \text{medicine}\}$
- **Multi-class:** Single label per input
  - Multi-class with two labels: **binary classification**
  - E.g., predicting disease state for a patient given weight, height, temperature, sistolic and diatolic blood pressure

Questions

1. What might be an example of a multi-label disease-state classification problem?
2. How could we represent that in the matrix form?

	wt [kg]	ht [m]	T [°C]	sbp [mmHg]	dbp [mmHg]	y
$\mathbf{x}_1$	91	1.85	36.6	121	75	-1
$\mathbf{x}_2$	75	1.80	37.4	128	85	$+1$
$\mathbf{x}_3$	54	1.56	36.6	110	62	-1

Which Formulation to Use?

It's **not always clear-cut** whether to treat a problem as classification or regression.

E.g., output space $\mathcal{Y} = \{0,1,2\}$

- Could be classification with three classes
- Could be regression on $[0,2]$

Question: What considerations would make us choose one category or another?

- Regression functions are often easier to learn (even for classification!)
- If classes have no **order** (e.g., $\{$likes apples, likes bananas, likes oranges$\}$), then regression will be based on faulty assumptions
- If classes *do* have order (e.g., $\{$Good, Better, Best$\}$) then classification will not be able to exploit **that structure**

Optimal Prediction

Suppose we know the true joint distribution $p(\mathbf{x}, y)$, and we want to use it to make predictions in a classification problem.

The **optimal classification predictor** makes the best use of this function.

As with the optimal estimator, we measure the quality of a predictor $f(\mathbf{x})$ by its **expected cost** $\mathbb{E}[C]$. The optimal predictor **minimizes** $\mathbb{E}[C]$.

$$\mathbb{E}[C] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} \operatorname{cost}\big(f(\mathbf{x}), y\big)\, p(\mathbf{x}, y)\, d\mathbf{x},$$

where $\operatorname{cost}(\hat{y}, y)$ is the cost for predicting $\hat{y}$ when the true value is y, and $C = \operatorname{cost}\big(f(X), Y\big)$ is a random variable.

Questions

1. What could we mean by "best"?
2. Why aren't we using MAP or MLE instead of expected cost?

Cost Functions: Classification

- A very common cost function for classification: **0-1 cost**

$$\text{cost}(\hat{y}, y) = \begin{cases} 0 & \text{if } \hat{y} = y, \\ 1 & \text{if } \hat{y} \neq y\,. \end{cases}$$

  - No cost for the right answer; **same cost** for every wrong answer
- **Question:** when might this be inappropriate?
  - Some wrong answers can be much more costly than others
- E.g., in medical domain:
  - **false positive:** leads to an **unnecessary test**
  - **false negative:** leads to an **untreated disease**

$\hat{Y}$ \ Y	-1 (No disease)	1 (Has disease)
-1 (No disease)	0	999
1 (Has disease)	1	0

Multi-class: *Single* label per input
Multi-label: *Set* of labels per input

Optimal Classifier

$$\mathbb{E}[C] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} \text{cost}\left(f(\mathbf{x}), y\right) p(\mathbf{x}, y)\, d\mathbf{x}$$

- Can't actually achieve zero cost when doing **multi-class** classification
  - $f(\mathbf{x})$ has to output a **single label** for observation $\mathbf{x}$
  - But there might be instances with the **same observations** but **different labels**
    - i.e., in general $\forall \mathbf{x} : p(y \mid \mathbf{x}) \neq 1$
- **Question:** Is this also true for **multi-label** classification?

Deriving Optimal Classifier

$$\mathbb{E}[C] = \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} \text{cost}\left(f(\mathbf{x}), y\right) p(\mathbf{x}, y)\, d\mathbf{x}$$

$$= \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} \text{cost}\left(f(\mathbf{x}), y\right) p(y \mid \mathbf{x}) p(\mathbf{x})\, d\mathbf{x}$$

$$= \int_{\mathcal{X}} p(\mathbf{x}) \underbrace{\boxed{\sum_{y \in \mathcal{Y}} \text{cost}\left(f(\mathbf{x}), y\right) p(y \mid \mathbf{x})}}_{\mathbb{E}[C \mid X = \mathbf{x}]} d\mathbf{x}$$

$$= \int_{\mathcal{X}} p(\mathbf{x}) \mathbb{E}[C \mid X = \mathbf{x}]\, d\mathbf{x}$$

- We can minimize

$$\mathbb{E}[C \mid X = \mathbf{x}] = \sum_{y \in \mathcal{Y}} \text{cost}\left(f(\mathbf{x}), y\right) p(y \mid \mathbf{x})$$

  separately for each $\mathbf{x}$ (**why?**)

- *Proof:* Suppose $f^\dagger(\mathbf{x})$ is not optimal for a specific value $\mathbf{x}_0$
- Then let

$$f^*(\mathbf{x}) = \begin{cases} f^\dagger(\mathbf{x}) & \text{if } \mathbf{x} \neq \mathbf{x}_0, \\ \arg\min_{\hat{y} \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} \text{cost}(\hat{y}, y) p(y \mid \mathbf{x}_0) & \text{if } \mathbf{x} = \mathbf{x}_0\,. \end{cases}$$

- f^* has lower expected cost at $\mathbf{x}_0$ and same expected cost at all other $\mathbf{x}$

Deriving Optimal Classifier for 0-1 Cost

$$f^*(\mathbf{x}) = \arg\min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} \text{cost}(\hat{y}, y)p(y \mid \mathbf{x}) \quad = \arg\min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} \text{cost}(\hat{y}, y)p(y \mid \mathbf{x}) - 1$$

$$= \arg\max_{\hat{y}\in\mathcal{Y}} 1 - \sum_{y\in\mathcal{Y}} \text{cost}(\hat{y}, y)p(y \mid \mathbf{x})$$

$$= \arg\max_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} \left(1 - \text{cost}(\hat{y}, y)\right) p(y \mid \mathbf{x})$$

$$= \arg\max_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}, y\neq\hat{y}} 0 \cdot p(y \mid \mathbf{x}) + \sum_{y\in\mathcal{Y}, y=\hat{y}} 1 \cdot p(y \mid \mathbf{x})$$

$$= \arg\max_{\hat{y}\in\mathcal{Y}} p(y \mid \mathbf{x}) \quad \blacksquare$$

This is the **Bayes risk classifier**