Prediction & Optimal Predictors

CMPUT 267: Basics of Machine Learning

Textbook §6.1-6.2

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

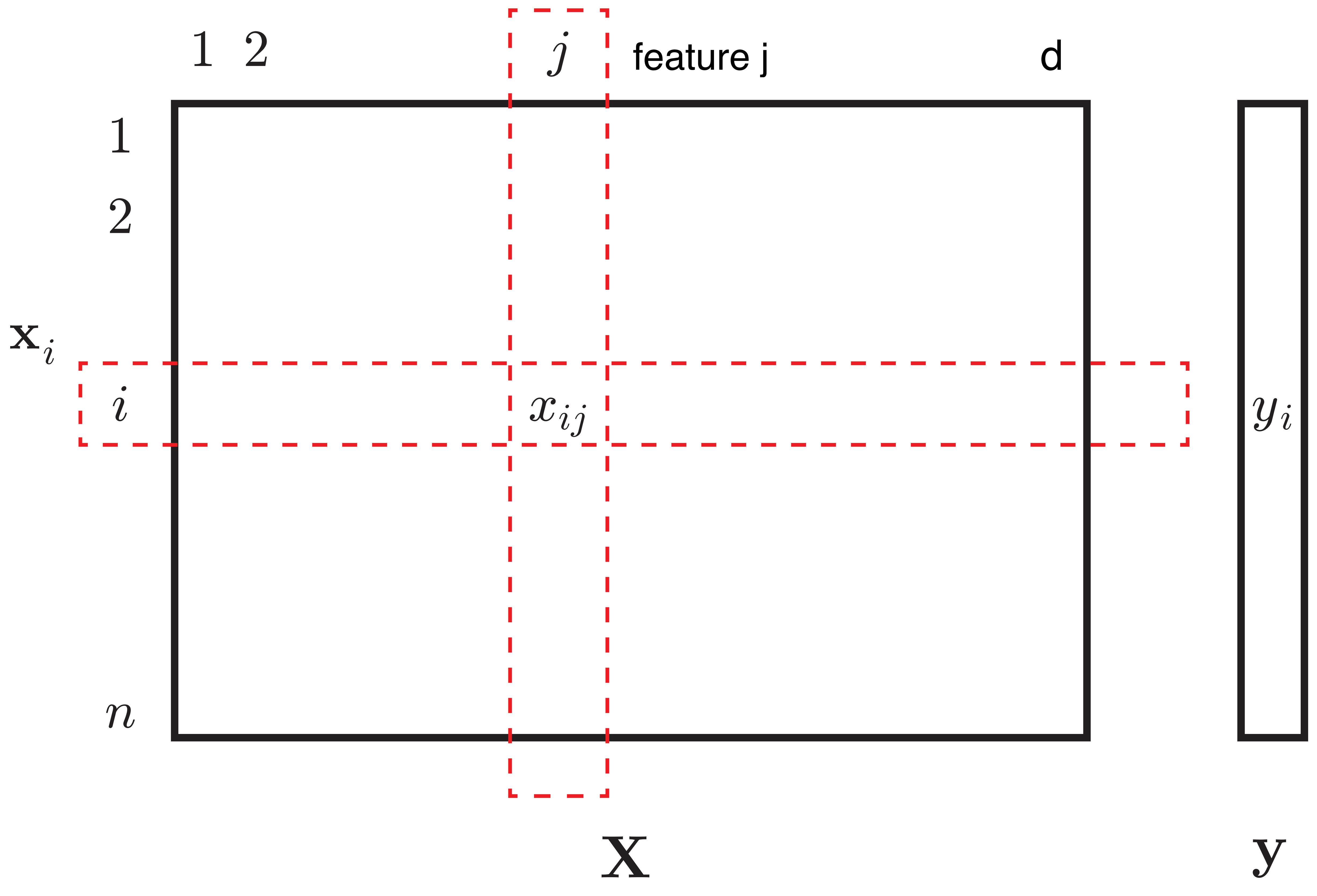
Supervised Prediction

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

Dataset as Matrix (2d array)

- 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

$\mathbf{X}$ y

Another regression example

- x = [house size, temperature outside, temperature inside]
  - d = 3, three-dimensional input vector (array)
- y = gas usage for the day (real-valued)

Classification

Classification: Predict discrete **class labels**

- Usually not that many labels, e.g. $\mathscr{Y} = \{\text{healthy}, \text{diseased}\}$
- **Multi-label:** A single input may be assigned multiple labels, e.g., categories from $\mathscr{Y} = \{\text{sports}, \text{politics}, \text{travel}, \text{medicine}\}$
- **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

	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

Another classification example

- x = [house size, temperature outside, temperature inside]
  - d = 3, three-dimensional input vector (array)
- for regression we had y = gas usage for the day (real-valued)
- for classification, we might have $y \in \{\text{Low}, \text{Med}, \text{High}\}$

Multi-label vs Multi-class

- We can always turn a multi-label problem into a multi-class one
  - **multi-label** with $\mathscr{Y} = \{1,2,3\}$ is the same as **multi-class** with classes $\mathscr{Y} = \big\{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\big\}$
  - but this multi-class problem **scales really poorly** with more labels, so we very rarely use this approach (e.g., how many classes from 10 labels?)
- The **simplest solution** is to treat each label as a binary prediction problem
  - independently output $f_1(\mathbf{x}) = 0$ or 1 for label 1, $f_2(\mathbf{x}) = 0$ or 1 for label 2, …
  - or learn $p(y_1 = 1 \mid \mathbf{x})$ for label 1, $p(y_2 = 1 \mid \mathbf{x})$ for label 2, …,

Multi-label vs Multi-class

- We can always turn a multi-label problem into a multi-class one
- The **simplest solution** is to treat each label as a binary prediction problem
  - learn $p(y_1 = 1 \mid \mathbf{x})$ for label 1, $p(y_2 = 1 \mid \mathbf{x})$ for label 2, …,
- **Smarter strategies** look at **relationships** between labels
  - $p(y_1, y_2 \mid \mathbf{x}) \neq p(y_1 \mid \mathbf{x})p(y_2 \mid \mathbf{x})$
- For this course, we focus on the **simplest approaches**. Therefore, we will focus on **binary classification**
  - which provides at least a basic solution for the multi-label problem

Which Formulation to Use?

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

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

- 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** $\mathbb{E}[C]$.

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

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

Questions

1. 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

"Optimal" Classifier is Not Always Right

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

- 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 $\forall \mathbf{x} : p(y \mid \mathbf{x}) \neq 1$

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

$$
\begin{aligned}
f^*(\mathbf{x}) &= \arg\min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} \text{cost}(\hat{y}, y)p(y \mid \mathbf{x}) \quad = \arg\min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} \text{cost}(\hat{y}, y)p(y \mid \mathbf{x}) - 1 \\
&= \arg\max_{\hat{y}\in\mathcal{Y}} 1 - \sum_{y\in\mathcal{Y}} \text{cost}(\hat{y}, y)p(y \mid \mathbf{x}) \\
&= \arg\max_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} \left(1 - \text{cost}(\hat{y}, y)\right) p(y \mid \mathbf{x}) \\
&= \arg\max_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}, y\neq\hat{y}} 0 \cdot p(y \mid \mathbf{x}) + \sum_{y\in\mathcal{Y}, y=\hat{y}} 1 \cdot p(y \mid \mathbf{x}) \\
&= \arg\max_{\hat{y}\in\mathcal{Y}} p(y \mid \mathbf{x}) \quad \blacksquare
\end{aligned}
$$

Cost Functions: Regression

- Two most common cost functions for regression:
    1. **Squared error:** $\mathrm{cost}(\hat{y}, y) = (\hat{y} - y)^2$
    2. **Absolute error:** $\mathrm{cost}(\hat{y}, y) = \left|\hat{y} - y\right|$
- Squared error penalizes **large errors** more heavily than absolute error
- Other possibilities that depend on the size of the target
    - E.g., **percentage error:** $\mathrm{cost}(\hat{y}, y) = \dfrac{\left|\hat{y} - y\right|}{|y|}$

Deriving Optimal Regressor for Squared Error

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

$$= \int_{\mathcal{X}} \int_{\mathcal{Y}} \left(f(\mathbf{x}) - y\right)^2 p(\mathbf{x}, y)\, dy\, d\mathbf{x}$$

$$= \int_{\mathcal{X}} p(\mathbf{x}) \underbrace{\boxed{\int_{\mathcal{Y}} \left(f(\mathbf{x}) - y\right)^2 p(y \mid \mathbf{x})\, dy}}_{\mathbb{E}[C \mid X = \mathbf{x}]} d\mathbf{x}$$

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

- Once again, we can directly optimize $\mathbb{E}[C \mid X = \mathbf{x}]$:

$$f^*(\mathbf{x}) = \arg\min_{\hat{y} \in \mathcal{Y}} g(\hat{y})$$

where

$$g(\hat{y}) = \int_{\mathcal{Y}} \left(\hat{y} - y\right)^2 p(y \mid \mathbf{x})\, dy$$

Deriving Optimal Regressor for Squared Error, cont.

$$g(\hat{y}) = \int_{\mathcal{Y}} (\hat{y} - y)^2 \, p(y \mid \mathbf{x}) \, dy$$

$$\frac{\partial g(\hat{y})}{\partial \hat{y}} = 2 \int_{\mathcal{Y}} (\hat{y} - y) \, p(y \mid \mathbf{x}) \, dy = 0$$

$$\iff \int_{\mathcal{Y}} \hat{y} p(y \mid \mathbf{x}) \, dy = \int_{\mathcal{Y}} y p(y \mid \mathbf{x}) \, dy$$

$$\iff \hat{y} \boxed{\underbrace{\int_{\mathcal{Y}} p(y \mid \mathbf{x}) \, dy}_{= 1}} = \int_{\mathcal{Y}} y p(y \mid \mathbf{x}) \, dy$$

$$\iff \hat{y} = \boxed{\int_{\mathcal{Y}} y p(y \mid \mathbf{x}) \, dy} = \mathbb{E}[Y \mid X = \mathbf{x}]$$

So,

$$f^*(\mathbf{x}) = \arg\min_{\hat{y} \in \mathcal{Y}} g(\hat{y})$$

$$= \mathbb{E}[Y \mid X = \mathbf{x}] \quad \blacksquare$$

Irreducible Error

What is our **expected squared error** when we use the **optimal** predictor?

$f^*(\mathbf{x}) = \mathbb{E}[Y \mid X = \mathbf{x}]$, so

$$\mathbb{E}[C] = \int_{\mathcal{X}} p(\mathbf{x}) \int_{\mathcal{Y}} \left(f^*(\mathbf{x}) - y\right)^2 p(y \mid X = \mathbf{x})\, dy\, d\mathbf{x}$$

$$= \int_{\mathcal{X}} p(\mathbf{x}) \boxed{\int_{\mathcal{Y}} \left(\mathbb{E}[Y \mid X = \mathbf{x}] - y\right)^2 p(y \mid X = \mathbf{x})\, dy}\, d\mathbf{x}$$

$$= \int_{\mathcal{X}} p(\mathbf{x}) \mathrm{Var}[Y \mid X = \mathbf{x}]\, d\mathbf{x}$$

Error for any predictor f

What is our **expected squared error** when we use a **suboptimal** predictor?

$$\mathbb{E}[C \mid X] = \mathbb{E}\left[\left(f(\mathbf{x}) - Y\right)^2 \middle| X = \mathbf{x}\right] = \mathbb{E}\left[\left(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}] + \mathbb{E}[Y \mid X = \mathbf{x}] - Y\right)^2 \middle| X = \mathbf{x}\right]$$

$$= \mathbb{E}\left[\left(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}]\right)^2 + 2\underbrace{\boxed{\left(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}]\right)\left(\mathbb{E}[Y \mid X = \mathbf{x}] - Y\right)}}_{=0}\right.$$

$$\left. + \left(\mathbb{E}[Y \mid X = \mathbf{x}] - Y\right)^2 \middle| X = \mathbf{x}\right]$$

We'll take expectation again at the end to get to $\mathbb{E}[C] = \mathbb{E}[\mathbb{E}[C \mid X]]$

Middle Term is 0

$$\mathbb{E}\left[\boxed{\big(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}]\big)\big(\mathbb{E}[Y \mid X = \mathbf{x}] - Y\big)}\,\Big|\, X = \mathbf{x}\right]$$

$$= \big(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}]\big)\,\mathbb{E}\left[\big(\mathbb{E}[Y \mid X = \mathbf{x}] - Y\big)\,\Big|\, X = \mathbf{x}\right]$$

$$= \big(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}]\big)\big(\mathbb{E}[Y \mid X = \mathbf{x}] - \mathbb{E}[Y|X = \mathbf{x}]\big)$$

$$= \big(f(\mathbf{x}) - \mathbb{E}[Y \mid X = \mathbf{x}]\big)\,0$$

$$= 0$$

Expected Cost for f

What is our **expected squared error** when we use a **suboptimal** predictor?

$$= 0$$

$$\mathbb{E}\left[\mathbb{E}[C \mid X]\right] = \mathbb{E}\left[\left(f(X) - \mathbb{E}[Y \mid X]\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}[Y \mid X] - Y\right)^2\right]$$

$$\mathbb{E}[C] = \underbrace{\boxed{\mathbb{E}\left[\left(f(X) - f^*(X)\right)^2\right]}}_{\text{Reducible error}} + \underbrace{\boxed{\mathbb{E}\left[\left(f^*(X) - Y\right)^2\right]}}_{\text{Irreducible error}}$$

How do we reduce reducible error?

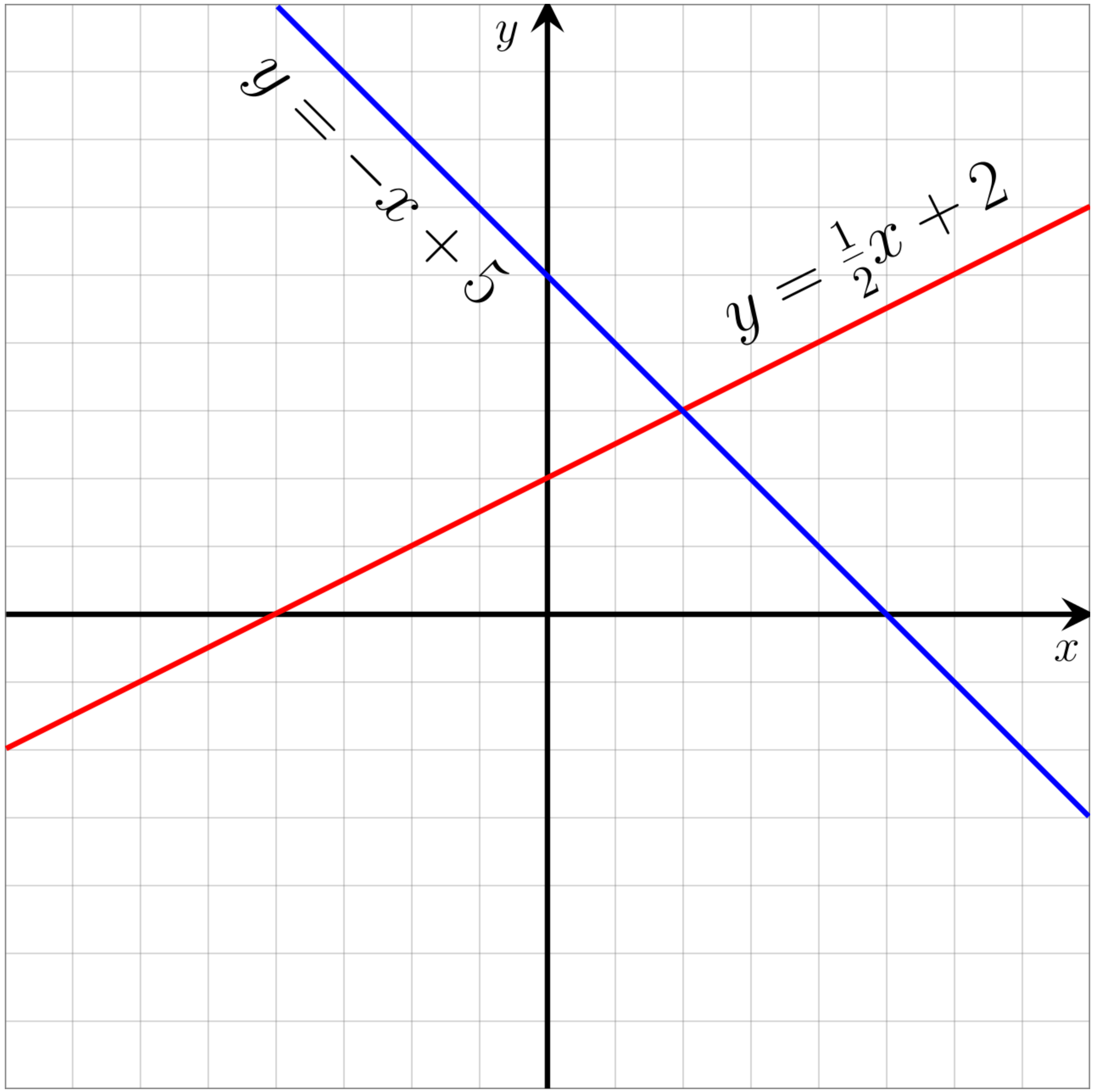
- i.e., how do we make the difference between f and f* smaller
- Imagine you learn f from a batch of n samples
- Further, let’s imagine you decide to learn a linear function

Linear vs Nonlinear Functions

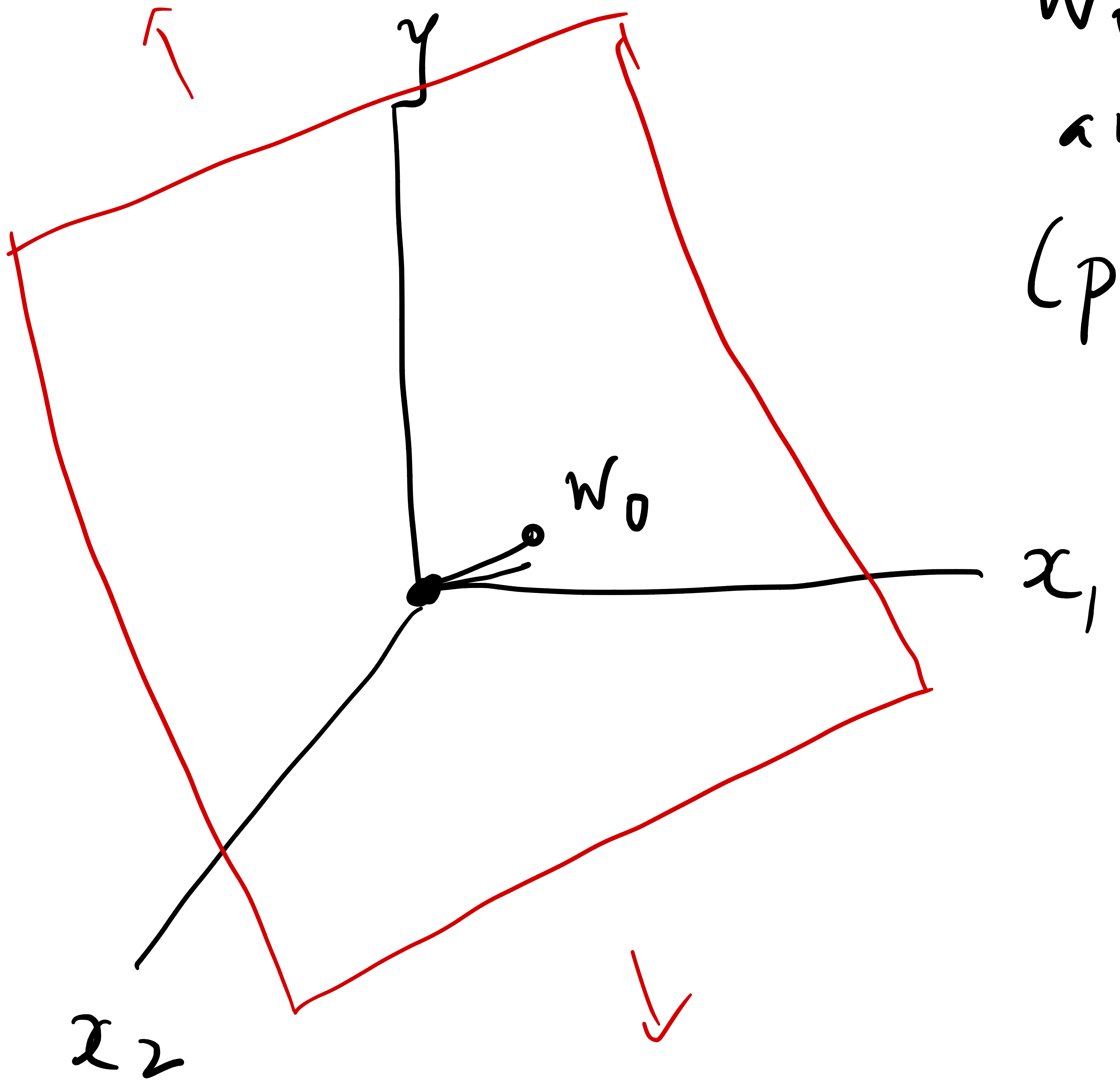
- Linear functions: functions that weight features and add them
  - e.g., $f(x) = w_0 + w_1 x_1 + w_2 x_2$
- Nonlinear functions: any functions that are not linear

Linear functions (1d)

- $f(x) = w_0 + w_1 x_1$.
  What is w_1 and w_0?



Linear functions (2d)



w_0 shifts plane away from origin (positive here)

$$f(x) = w_0 + w_1 x_1 + w_2 x_2$$

How do we reduce reducible error?

- i.e., how do we make the difference between f and f* smaller
- Imagine you learn f from a batch of n samples
- Further, let's imagine you decide to learn a linear function
- What are the sources of inaccuracy? $\mathbb{E}\left[\left(f(X) - f^*(X)\right)^2\right]$

How do we reduce reducible error?

- Imagine you learn f from a batch of n samples
- Further, let's imagine you decide to learn a linear function
- What are the sources of inaccuracy? $\mathbb{E}\left[\left(f(X) - f^*(X)\right)^2\right]$
- **Source 1: limited hypothesis space.** f is a linear function, f* might be a nonlinear function
- **Source 2: optimization was insufficient.** Maybe we used gradient descent, and didn't fully optimize f (stopped too early)
- **Source 3: limited data.** Not enough samples to identify a good f

How do we reduce reducible error?

- **Source 1: limited hypothesis space.** f is a linear function, f* might be a nonlinear function
    - Solution: make the hypothesis space bigger (e.g., learn polynomials)
- **Source 2: optimization was insufficient.** Maybe we used gradient descent, and didn’t fully optimize f (stopped too early)
    - Solution: more carefully ensure you get to a stationary point
- **Source 3: limited data.** Not enough samples to identify a good f
    - Solution: gather more data

Can we reduce **irreducible** error?

- It's called irreducible for a reason…
- It is the variance of Y given X: Var$(Y \mid X = x)$
- Improving our learned function f **cannot** change the inherent variance in Y
- But, can you think of a way to reduce the variance of Y conditioned on our inputs? What is the source of variance in Y given x?
  - hint: think about the gumball machine example from earlier
  - hint: think about why gas usage was variable, conditioned on house size, temperature outdoors and desired indoor temperature

Summary

- **Supervised learning problem:** Learn a **predictor** $f : \mathcal{X} \to \mathcal{Y}$ from a dataset $\mathcal{D} = \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{n}$
  - $\mathcal{X}$ is the set of **observations**, and $\mathcal{Y}$ is the set of **targets**
- **Classification** problems have discrete targets
- **Regression** problems have continuous targets
- Predictor performance is measured by the **expected** $\text{cost}(\hat{y}, y)$ of predicting $\hat{y}$ when the true value is y
- An **optimal predictor** for a given distribution **minimizes** the expected cost
- Even an optimal predictor has some **irreducible error**.
  Suboptimal predictors have additional, **reducible error**