Machine Learning I: Supervised Methods

B. Keith Jenkins

Announcements

- TA and instructor office hours are posted on piazza
- Homework 1 will be posted Friday 1/19
  - Due on Friday 1/26, 11:59 PM (PT)
- Discussion Session 2 this Friday
- NEW: Python instruction sessions
  - For students learning Python
  - Fridays 12:00 PM - 12:50 PM
  - OHE 120 and den broadcast
  - For 3 weeks (3 sessions total)
  - Starts this Friday (1/19)
  - Video will be posted on D2L

Reading

- Bishop 5.2.4 (Gradient descent)
  - Note: Bishop's E is our J
  - = criterion function for optimization

Today's Lecture

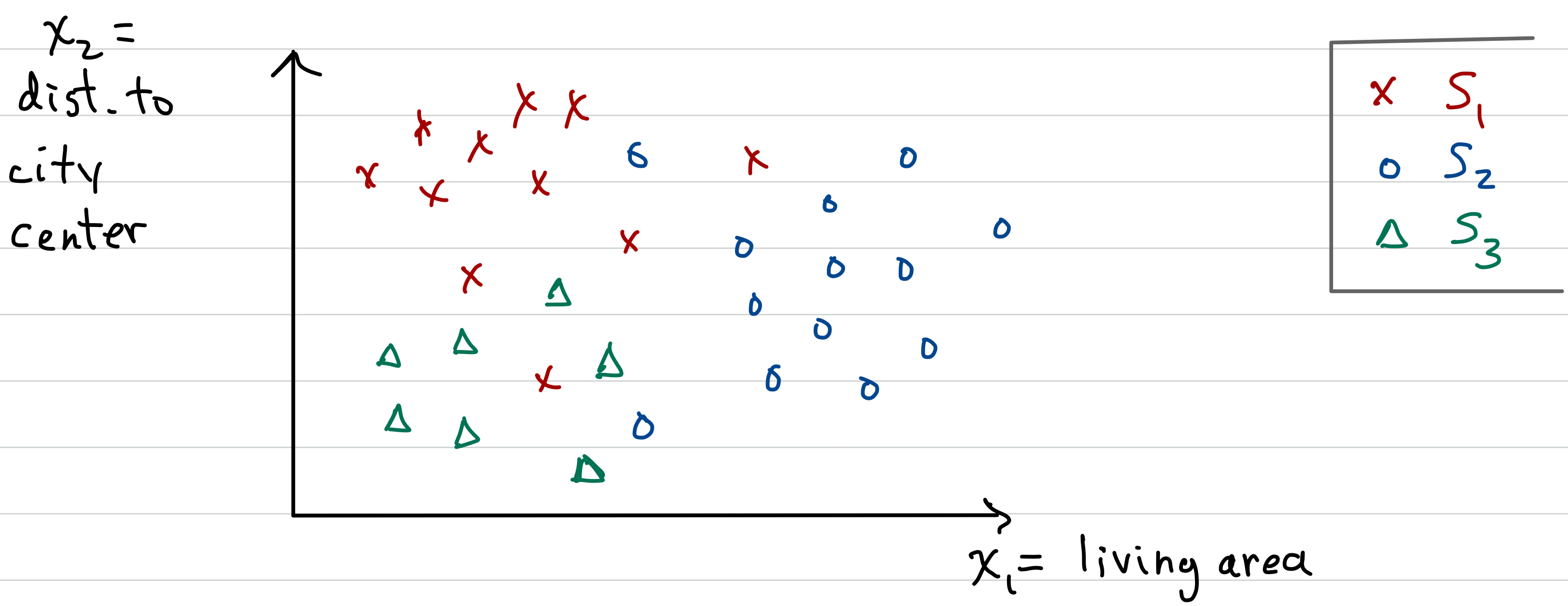
- Feature space
  - Scatter plots and decision regions
  - Ex: nearest-means classifier
- Discriminant functions for classification
  - 2-class problems
  - Notation
  - Multiclass problems (part 1), $C > 2$.

Feature space plots and notation

				Bishop
Let	S_1:	housing price increase	$(y=+1)$	C_1
	S_2:	housing price unchanged	$(y=0)$	C_2
	S_3:	housing price decrease	$(y=-1)$	C_3

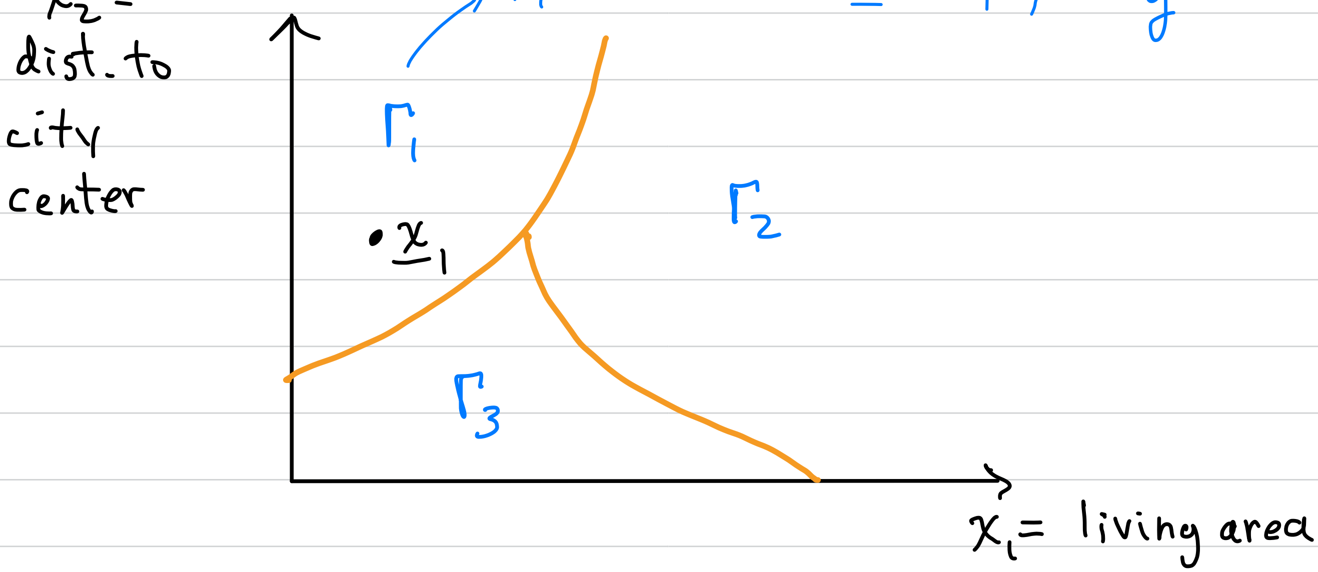
Data points labeled S_k: $\quad \underline{x}_i^{(k)}, \quad i=1,2,\cdots,N_k; \quad k=1,2,3.$

Plot training data in feature space:



To classify any point $\underline{x}$, can learn <u>decision regions</u> and <u>decision boundaries</u> in feature space, e.g.:

$\Gamma_1 \Rightarrow$ decide $\underline{x} \in S_1$, or $\hat{y}(\underline{x}) = +1$. (predict price will increase).



Γ_i is the <u>decision region</u> for S_i

$\Rightarrow$ unknown $\underline{x}_1$ would be predicted to be in S_1 (price increase)

is a <u>decision boundary</u>

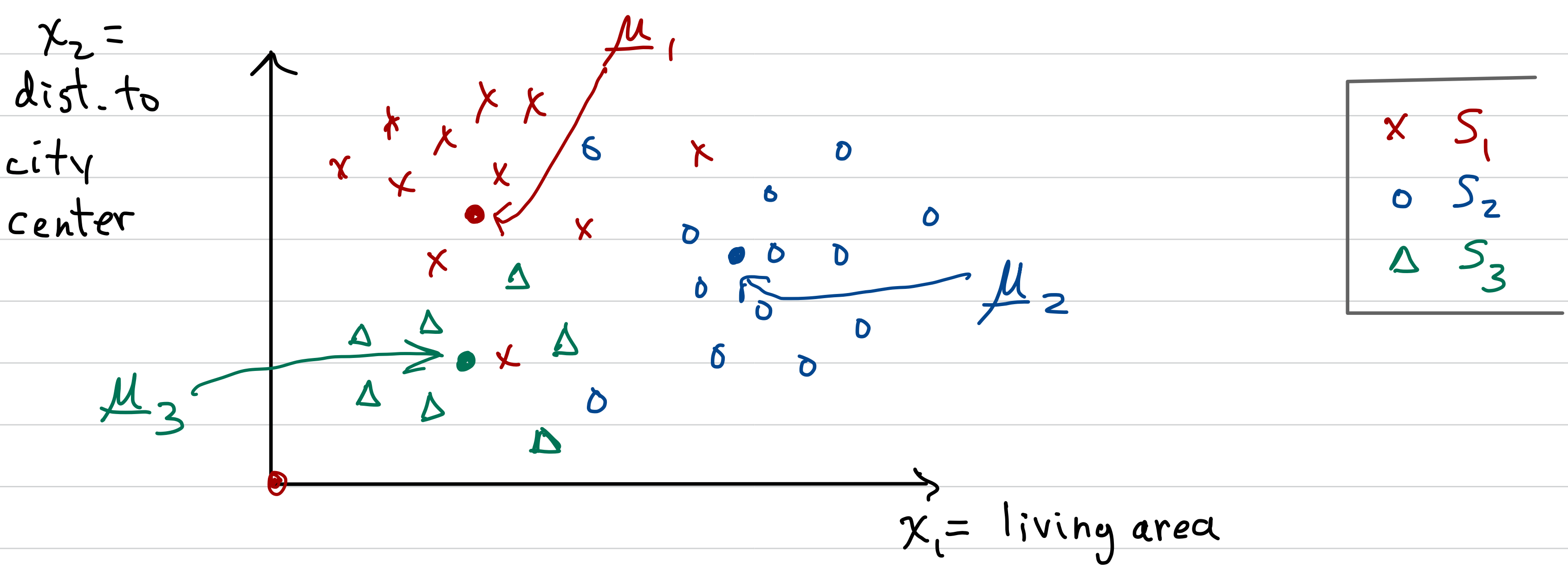
Example: Nearest-means classifier

Let N_k = # data points of class S_k in training data set, $k = 1, 2, \cdots, C$

1. Represent each class S_k by its sample mean $\underline{\mu}_k$ of the training data:

$$\underline{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \underbrace{\underline{x}_i^{(k)}}_{\text{training data point } i \text{ of class } S_k.}$$

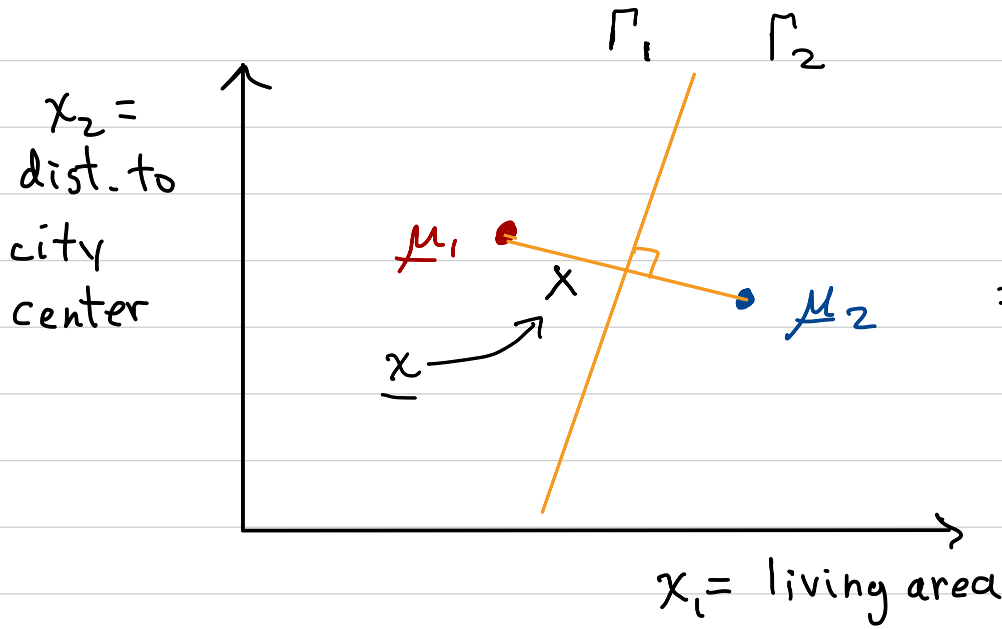
Ex: $C = 3$-class problem:



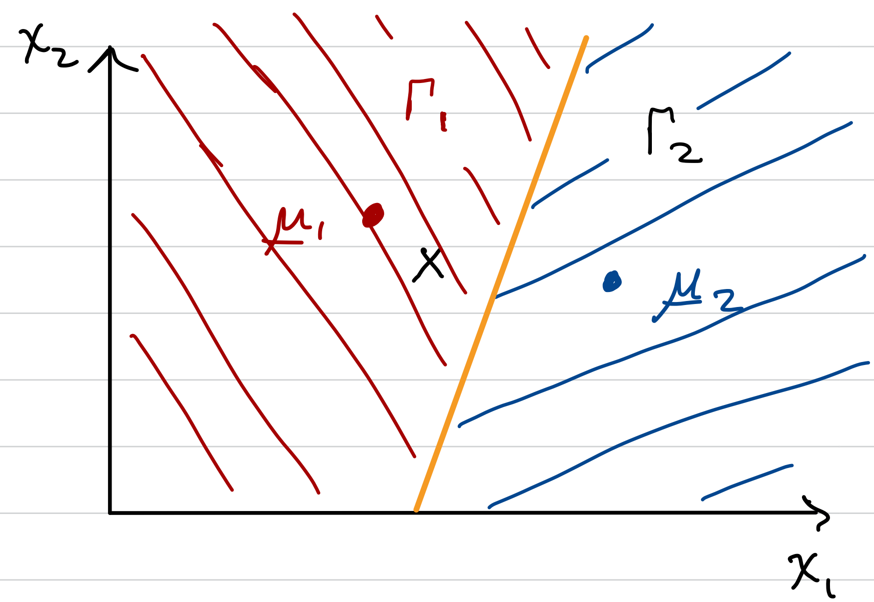
2. Each point $\underline{x}$ is classified as the class S_k of its nearest mean.

3. This defines decision regions and boundaries

Ex: $C = 2$-class problem:



$\Rightarrow$

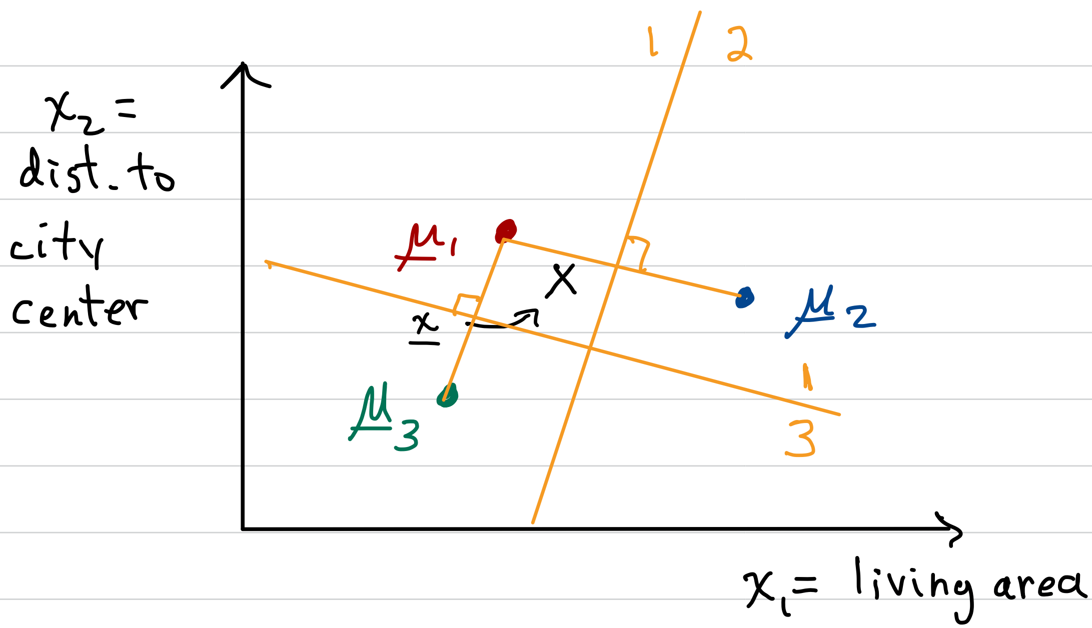


$\Rightarrow \underline{x} \in \Gamma_1 \Rightarrow$ predict $\hat{y}(\underline{x}) = +1$

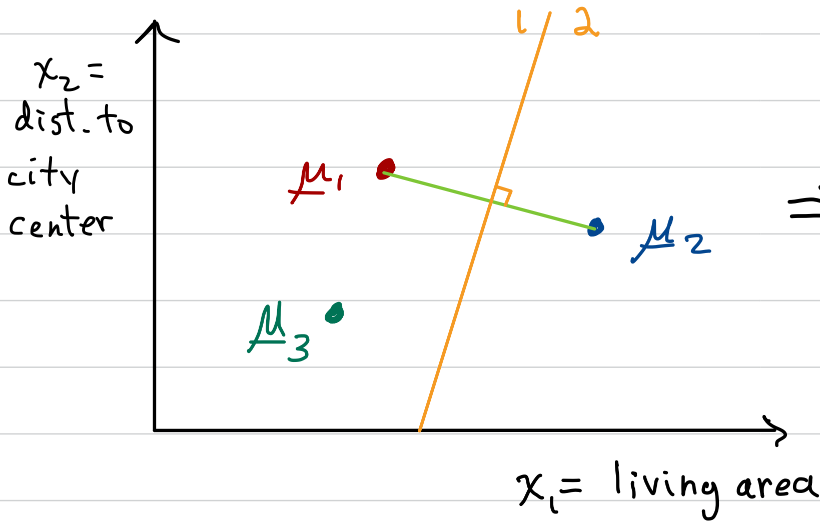
(price will increase)

Ex: C = 3-class problem

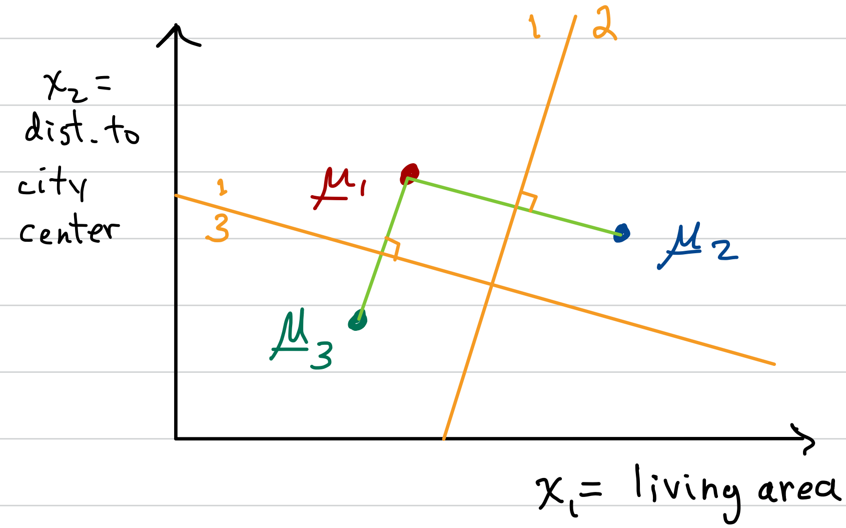
2. Each point $\underline{x}$ is classified as the class S_k of its nearest mean.



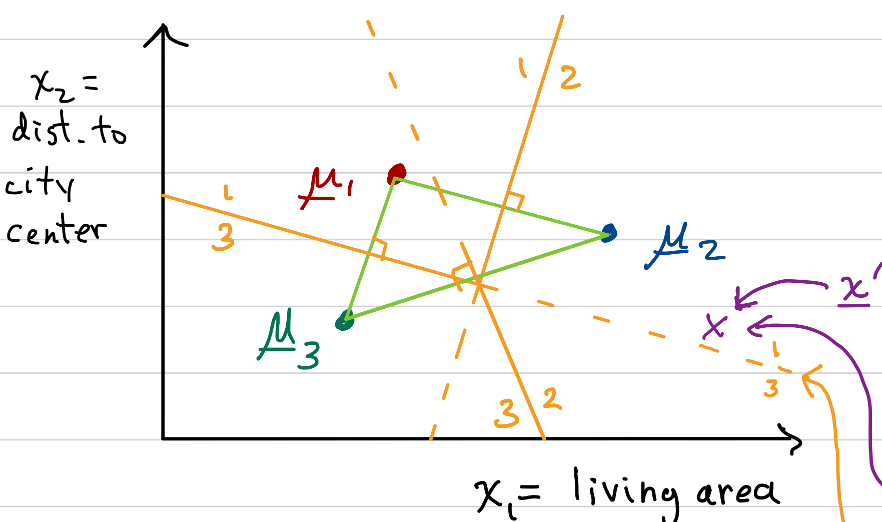
3. This defines the decision regions and boundaries



$\Rightarrow$



$\Leftarrow$



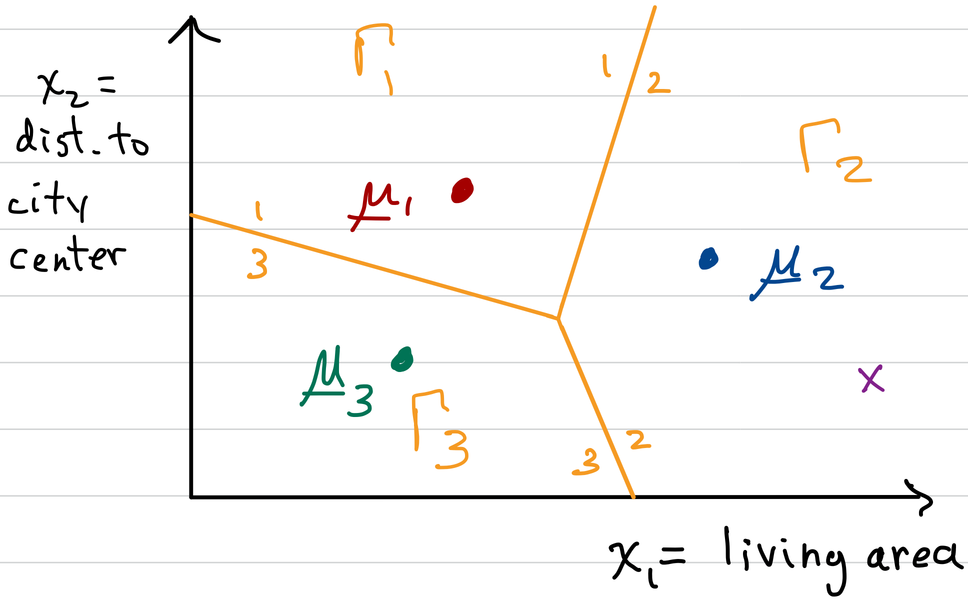
$\frac{i}{j}$ decision boundary between class S_i and S_j.

– – – Some might be redundant for a final 3-class decision boundary.

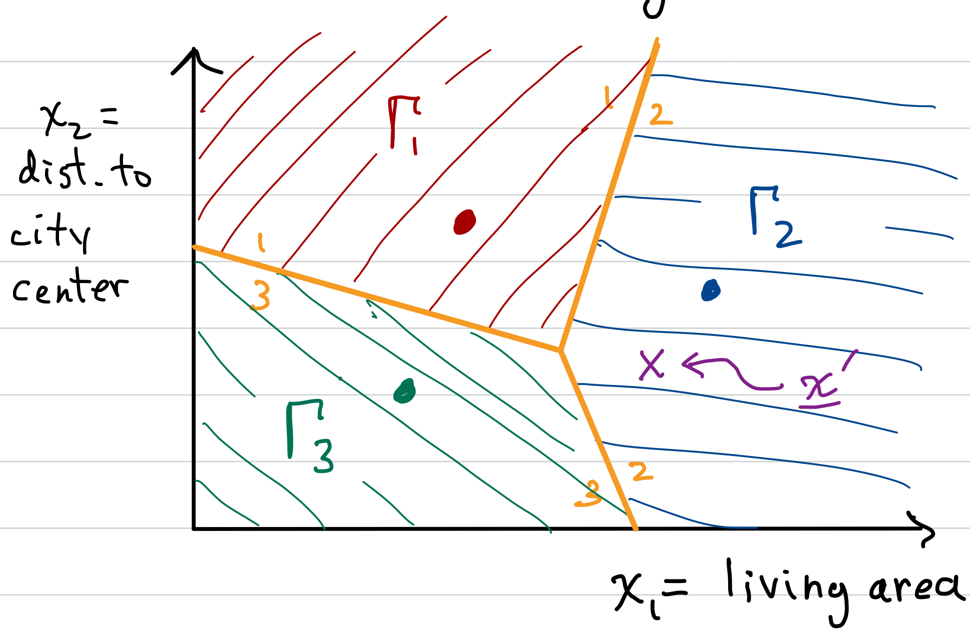
μ_2 is closer than μ_1 or μ_3 here; S_1 vs. S_3 decision is not needed.

redundant boundary.

Final 3-class decision boundaries



Final 3-class decision regions and boundaries



$\underline{x}' \in \Gamma_2 \Rightarrow \hat{y}(\underline{x}') = 0$

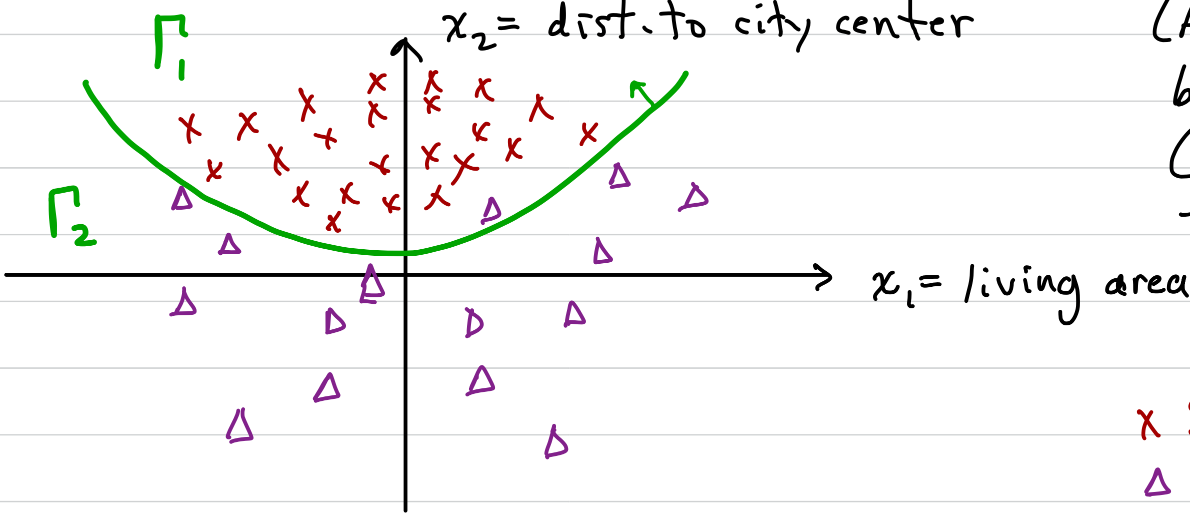
(price unchanged)

Discriminant Functions for Classification [Bishop 4.1]

1) 2-class problems

A way of representing (decision) regions and (decision) boundaries in feature space, algebraically.

Scatter plot of training data in feature space:



(Assume features have been normalized (centered) to have 0 mean)

x S_1: price increase
Δ S_2: price decrease

Γ_1 is decision region for classifying $\underline{x}$ as S_1
Γ_2 is decision region for classifying $\underline{x}$ as S_2

Let $g(\underline{x})$ = <u>discriminant function</u>, defined so that:

Decision rule: $g(\underline{x}) > 0 \Rightarrow \underline{x} \in \Gamma_1$ (predicts price will increase)
$g(\underline{x}) < 0 \Rightarrow \underline{x} \in \Gamma_2$ (predicts price will decrease)
$g(\underline{x}) = 0 \Rightarrow \underline{x}$ is on decision boundary.

Shorthand notation: $g(\underline{x}) \underset{\Gamma_2}{\overset{\Gamma_1}{\gtrless}} 0$ ← ($c = 2$ classes)

<u>Linear case</u>

$g(\underline{x})$ can be expressed as a linear fcn. of $\underline{x}$. Let $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ (2D case)

2D: $g(\underline{x}) = w_0 + w_1 x_1 + w_2 x_2$

D-DIMENSIONS: $g(\underline{x}) = w_0 + \underline{w}^T \underline{x}$

<u>Def</u>: A 2-class classifier is <u>linear</u> iff $g(\underline{x})$ can be expressed as a linear function of $\underline{x}$

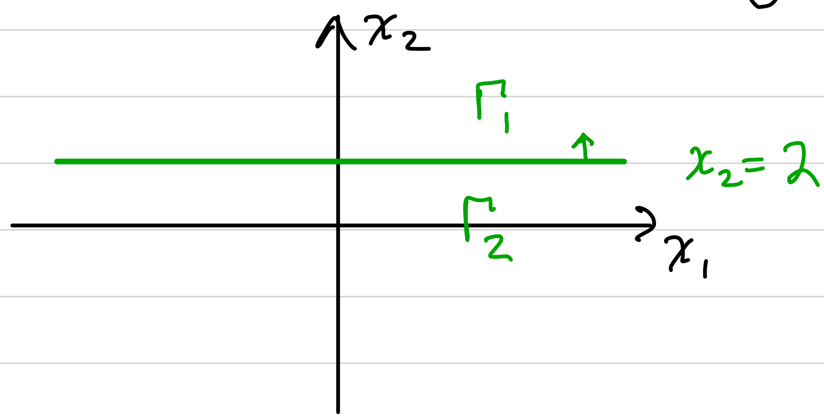
A set of data points is <u>linearly separable</u> in a 2-class problem if all the points can be correctly classified using a linear $g(\underline{x})$.

Nonlinear case

If g cannot be expressed as a linear fcn. of $\underline{x}$, then g represents a <u>nonlinear classifier</u>.

Uniqueness of $g(\underline{x})$

For a given decision boundary H and decision regions Γ_k, is $g(\underline{x})$ uniquely determined?



we need
$$\begin{cases} g\left(\begin{bmatrix} x_1 \\ 2 \end{bmatrix}\right) = 0 \\ g(x_2 > 2) > 0 \\ g(x_2 < 2) < 0. \end{cases}$$

Examples:
$$\left.\begin{aligned} g(\underline{x}) &= (x_2 - 2) \quad \checkmark \\ g(\underline{x}) &= 2(x_2 - 2) \quad \checkmark \\ g(\underline{x}) &= (x_2 - 2)^3 \quad \checkmark \end{aligned}\right\} \Rightarrow \text{Not unique.}$$

EE 559

Notation

(part 1)

Classes

Number of classes = C

k^{th} class is denoted S_k (Bishop C_k)

S_k, $k = 1, 2, \cdots, C$ defines all classes

Feature space (non-augmented)

(dimensionality D = # of features)

Later: $\mathcal{D}_{val}$ (validation set)

Datasets (full dataset, training dataset, test dataset): $\mathcal{D}$, $\mathcal{D}_{\text{Tr}}$, $\mathcal{D}_{\text{Test}}$

contain N, N_{Tr}, N_{Test} data points, respectively. , N_{val}

Data points: $\underline{x}_i$, $i = 1, 2, \cdots, N$

Data points of class S_k: $\underline{x}_i^{(k)}$, $i = 1, 2, \cdots, N_k$

living area

A data point, showing its vector components (features): $\underline{x} = \left(x_1, x_2, \cdots, x_j, \cdots, x_D\right)^T$

$$\underline{x}_i^{(k)} = \left(x_{i1}^{(k)}, x_{i2}^{(k)}, \cdots, x_{ij}^{(k)}, \cdots, x_{iD}^{(k)}\right)^T = i^{\text{th}} \text{ data point of class } S_k$$

2nd feature (dist. to c.c.) of $i^{\underline{th}}$ data pt.

$i^{\underline{th}}$ data pt. (house)

$1^{\underline{st}}$ feature (liv. area) of $i^{\underline{th}}$ data pt. ($i^{\underline{th}}$ house)

Thus, $x_{ij}^{(k)} = j^{\text{th}}$ feature of i^{th} data point of class S_k

Outputs and class labels

Classification output (predicted class): $\hat{y}$ or $\hat{y}\left(\underline{x}\right)$

True (correct) class label: y (Bishop: t)

Discriminant function (2-class problem): $g\left(\underline{x}\right)$ (Bishop: $y\left(\underline{x}\right)$)

Decision region in feature space: Γ_k for deciding class S_k

Regression output (predicted value): $\hat{y} = \hat{f}\left(\underline{x}\right)$ (Bishop: $y\left(\underline{x}\right)$)

2. Multiclass ($C > 2$) problems

→ Can we pose a C-class problem ($C > 2$) as a set of 2-class problems?
Yes.

(i) Use C discriminant fcns: $g_k(\underline{x})$, $k = 1, 2, \cdots, C$.

One vs. rest (OvR) (also called One vs-all)
to define C 2-class problems.
Each 2-class problem: S'_k vs. $\overline{S'_k}$ (e.g., hs. price decr. vs. hs. price not decr.) (cat vs. not cat)

for $k = 1, 2, \cdots, C$.

[see plot below]

Combine results:

OvR Decision rule: $\underline{x} \in \Gamma_k$ IFF $\underline{x} \in \Gamma'_k$ AND $\underline{x} \in \overline{\Gamma'_j}$ $\forall j \neq k$.

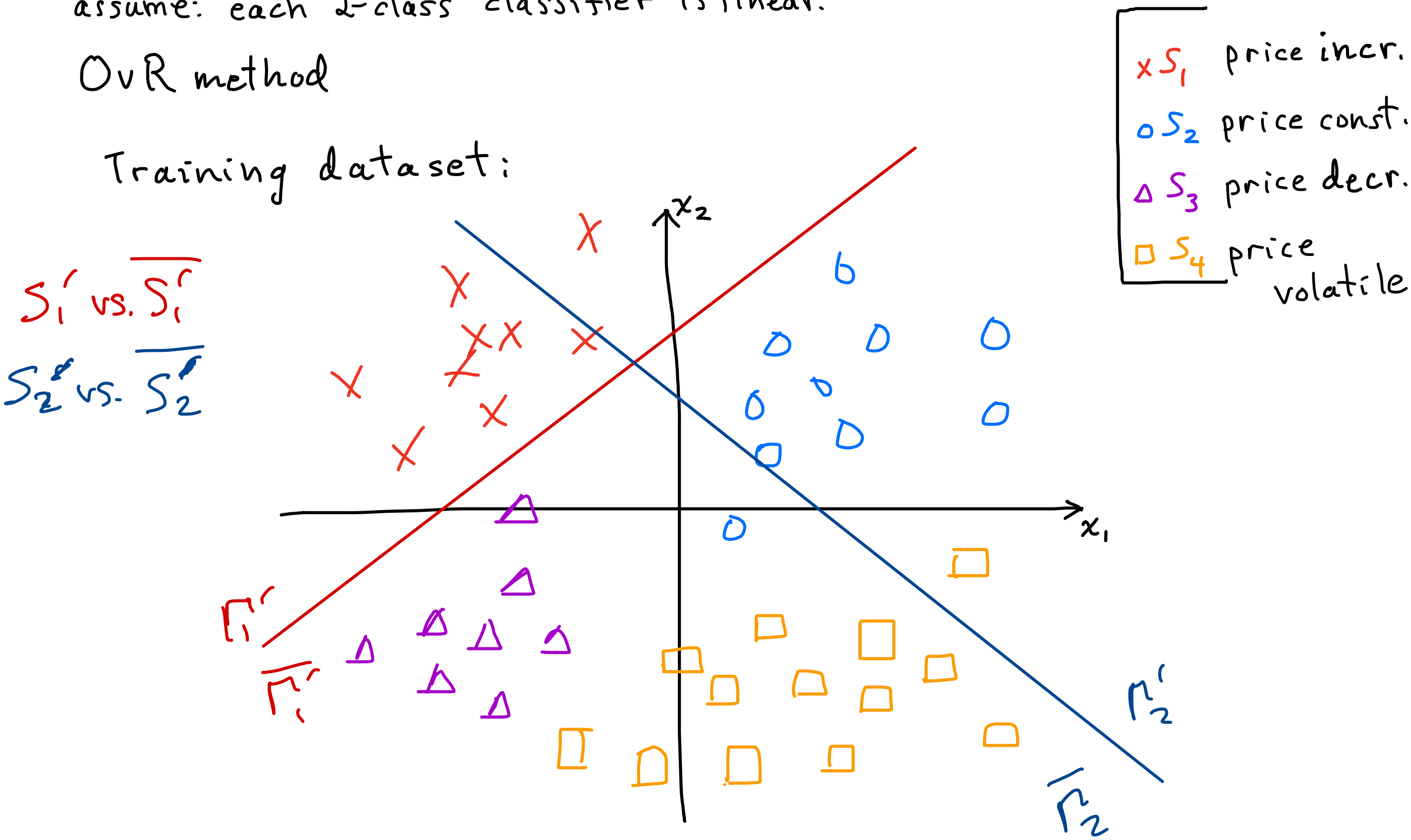
our default OvR decision rule. Other OvR decision rules are possible.

Example: Consider a $C=4$-class problem with $D=2$ features

assume: each 2-class classifier is linear.

OvR method

Training dataset:



Apply 4 2-class problems: each S_k' vs $\overline{S_k'}$

Γ_1 is defined by: all $\underline{x} \in \Gamma_1'$ and $\overline{\Gamma_2'}$ and $\overline{\Gamma_3'}$ and $\overline{\Gamma_4'}$

<u>Comment</u> In practice, often an additional ad-hoc rule is used to classify points in indeterminate regions.