Statistical Learning VII

Unsupervised Learning:

1. K-means
2. EM for Gaussian Mixtures

Constructing Features:

3. Intro
4. Splines:
   a) Approach 1
   b) Approach 2
   c) Example

Unsupervised Learning

Intro

Supervised: $T = \begin{pmatrix} y_1 \\ x_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_N \\ x_N \end{pmatrix}$ ← can evaluate every method by EPE

Unsupervised: $T = x_1, \ldots, x_N$

What can we do?

Clustering: split data into groups of points (clusters) that are similar

1. What do you mean by "similar"?
2. What kind of clusters are you looking for?

K-means

(1) Squared Euclidean distance between $x_i, x_{i'}$:
$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

(2) Find K clusters s.t. distance to mean within each cluster is small.

$C(i) \in \{1, \dots, K\}$ is cluster assigned to x_i.

$$\text{minimize} \sum_{k=1}^{K} \sum_{i:C(i)=k} d(x_i, \mu_k) \qquad (*)$$

where μ_k = mean of points in cluster k

NB. Mistake in book! (multiplies by N_k in each cluster)

Algorithm:
1. Initialize cluster assignment C
2. Set μ_k to current mean in cluster k
3. Given means $\mu_1, \dots, \mu_K$, assign each x_i to nearest mean to get new C
4. repeat from 2 until no changes in C

- Converges to local minimum
- May choose K s.t. increasing K does not reduce (*) very much.

Gaussian Mixtures and EM

- like k-means, but with "soft" cluster assignments (each cluster is a Gaussian modelled by)
- for simplicity: explain for two clusters

cluster 1: $X \sim N(\mu_1, \sigma_1^2)$ w.p. $1-\pi$
cluster 2: $X \sim N(\mu_2, \sigma_1^2)$ w.p. π

probability density: $(1-\pi)\phi_{\mu_1,\sigma_1^2}(x) + \pi\phi_{\mu_2,\sigma_2}(x)$

To find parameters:

ML is difficult numerically

+ gives bad solution with $\hat{\sigma}_1 = 0$

$\hat{\mu}_1 = x_i$ for some i.

Solution: algorithm very similar to K-means

Hidden variables $\Delta_i = \begin{cases} 0 & \text{if } x_i \text{ in cluster 1} \\ 1 & \text{if } x_i \text{ in cluster 2} \end{cases}$

1. Initializes parameters $\underbrace{\hat{\pi}, \hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2}_{\hat{\theta}}$, $(\hat{\sigma}_1 > 0.5, \hat{\sigma}_2 > 0.5)$

2. Soft cluster assignments:

$$\hat{\gamma}_i = \mathbb{E}[\Delta_i | \hat{\theta}, T] = \frac{\hat{\pi}\varphi_{\hat{\mu}_2, \hat{\sigma}_2}(x_i)}{\hat{\pi}\varphi_{\hat{\mu}_2, \hat{\sigma}_2}(x_i) + (1-\hat{\pi})\varphi_{\hat{\mu}_1, \hat{\sigma}_1}(x_i)}$$

3. Update parameter estimates for clusters:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N}(1-\hat{\gamma}_i)x_i}{\sum_i (1-\hat{\gamma}_i)} \qquad \hat{\mu}_2 = \frac{\sum_{i=1}^{N}\hat{\gamma}_i x_i}{\sum_i \hat{\gamma}_i}$$

$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^{N}(1-\hat{\gamma}_i)(x_i - \hat{\mu}_1)^2}{\sum_i (1-\hat{\gamma}_i)} \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^{N}\hat{\gamma}_i(x_i - \hat{\mu}_2)^2}{\sum_i \hat{\gamma}_i}$$

$$\hat{\pi} = \frac{\sum_{i=1}^{N}\hat{\gamma}_i}{\sum_{i=1}^{N}\hat{\gamma}_i + \sum_{i=1}^{N}(1-\hat{\gamma}_i)} = \frac{\sum_{i=1}^{N}\hat{\gamma}_i}{N}$$

4. repeat from 2 until convergence

- converges to local minimum.

3. Constructing Features: Intro

How do we construct new features in a principled way?
Today: statistical approach
Next week: neural networks

Motivation: figs 12.2, 12.3

$\begin{pmatrix} y_1 \\ x_1 \end{pmatrix}, \ldots \begin{pmatrix} y_N \\ x_N \end{pmatrix}$ $\quad$ p features: $x \in \mathbb{R}^p$

$\downarrow$

M features: $\tilde{x} \in \mathbb{R}^M$

$$\tilde{x} = h(x) = \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_M(x) \end{pmatrix}$$ basis functions

$$f(x) = \tilde{x}^T \beta = \sum_{m=1}^{M} \beta_m h_m(x)$$

How to fit parameters β? This is just a linear model in the new features $\tilde{x}$!

To avoid overfitting:
- be very selective and keep M small
- or use model selection (e.g. Lasso, best-subset, ...)
- or use regularization (e.g. ridge)

Examples:
* $h_m(x) = x_m$ for $m = 1, \ldots, p$ recovers original features
* Suppose $p = 1$. Then
  $h_m(x) = x^{m-1}$ for $m = 1, \ldots, M$ gives polynomial of degree $M-1$ = order M

* $h_m(x) = \log(x_m)$ or $h_m(x) = \sqrt{x_m}$:
commonly used transformations you might know from other courses

* $h_m(x) = I(a_m \le x_m < b_m) = \begin{cases} 1 & \text{if } a_m \le x_m < b_m \\ 0 & \text{otherwise} \end{cases}$

some numbers a_m, b_m

I: indicator function

This is like constructing a categorical variable that tells us if $x_m \in [a_m, b_m)$ or not.

* last week:
basis functions with parameters that were estimated from data
this week:
basis functions are fixed.

4. Splines

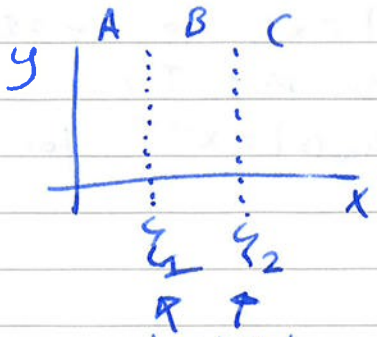
Suppose $p = 1$. (We start with only one feature.)

Motivation: tweaking coefficients of a polynomial for one region may have large effect on other regions. (see slides)

piece-wise polynomial: divide feature space in regions and fit a separate polynomial for each region.

Approach 1:

$$f(x) = \begin{cases} \beta_1 + \beta_2 x + \beta_3 x^2 & \text{if } x \in A \\ \beta_4 + \beta_5 x + \beta_6 x^2 & \text{if } x \in B \\ \beta_7 + \beta_8 x + \beta_9 x^2 & \text{if } x \in C \end{cases}$$



knots: boundaries between regions (Greek letter xi)

Can we do this with a single linear model? yes!

$$f(x) = \beta_1 I(x\in A) + \beta_2 x I(x\in A) + \beta_3 x^2 I(x\in A)$$
$$+ \beta_4 I(x\in B) + \beta_5 x I(x\in B) + \beta_6 x^2 I(x\in B)$$
$$+ \beta_7 I(x\in C) + \beta_8 x I(x\in C) + \beta_9 x^2 I(x\in C)$$

Dictionary:
$$h_1(x) = I(x\in A) \quad h_2(x) = x\cdot I(x\in A) \quad h_3(x) = x^2 I(x\in A)$$
$$h_4(x) = I(x\in B) \quad h_5(x) = x I(x\in B) \quad h_6(x) = x^2 I(x\in B)$$
$$h_7(x) = I(x\in C) \quad h_8(x) = x I(x\in C) \quad h_9(x) = x^2 I(x\in C)$$

Disadvantage (figs 5.1, 5.2): piece-wise polynomials are not continuous at knots. But we can fix that!

Require continuity at knots:

$$\beta_1 + \beta_2 \xi_1 + \beta_3 \xi_1^2 = \beta_4 + \beta_5 \xi_1 + \beta_6 \xi_1^2$$
$$\beta_4 + \beta_5 \xi_2 + \beta_6 \xi_2^2 = \beta_7 + \beta_8 \xi_2 + \beta_9 \xi_2^2$$

Two linear equalities, so can eliminate two parameters. For instance, can express β_4 and β_7 in terms of the other parameters.

Disadvantage (figs 5.1, 5.2): first derivatives can still be discontinuous at knots. But we can fix that!

$$\frac{d}{dx}\beta_1 + \beta_2 x + \beta_3 x^2 = \frac{d}{dx}\beta_4 + \beta_5 x + \beta_6 x^2 \quad \text{at } x = \xi_1$$
$$\beta_2 + 2\beta_3 \xi_1 = \beta_5 + 2\beta_6 \xi_1$$

likewise:
$$\beta_5 + 2\beta_6 \xi_2 = \beta_8 + 2\beta_9 \xi_2$$

Again two linear equalities, so can eliminate two parameters. E.g. can express β_5, β_8 in terms of the other parameters.

Parameters left: $\beta_1, \beta_2, \beta_3, \beta_6, \beta_9$

book: order $M = d+1$

Splines: piece-wise polynomials of degree d with
K knots + continuous at knots
+ first $d-1$ derivatives all continuous at knots

Most common in practise: $d=0$, $d=1$, $d=3$

Number of remaining parameters:

$$(K+1 \text{ regions}) \cdot (d+1 \text{ parameters per region}) - (K \text{ knots}) \cdot (d \text{ constraints per knot}) = d+1+K$$

In our example: $d=2$, $K=2$, so $2+1+2=5$

b) Approach 2

Eliminating parameters using constraints is inconvenient.
Can we find a basis that already incorporates the constraints? Yes!

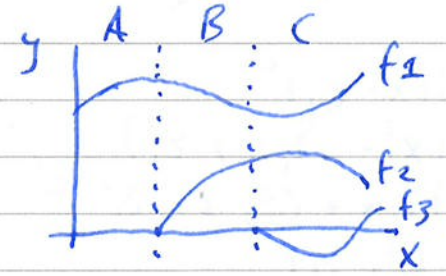
$$z_+ = \max\{z, 0\} = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

standard polynomial $\Rightarrow$ $h_1(x)=1$, $h_2(x)=x$, $h_3(x)=x^2$, $\ldots$, $h_{d+1}(x)=x^d$,

$$h_{d+2}(x) = (x-\xi_1)_+^d, \; h_{d+3}(x) = (x-\xi_2)_+^d, \ldots, h_{d+1+K}(x) = (x-\xi_K)_+^d$$

one extra feature per region except for the first region.

Idea behind this:



$f(x) = f_1(x) + f_2(x) + f_3(x)$

1. Make polynomial $f_1(x)$ over regions A, B, C that is correct in region A
2. Add polynomial $f_2(x)$ over regions B, C that makes $f_1(x)+f_2(x)$ correct over region B
3. Add polynomial $f_3(x)$ over region C that makes $f_1(x)+f_2(x)+f_3(x)$ correct over region C.

$$f_1(x) = \beta_1 + \beta_2 x + \beta_3 x^2$$
$$f_2(x) = \beta_4 (x-\xi_1)^2 I(x \geq \xi_1) = \beta_4 (x-\xi_1)_+^2$$
$$f_3(x) = \beta_5 (x-\xi_2)^2 I(x \geq \xi_2) = \beta_5 (x-\xi_2)_+^2$$

Do the features/basis functions in approach 2 express the same splines as those from approach 1?

Yes! Will show in case one knot ($K=1$) that approach 2 can express any function from approach 1. (Then, as approach 2 has the same number of degrees of freedom $d+1+K$, we also know that it cannot express more functions than approach 1.)

Approach 1: $$f(x) = \begin{cases} \beta_1 + \beta_2 x + \beta_3 x^2 & \text{if } x < \xi_1 \\ \beta_4 + \beta_5 x + \beta_6 x^2 & \text{if } x \geq \xi_1 \end{cases}$$

At $x = \xi_1$:
$$\beta_1 + \beta_2 x + \beta_3 x^2 = \beta_4 + \beta_5 x + \beta_6 x^2 \quad \text{(continuity)}$$
$$\beta_2 x + 2\beta_3 x = \beta_5 + 2\beta_6 x \quad \text{(continuous first derivative)}$$

$$\Downarrow$$

$$\beta_2 - \beta_5 = 2(\beta_6 - \beta_3)\xi_1$$
$$\beta_4 - \beta_1 = (\beta_2 - \beta_5)\xi_1 + (\beta_3 - \beta_6)\xi_1^2$$
$$= 2(\beta_6 - \beta_3)\xi_1^2 + (\beta_3 - \beta_6)\xi_1^2 = (\beta_6 - \beta_3)\xi_1^2$$

To express $f(x)$ in approach 2, take
$$g(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \underbrace{(\beta_6 - \beta_3)}_{\text{this becomes parameter in approach 2}} (x-\xi_1)_+^2$$

If $x < \xi_1$: $g(x) = f(x)$

If $x \geq \xi_1$:
$$g(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + (\beta_6 - \beta_3)(x-\xi_1)^2$$
$$= \beta_1 + \beta_2 x + \beta_3 x^2 + (\beta_6 - \beta_3)(x^2 - 2\xi_1 x + \xi_1^2)$$
$$= \beta_1 + \beta_2 x + \beta_6 x^2 - 2(\beta_6 - \beta_3)\xi_1 x + (\beta_6 - \beta_3)\xi_1^2$$
$$= \beta_1 + \beta_2 x + \beta_6 x^2 + (\beta_5 - \beta_2)x + (\beta_4 - \beta_1)$$
$$= \beta_4 + \beta_5 x + \beta_6 x^2 = f(x)$$

c) Example from book Section 5.2.2:

Classification in South African Heart Disease Data by logistic regression

$p = 6$ features: 5 numeric features
1 binary feature

slight extension of splines

* transform each numeric feature to a $d = 3$ natural spline basis with 4 basis functions + add intercept
* this gives $2 + 5 \cdot 4 = 22$ new features

* space knots by quantiles of the features

this uses the data, so strictly speaking the basis functions are not fixed anymore

* model selection:
  - a form of backward stepwise selection
  - do not drop single feature, but drop all the features that belong to the same original feature simultaneously.

(Example R code)