COMPSCI 330 Lecture 23 PCA, Linear Regression

Monday, November 21, 2016 1:00 PM

- Principal Component Analysis (continued)
- Linear models and Least Squares

- Recall: eigenvector decomposition : $A = \sum_{i=1}^{n} x_i x_i^T$

$$A = \sum_{i=1}^{d} \lambda_i v_i v_i^T \qquad (\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d)$$

$\{\lambda_i\}$'s are eigenvalues, $\{v_i\}$'s are eigenvectors,

$\{v_i\}$'s are orthogonal to each other.

$$\left(A v_j = \left[\sum_{i=1}^{d} (\lambda_i v_i v_i^T)\right] v_j = \sum_{i=1}^{d} \lambda_i v_i (v_i^T v_j) = \lambda_j v_j \right)$$

- Power method

initialize u^0 as a random vector

for $i = 1$ to t

$$u^i = \frac{A u^{i-1}}{\|A u^{i-1}\|}$$

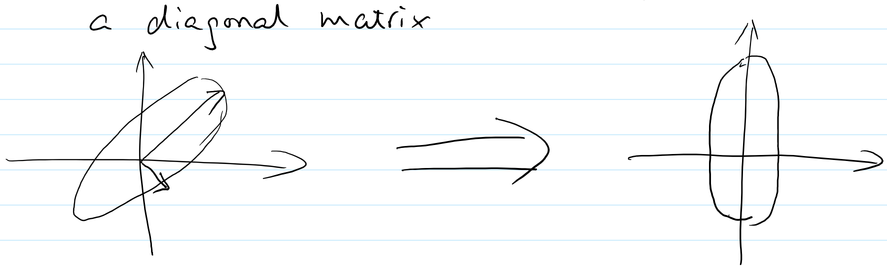
return u^t

observe $u^1 \sim A u^0$ $\quad u^2 \sim A(Au^0) = A^2 u^0$

$$u^t \sim A^t u^0$$

- Claim: With high probability, when $t \geq \frac{(\log \frac{d}{\varepsilon})\lambda_1}{\lambda_1 - \lambda_2}$, then $\|u^t - v_1\| \leq \varepsilon$.

- Proof: intuition: change basis to v_i's, and interpret the matrix as a diagonal matrix



because $v_1, v_2, \cdots, v_d$ are orthogonal to each other

can write $u^0 = c_1^0 v_1 + c_2^0 v_2 + \cdots + c_d^0 v_d$

$$u^1 = c_1^1 v_1 + c_2^1 v_2 + \cdots + c_d^1 v_d$$

$$\cdots$$

$$u^t = c_1^t v_1 + c_2^t v_2 + \cdots + c_d^t v_d$$

for now, assume $u^i = A u^{i-1}$ (will do normalization at the end)

$$u^i = A u^{i-1} = A(c_1^{i-1} v_1 + c_2^{i-1} v_2 + \cdots + c_d^{i-1} v_d)$$

$$(A v_1 = \lambda_1 v_1 , A v_2 = \lambda_2 v_2 , \cdots)$$

$$= \lambda_1 c_1^{i-1} v_1 + \lambda_2 c_2^{i-1} v_2 + \cdots + \lambda_d c_d^{i-1} v_d$$

but u^i can also be written as

$$u^i = c_1^i v_1 + c_2^i v_2 + \cdots + c_d^i v_d$$

therefore $\quad c_j^i = \lambda_j c_j^{i-1}$

$$c_j^i = (\lambda_j)^i c_j^0$$

for simplicity assume $c_j^0 \approx \pm 1$ (in reality $c_j^0 \sim N(0,1)$)

$$c_j^t \sim \pm (\lambda_j)^t$$

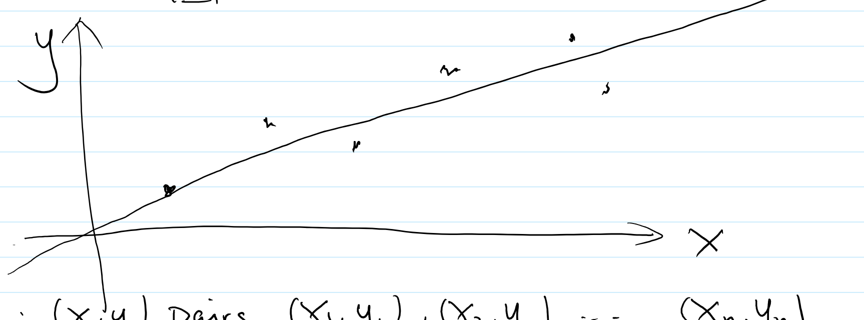
when t is as claimed

$$|c_1^t| > \frac{d}{\varepsilon} |c_2^t| \geq \frac{d}{\varepsilon} |c_j^t| \quad (\forall j > 2)$$

$\Longrightarrow$ u^t is ε-close to v_1 $\square$.

- simplest example of supervised learning : linear regression
  - assumption: output (y) is a linear function over the input (x)

$$y \approx \sum_{i=1}^{d} w_i x_i = \langle w, x \rangle$$



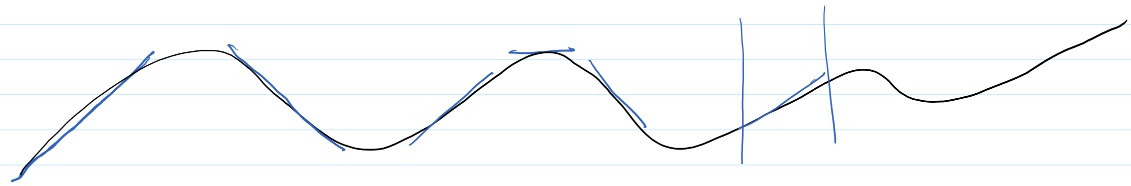
  - Given : (x, y) pairs $(x_1, y_1), (x_2, y_2) \cdots, (x_n, y_n)$

$$x_i \in \mathbb{R}^d \quad y_i \in \mathbb{R}$$

want to find $w \in \mathbb{R}^d$ s.t. $y \approx \langle w, x \rangle$

- when do we expect a linear relationship?
  1. when there is a plausible model
  2. when approximating in a small interval

  recall : calculus



3. as a first attempt.

- Common tricks

  1. handle categorical data carefully

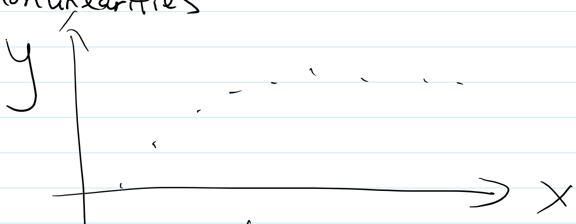
     genre for movies

     - wrong: assign each category a number 1, 2, 3 .
     - right: assign each category a dimension

       example: 5 categories $\longrightarrow$ 5 dimensions

       category 4 $\longrightarrow$ $(0, 0, 0, 1, 0)$

  2. Consider nonlinearities



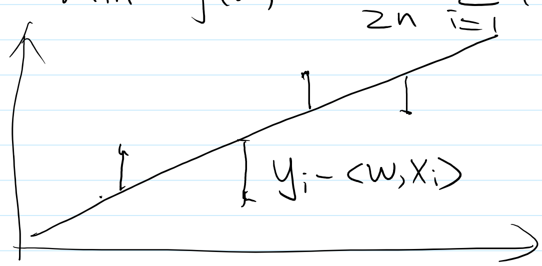
can change $x \to \sqrt{x}$

- Formalizing the regression problem

  - least-squares

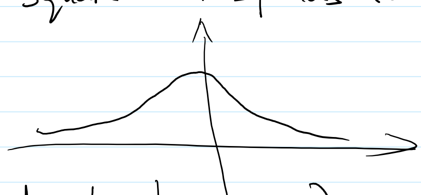
    Given $(x_1, y_1) \cdots (x_n, y_n)$, want to find w s.t.

    $$\min \; f(w) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \langle w, x_i \rangle)^2$$



  - why squares?

    1. easy to solve
    2. square corresponds to Gaussian distribution.



- how to find the best w?

- how to find the best w?
  - gradient descent algorithm
    - recall: $f : \mathbb{R}^d \to \mathbb{R}$

      gradient of f $\quad \nabla f \in \mathbb{R}^d \quad (\nabla f)_i = \frac{\partial}{\partial w_i} f(w)$

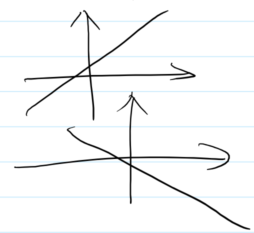
Taylor's Theorem: $f(w) \approx f(w_0) + \langle \nabla f(w_0), w - w_0 \rangle$

(true when w is close to w_0)

(in one dimension $f(x+\varepsilon) \approx f(x) + f'(x) \cdot \varepsilon$)

intuition: if $f'(x) > 0$, set $\varepsilon < 0$

$f'(x) < 0$, set $\varepsilon > 0$



in high dimensions, $w - w_0 = -\eta \nabla f(w_0)$

step size / learning rate

- Gradient descent algorithm.

  initialize $w^0 = \vec{0}$

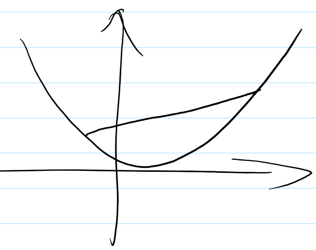
  for $i = 1$ to t

  $\quad w^i = w^{i-1} - \eta_i \nabla f(w^{i-1})$

  return w^t

- Claim: Gradient descent converges to optimal value of w.

- intuition: objective function $f(w)$ "looks like"



  - gradient descent will not work for functions like