Week 6
Advice for Applying ML

一. Evaluating a Learning Algorithm

1. Deciding what to try next

- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, \text{etc.})$
- Try decreasing λ
- Try increasing λ

$x_1, x_2, x_3, \ldots, x_{100}$

2. Evaluating a Hypothesis

Dataset { training set (70%) / test set (30%) } randomly

Steps:

<1>. Learn θ and minimize $J(\theta)$ ⇐ training emps.

<2> Compute the test set error $J_{test}(\theta)$.

$J_{test}(\theta)$

<1> Linear: $J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$

<2> Misclassification: (0, 1)类型.

$$err(h_\theta(x), y) = \begin{cases} 1, & h_\theta(x) \geq 0.5 \text{ and } y=0, \text{ or } h_\theta(x) < 0.5 \text{ and } y=1 \\ 0, & \text{otherwise} \end{cases}$$

3. Model Selection and Train / Validation / Test sets.

Dataset { training set 60% / cross validation set 20% / test set 20% }

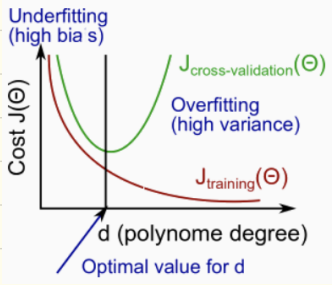
1. Optimize the parameters in Θ using the training set for each polynomial degree.
2. Find the polynomial degree d with the least error using the cross validation set.
3. Estimate the generalization error using the test set with $J_{test}(\Theta^{(d)})$, (d = theta from polynomial with lower error);

Week 6
Advice for Applying ML

二. Bias vs. Variance

1. Diagnosing Bias vs. Variance

分析模型问题到底是由 bias 还是 variance 造成的？



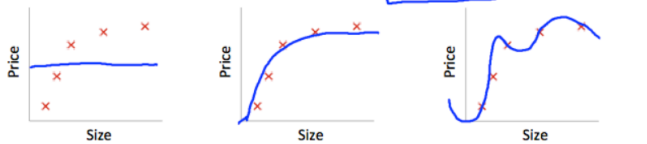
} High bias (underfitting)
} High variance (overfitting)

2. Regularization and Bias / Variance

Linear regression with regularization

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$ ←

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$ ←



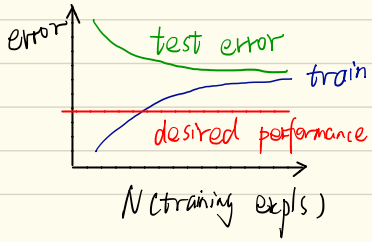
Large λ ← → High bias (underfit)
→ $\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$
$h_\theta(x) \approx \theta_0$

Intermediate λ ← "Just right"

→ Small λ High variance (overfit)
→ λ = 0

3. Learning curves

① High bias

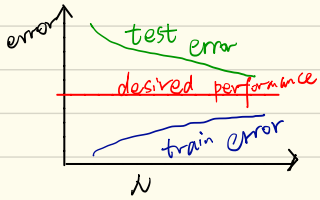


training set size ↓ : $J_{train}(\theta)$ ↓, $J_{cv}(\theta)$ ↑

training set size ↑ : $J_{train}(\theta)$ ↑, $J_{cv}(\theta)$ ↓

Week 6
Advice for Applying ML

② High Variance



training set size ↓: $J_{train}(\theta)$ ↓ $J_{cv}(\theta)$ 大
training set size ↑: $J_{train}(\theta)$ 增大 $J_{cv}(\theta)$ 减小
$J_{train}(\theta) < J_{cv}(\theta)$
且 相差很大

四. Deciding What to Do Next Revisited

- **Getting more training examples:** Fixes high variance
- **Trying smaller sets of features:** Fixes high variance
- **Adding features:** Fixes high bias
- **Adding polynomial features:** Fixes high bias
- **Decreasing λ:** Fixes high bias
- **Increasing λ:** Fixes high variance.

对 neutral network.
参数多 → underfitting high bias
→ overfitting high variance 下手调节 → layer 或 → unit

Model Complexity Effects:

- Lower-order polynomials (low model complexity) have high bias and low variance. In this case, the model fits poorly consistently.
- Higher-order polynomials (high model complexity) fit the training data extremely well and the test data extremely poorly. These have low bias on the training data, but very high variance.
- In reality, we would want to choose a model somewhere in between, that can generalize well but also fits the data reasonably well.

三. Building a Spam Classifier

Week 6
Advice for Applying ML

1. Prioritizing What to Work on

选择 10000 ~ 50000 常出现 (email 中) word

⇒ X feature

- A lot of data
- Sophisticated features
- Algorithms

2. Error Analysis

首先找一个粗简模型，快速实现。

以决定 where to cost time next

Recommended approach to solving ML problem

stemming 归类衍生词

- Start with a simple algorithm, implement it quickly, and test it early on your cross validation data.
- Plot learning curves to decide if more data, more features, etc. are likely to help.
- Manually examine the errors on examples in the cross validation set and try to spot a trend where most of the errors were made.

三. Handling Skewed Data

1. Error Metrics for Skewed Classes

		Actual class: 1	Actual class: 0
Predicted class	1	True Positive	False Positive
Predicted class	0	False Negative	True Negative

$$\text{precision} = \frac{\text{True Pos}}{\#\text{predicted pos}} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Pos}}$$

$$\text{Recall} = \frac{\text{True Pos}}{\#\text{actual Pos}} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Neg}}$$

2. Trading Off precision and recall

Week 6
Advice for Applying ML

Logistic regression classifier
- Predict $y=1$ if $h_\theta(x) \geq$ threshold
- Predict $y=0$ if $h_\theta(x) <$ threshold

way to determine threshold.

$$F_1 \text{ Score} = 2\frac{PR}{P+R}$$ (maximize F_1 Score)

i.e. P (Precision)
R (recall) from cross validation set

III. Using Large Data Sets

It's not who has the best algorithm that wins,
but who has the most training sets

Week 7

Support Vector Machines

一. Large Margin Classification

1. Optimization Objective

Logistic regression:

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1-y^{(i)})\left((-\log(1-h_\theta(x^{(i)}))\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$\text{cost}_1(\theta^T x^{(i)})$ $\quad$ $\text{cost}_0(\theta^T x^{(i)})$

Support vector machine

$$\min_\theta C\sum_{i=1}^{m}\left[y^{(i)}\text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\text{cost}_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

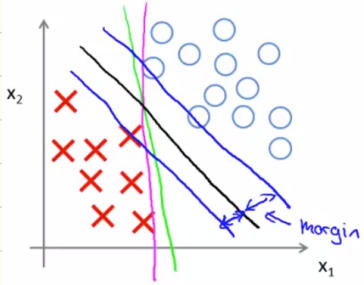
⇓ $C = \frac{1}{\lambda}$

SVM Hypothesis:

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

2. Large Margin Intuition

SVM Decision Boundary: Linearly separable case



Large margin classifier

二. kernels

↳ Compute new feature depending on landmarks.

Week 7

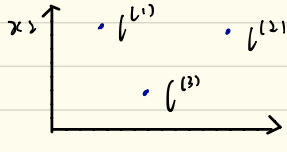
Support Vector Machines

Kernels and Similarity

$$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^{(1)})^2}{2\sigma^2}\right)$$

if $x \approx l^{(1)}$: $f_1 \approx 1$

if x far from $l^{(1)}$: $f_1 \approx 0$



SVM with Kernels

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$,
choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

Given example x:

$$f_1 = \text{similarity}(x, l^{(1)})$$
$$f_2 = \text{similarity}(x, l^{(2)})$$
$$\ldots$$

$$f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \end{bmatrix} \qquad f_0 = 1$$

For training example $(x^{(i)}, y^{(i)})$:

$x^{(i)}$: $f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)})$ ⇐ $f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp(\cdots)$.

$\vdots$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

SVM Parameters

λ ↑

(1) $C\ (= \frac{1}{\lambda})$ { Large C : Big variance & small bias
Small C : 反之

(2) σ^2 { Large σ^2 : f_i vary more smoothly ⇒ Higher bias & lower variance
Small σ^2 : 反之

Week 7

Support Vector Machines

SVM in Practice

Use SVM software package (e.g. liblinear, libsvm, ...) to solve for parameters θ.

No Kernal ("linear kernel")

Need to specify:

→ Choice of parameter C.

Choice of kernel (similarity function):

E.g. No kernel ("linear kernel")

Predict "y = 1" if $\theta^T x \geq 0$ $\quad \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \geq 0$

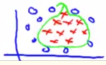
→ n large, m small $\quad x \in \mathbb{R}^{n+1}$

Gaussian kernel:

$f_i = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$, where $l^{(i)} = x^{(i)}$.

$x \in \mathbb{R}^n$, n small and/or m large

Need to choose σ^2.



Logistic regression vs. SVMs

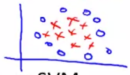
n = number of features ($x \in \mathbb{R}^{n+1}$), m = number of training examples

If n is large (relative to m): (e.g. $n \geq m$, $n = 10{,}000$, $m = 10 \cdots 1000$)

Use logistic regression, or SVM without a kernel ("linear kernel")

If n is small, m is intermediate: ($n = 1\text{-}1000$, $m = 10\text{-}10{,}000$) ←

→ Use SVM with Gaussian kernel



If n is small, m is large: ($n = 1\text{-}1000$, $m = 50{,}000+$)

→ Create/add more features, then use logistic regression or SVM without a kernel

Neural network likely to work well for most of these settings, but may be slower to train.

Week 8
Unsupervised Learning

一. Clustering

unsupervised learning. input only x. no y

cluster is an example of unsupervised ~.

1. K-means Algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$ } cluster assignment

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k } move centroid

eg. $k=2$ $x^{(1)}, x^{(3)}, x^{(5)}, x^{(6)} \Rightarrow c^{(1)}=2, c^{(3)}=2 \dots$

$\mu_2 = \frac{1}{4}[x^{(1)}, x^{(3)}, x^{(5)}, x^{(6)}] \in \mathbb{R}^n$

}

2. Optimization Objective

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2 \quad \Leftarrow \min$$

Steps

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

3. Random Initialization

① Should have $K < m$

② Randomly pick K examples

③ Set $\mu_1, \dots, \mu_K$ equal to these examples

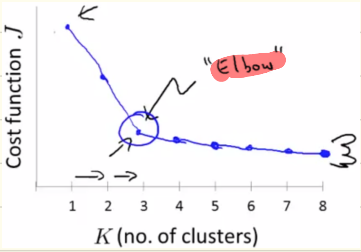
Week 8

Unsupervised Learning

4. Choosing the Number of Clusters

Choosing the value of K

Elbow Method



但在elbow不明显的情况下.
根据具体问.
例如 T-shirt: s m l
xs s m l xl

二. Motivation

1. Motivation I: Data Compression

Suppose we apply dimensionality reduction to a dataset of m examples $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, where $x^{(i)} \in \mathbb{R}^n$. As a result of this, we will get out:

- A lower dimensional dataset $\{z^{(1)}, z^{(2)}, \ldots, z^{(k)}\}$ of k examples where $k \leq n$.
- A lower dimensional dataset $\{z^{(1)}, z^{(2)}, \ldots, z^{(k)}\}$ of k examples where $k > n$.
- (selected) A lower dimensional dataset $\{z^{(1)}, z^{(2)}, \ldots, z^{(m)}\}$ of m examples where $z^{(i)} \in \mathbb{R}^k$ for some value of k and $k \leq n$.

Correct

- A lower dimensional dataset $\{z^{(1)}, z^{(2)}, \ldots, z^{(m)}\}$ of m examples

2. Motivation II: Visualization

三. Principal Component Analysis

1. PCA Problem

Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Week 8

Unsupervised Learning

PCA is Not Linear Regression

① l-r. x, y 方向 "投影".

② PCA. 垂直于 vector 的方向

2- PCA Algorithm

PCA Algorithm Summary

After mean normalization (ensure every feature has zero mean) and optionally feature scaling:

$$\text{Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

$$X = \begin{bmatrix} — (x^{(1)})^T — \\ \vdots \\ — (x^{(m)})^T — \end{bmatrix}$$

Sigma = (1/m) * X' * X;

```
[U,S,V] = svd(Sigma);
Ureduce = U(:,1:k);
z = Ureduce'*x;
```

$x \in \mathbb{R}^n$ ~~$x_0 = 1$~~

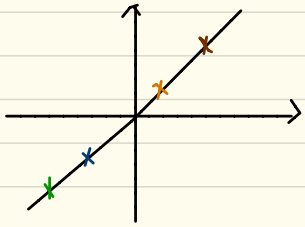
II. Applying PCA

1. Reconstruction from Compressed (representation)

$$z = U_{reduce}^T \cdot x$$

reconstruction

$$X_{approx} = U_{reduce} \cdot z$$



2. Choosing the Number of Principal Components k (dimensions)

~~平方~~ 投影误差. $\frac{1}{m}\sum_{i=1}^{m}|x^{(i)} - x_{approx}^{(i)}|^2$

Total Variation $\frac{1}{m}\sum_{i=1}^{m}|x^{(i)}|^2$.

Week 8

Unsupervised Learning

Typically, choose k to be smallest value so that

$$\frac{\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} - x_{approx}^{(i)} \|^2}{\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} \|^2} \leq 0.01 \quad (1\%)$$

(0.05, 0.10 都可，按需求)

"99% of variance is retained"

(99% 样本都被保留)

$$1 - \frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \leq 0.01 \quad (\text{For given } k)$$

`[U,S,V] = svd(Sigma)`

Pick smallest value of k for which

$$\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{m} S_{ii}} \geq 0.99$$

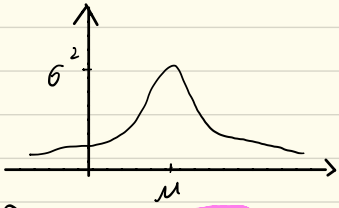
(99% of variance retained)

Week 9

Anomaly Detection

一. Density Estimation

1. Problem Motivation
2. Gaussian (Normal) Distribution (正态分布)



$$p(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Parameter Estimation

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)}-\mu)^2$$

For given dataset $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

3. Algorithm

Anomaly detection algorithm

1. Choose features x_i that you think might be indicative of anomalous examples. $\{x^{(1)}, \dots, x^{(m)}\}$
2. Fit parameters $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$$

$p(x_j; \mu_j, \sigma_j^2)$

$\mu_1, \mu_2, \dots, \mu_n$

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

3. Given new example x, compute $p(x)$:

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if $p(x) < \varepsilon$

二. Building an Anomaly Detection System.

training sets 都来自 good (normal) examples

Week 9

Anomaly Detection

Algorithm evaluation

Fit model $p(x)$ on training set $\{x^{(1)}, \ldots, x^{(m)}\}$

On a cross validation/test example x, predict

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

Possible evaluation metrics:

- True positive, false positive, false negative, true negative
- Precision/Recall
- F_1-score

skewed classification problem

Can also use cross validation set to choose parameter ε

2. Anomaly Detection vs. Supervised Learning

Anomaly detection	vs.	Supervised learning
Very small number of positive examples ($y = 1$). (0-20 is common). Large number of negative ($y = 0$) examples. $p(x)$		Large number of positive and negative examples.
Many different "types" of anomalies. Hard for any algorithm to learn from positive examples what the anomalies look like; future anomalies may look nothing like any of the anomalous examples we've seen so far.		Enough positive examples for algorithm to get a sense of what positive examples are like, future positive examples likely to be similar to ones in training set.
Fraud detection $y=1$		• Email spam classification (Spam)
Manufacturing (e.g. aircraft engines)		• Weather prediction (sunny/rainy/etc).
Monitoring machines in a data center		• Cancer classification
⋮		⋮

3. Choosing what features to use

Week 9

Anomaly Detection

二. Predicting Movie Ratings

1. Content Based Recommendations 电影推荐.

$r(i,j) = 1$, if user j has rated movie i (0 otherwise)

$y^{(i,j)}$ = rating by user j on movie i (if defined)

$\theta^{(j)}$ = parameter vetor for user j.

$x^{(i)}$ = feature vector for movie i

For user j, movie i, predicted rating: $(\theta^{(j)})^T (x^{(i)})$

$m^{(j)}$ = no. of movies rated by user j.

To learn $\theta^{(j)}$:

Optimization objective:

To learn $\theta^{(j)}$ (parameter for user j):

$$\rightarrow \min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

To learn $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(n_u)}$:

$$\min_{\theta^{(1)},\ldots,\theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

Gradient descent update:

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} \quad (\text{for } k = 0)$$

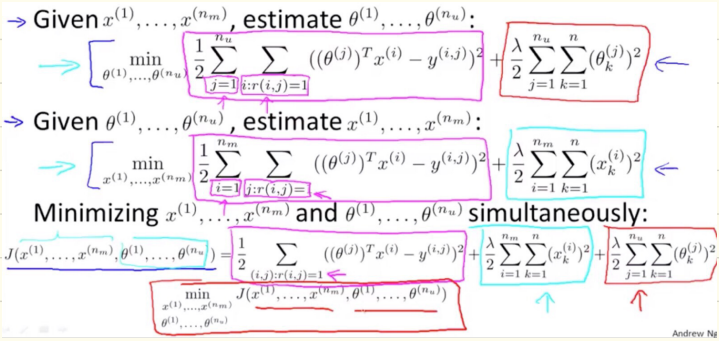
$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right) \quad (\text{for } k \neq 0)$$

三. Collaborative Filtering 协同过滤

$x^{(1)}, \ldots, x^{(n_m)}$ can estimate $\theta^{(1)}, \ldots, \theta^{(n_u)}$

Week 9

Anomaly Detection



协同过滤算法.

<1> 初始化 $x^{(1)},\dots,x^{(n_m)}, \theta^{(1)},\dots,\theta^{(n_u)}$ ⇒ 小的·随机值

<2> minimize $J(x^{(1)},\dots,x^{(n_m)}, \theta^{(1)},\dots,\theta^{(n_u)})$ (梯度下降)

<3> 对 user, θ, movie, feature x.

使用 $\theta^T x$ 规则评分

四. Low Rank Matrix Factorization

1. Vectorization

协同过滤: 相似度算法. (基于同一类别的物品)

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

Predicted ratings:

$$\begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & (\theta^{(2)})^T(x^{(1)}) & \dots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \dots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \dots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

2. Implementational Detail: Mean Normalization

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.25 & -2.25 & 2.75 & 1.75 & ? \\ -1.25 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

learn $\theta^{(j)}$, $x^{(i)}$

For user j, on movie i predict: $(\theta^{(j)})^T(x^{(i)}) + \mu_i$

Week 10

Large Scale Machine Learning

一. Gradient Descent with Large Datasets

plot $J_{train}(\theta)$ 和 $J_{cv}(\theta)$ 判断 m 对 bias/variance 的影响

1. Stochastic Gradient Descent

m 很大时：

$$cost(\theta, (x^{(i)}, y^{(i)})) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{train}(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost(\theta, (x^{(i)}, y^{(i)}))$$

Steps:

1. Randomly shuffle (reorder) training examples
2. Repeat {
   for $i := 1, \dots, m$ {
   $\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$
   (for every $j = 0, \dots, n$)
   }
   }

2. Mini-batch Gradient Descent

Batch gradient descent: Use all m examples in each iteration

Stochastic gradient descent: Use 1 example in each iteration

Mini-batch gradient descent: Use b examples in each iteration

相当于 $b = m$

For example, $b = 10$

$$\theta_j := \theta_j - \alpha\frac{1}{10}\sum_{k=i}^{i+9}(h_\theta(x^{(k)}) - y^{(k)})x_j^{(k)}$$

$i := i + 10$

Algorithm

Say $b = 10, m = 1000$.

Repeat {

for $i = 1, 11, 21, 31, \dots, 991$ {

$$\theta_j := \theta_j - \alpha\frac{1}{10}\sum_{k=i}^{i+9}(h_\theta(x^{(k)}) - y^{(k)})x_j^{(k)}$$

(for every $j = 0, \dots, n$)

}

}

Week 10

Large Scale Machine Learning

3. Stochastic Gradient Descent Convergence

Check for convergence

Batch gradient descent:

→ Plot $J_{train}(\theta)$ as a function of the number of iterations of gradient descent.

→ $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$ m = 300,000,000

Stochastic gradient descent:

→ $cost(\theta, (x^{(i)}, y^{(i)})) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$ → $(x^{(i)}, y^{(i)})$, $(x^{(i+1)}, y^{(i+1)})$ ...

→ During learning, compute $cost(\theta, (x^{(i)}, y^{(i)}))$ before updating θ using $(x^{(i)}, y^{(i)})$.

→ Every 1000 iterations (say), plot $cost(\theta, (x^{(i)}, y^{(i)}))$ averaged over the last 1000 examples processed by algorithm.

α is typically held constant. Can slowly decrease α over time if we want θ to converge.

(e.g. $\alpha = \frac{const\ 1}{iteration\ Number + const\ 2}$)

二. Advanced Topics

1. Online Learning

Example:

Product Search / Recommendation. ;

Choosing special offers to user ;

Customized selection of news articles

2. Map-Reduce and Data Parallelism

并行处理 · 数据并行

Map-Reduce

Batch gradient descent: $\theta_j := \theta_j - \alpha \frac{1}{400}\sum_{i=1}^{400}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$ m=400

Machine 1 : 1 ~ 100

Machine 2 : 101 ~ 201

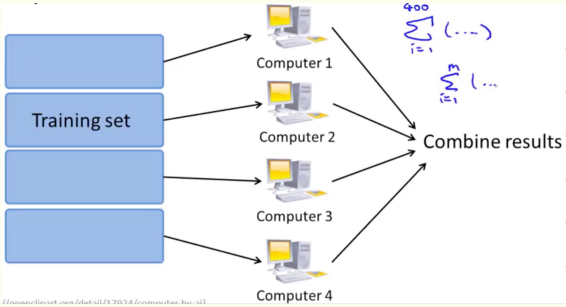
⋮

4 : 301 ~ 400

} add together

Week 10

Large Scale Machine Learning



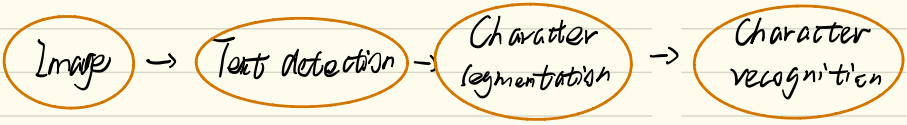
Idea: Many learning algorithm can be expressed as computing sum of functions over the training set.

Week 11

App Example: Photo OCR

—. Photo OCR

1. Photo OCR Pipeline



2. Sliding Windows

1. Text detection

2. Character segmentation

3. Character classification

3. Getting Lots of Data and Artificial Data

Discussion on getting more data

1. Make sure you have a low bias classifier before expending the effort. (Plot learning curves). E.g. keep increasing the number of features/number of hidden units in neural network until you have a low bias classifier.
2. "How much work would it be to get 10x as much data as we currently have?"
   - Artificial data synthesis
   - Collect/label it yourself
   - "Crowd source" (E.g. Amazon Mechanical Turk)

Summary

Main topics

Supervised Learning
- Linear regression, logistic regression, neural networks, SVMs

Unsupervised Learning
- K-means, PCA, Anomaly detection

Special applications/special topics
- Recommender systems, large scale machine learning.

Advice on building a machine learning system
- Bias/variance, regularization; deciding what to work on next: evaluation of learning algorithms, learning curves, error analysis, ceiling analysis.