L8 Advice for applying machine learning

8-1 Deciding what to try next

- How to debugging a learning algorithm
  - Get more training examples
  - Try smaller sets of features
  - Try getting additional features
  - Try adding polynomial features (x_1, x_2, etc.)
  - Try decreasing or increasing λ

(但如何选择?)

8-2 Evaluating a hypothesis 评估假设

- 评估假设的一种常用方式

将 "Dataset" 分割为 { Training set (70%); Test set (30%) } [Note: 需要随机分组]

- Training/testing procedure for linear regression
  - Learn parameter θ from training set
  - Compute test set error:

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

- ... for logistic regression
  - Learn parameter θ from training set
  - Compute test set error:

$$J_{test}(\theta) = -\frac{1}{m_{test}} \left[\sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log (1 - h_\theta(x_{test}^{(i)})) \right]$$

(0/1 misclassification error):

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

$$\text{Test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} error(h_\theta(x_{test}^{(i)}), y^{(i)})$$

验证

8.3 Model selection and training / validation / test set

- 为了检验模型在新样本上的泛化能力

⇒ 将"Dataset"分为三部分
- Training set (60%)
- Cross validation (20%) 交叉验证集
- Test set (20%)

[确定θ参数] Training error 即代价函数

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

[选择模型] · Cross validation error $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$

[测试泛化能力] · Test error $J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$

e.g.

Model selecting

$d=1$ 1. $h_\theta(x) = \theta_0 + \theta_1 x \xrightarrow{\min J(\theta)_{train}} \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$

$d=2$ 2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$

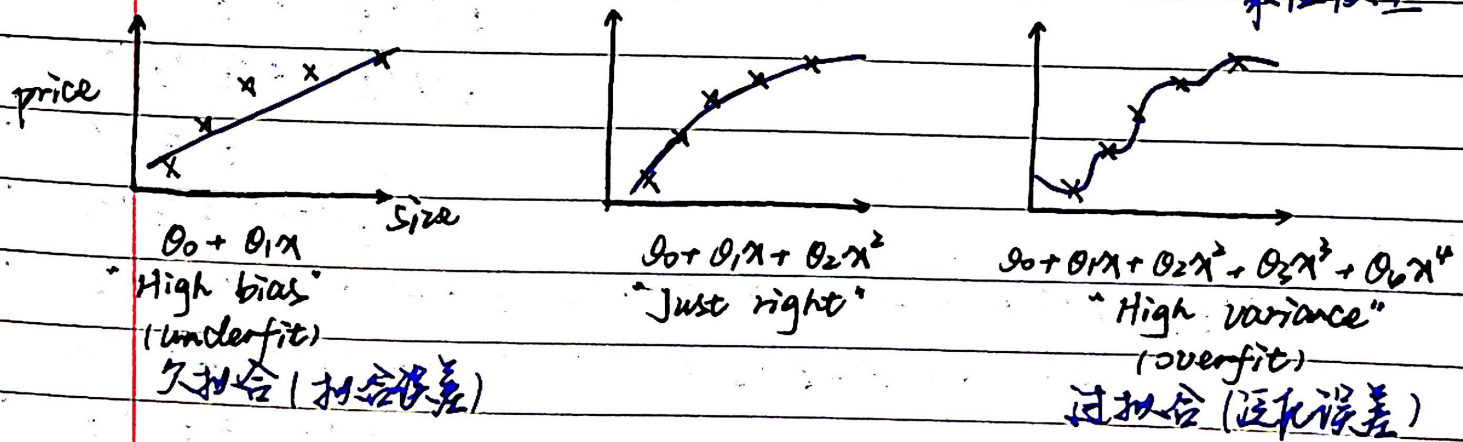
⋮ ⋮

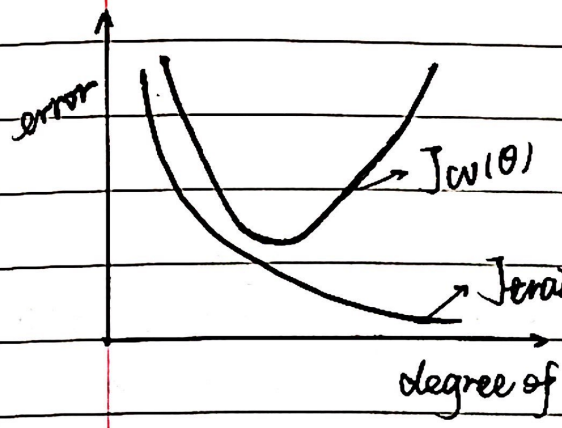
$d=10$ 10. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10} \rightarrow \theta^{(10)} \rightarrow J_{cv}(\theta^{(10)})$

} 选择 $\min J_{cv}(\theta^{(i)})$ 作为最佳模型

⇒ Estimate generalization error for test set $J_{test}(\theta^{(i)})$ ← 最佳模型

8.4 Diagnosing bias vs. variance 诊断偏差和方差





Bias: $J_{train}(\theta)$ will be high; $J_{cv}(\theta) \approx J_{train}(\theta)$

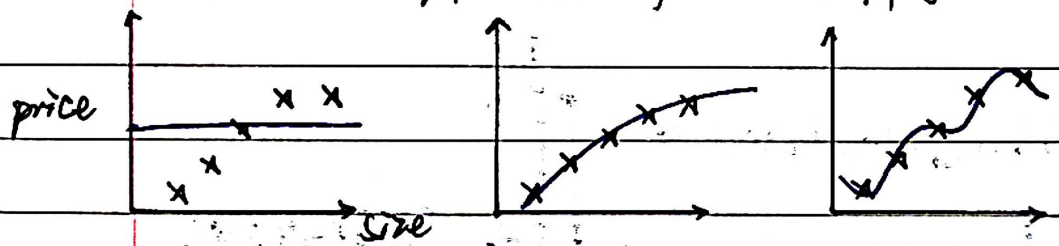
Variance: $J_{train}(\theta)$ will be low; $J_{cv}(\theta) \gg J_{train}(\theta)$

8.5 Regularization and bias/variance

- linear regression with regularization

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

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



large λ → High bias

$\lambda = 10000$, $\theta_0 \approx 0$, $\theta_1 \approx 0$

$h_\theta(x) = \theta_0$

Intermediate λ

Small λ → High variance → $\lambda = 0$

如何选择 λ 大小？ → 训练时才有正则项

1. Try $\lambda = 0$ → $\min J(\theta)$ → $\theta^{(1)}$ → $J_{cv}(\theta^{(1)})$
2. Try $\lambda = 0.01$ → $\theta^{(2)}$ → $J_{cv}(\theta^{(2)})$
3. Try $\lambda = 0.02$ → Pick $\theta^{(5)}$ (最佳假设)

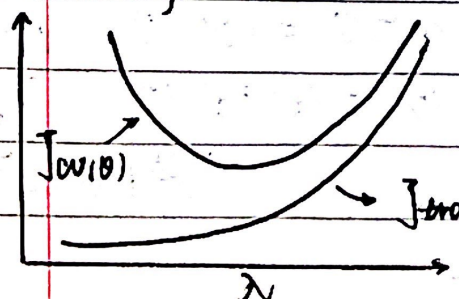
⋮ ⋮

12. Try $\lambda = 10.24$ → $\theta^{(12)}$ → $J_{cv}(\theta^{(12)})$

Test error:

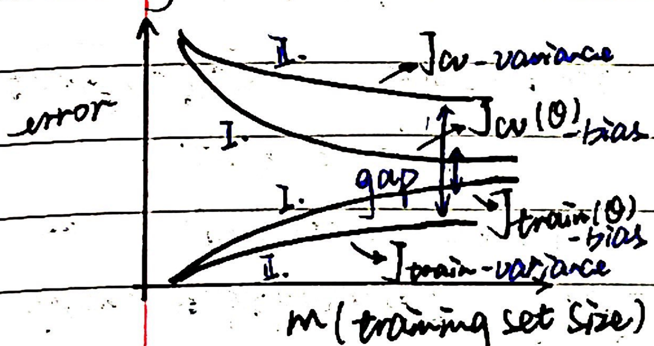
$J_{test}(\theta)$

$$= \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2 \qquad = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$



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

8-6 learning Curves 学习曲线 [模型一定] High bias / High variance



训练集越大，越难拟合 → $J_{train}(\theta)$ ↑

但同时模型的泛化能力会越好 → $J_{cv}(\theta)$ ↑

⇒ 说明收集更多训练数据对改善算法无益 ↑

I. 对于高偏差模型而言 m↑，gap没有↓空间

gap大小取决于是高偏差(小)还是高方差(大)

II. 但对于高方差模型而言，m继续增大，gap会相对减小 ⇒ 使用更多训练数据对改进算法有帮助

8-7 总结

- Get more training example → 适用于High variance
  通过绘制学习曲线来判断，i.e. $J_{cv}(\theta) \gg J_{train}(\theta)$
- Try ~~more~~ smaller sets of features → 适用于High variance
- Try getting additional features } 适用于High bias
- Try adding polynomial features } 适用于High bias
- Try decreasing λ → 适用于High bias
- Try increasing λ → 适用于High variance

对于神经网络来说

- "small" Neutral Network { 神经元个数 / Hiden layer 较少，易出现欠拟合，但计算量小
- "large" Neural Network，…较多，易出现过拟合，计算量大，但上限更高
  → Use Regularization (λ)

2.9 Machine learning system design

9-1 Prioritizing what to work on 确定执行时的先后

E.g. Building a spam classifier

Supervised learning. x = features of email. y = spam(1) or not spam(0).

Feature x: Choose 100 words indicative of spam/not spam.

like A. B. C. D ... 100维向量 x input = $[0\ 1\ 1\ 0\ 0 \cdots 1 \cdots]^T$

→ How to make it have low error? (邮件出现为1. 未出现为0)

- Collect lots of data?

$x_j = \begin{cases} 1, & \text{if word } j \text{ appears in email} \\ 0, & \text{otherwise} \end{cases}$

- Develop sophisticated features based on email routing information (from email header)
- Develop sophisticated features for message body.

e.g. Should 'discount' and 'discounts' be treated as the same word?

- Develop sophisticated algorithm to detect misspellings

e.g. medium, w4tches

9-2 Error analysis

- Reccomended approach
  - Start with a simple algorithm that you can implement quickly. Implement it and test it on your cross-validation data.
  - Plot learning curves to decide if more data, more features, etc are likely to help.
  - Error analysis: Manually examine the examples (in cross validation set) that your algorithm made errors on.

  ⇒ 从而启发如何改进算法(新特征?)

- The importance of numerical evaluation e.g 区分大小写 数值评估 不区分

  ⇒ 选择不区分

错误率 5.4% 2.7%

[倾斜类]

6.3 Error metrics for skewed classes 不对称性分类的误差评估

- Cancer classification example

Train logistic algorithm model $h_\theta(x)$ to find that you got 1% error on the test. but only 0.5% of patients have cancer.

(为解决倾斜偏差问题) → Precision / Recall [精确度和召回率] 评价指标

6.9

		Actual class 1	Actual class 0
Predicted class	1	True positive	False positive
	0	False negative	True negative

- Precision $= \frac{\text{True Positive}}{\#\text{predicted Positive}} = \frac{TP}{TP+FP}$ (详见AI note)
- Recall $= \frac{\text{True positive}}{\#\text{Actual Positive}} = \frac{TP}{TP+FN}$

6.4 Trading off precision and recall (权衡)

- Cancer classification example

- logistic regression : $0 \le h_\theta(x) \le 1$ { predict 1 if $h_\theta(x) \ge 0.5$; 0 if $h_\theta(x) < 0.5$

Suppose we want to predict $y=1$ (cancer only if very confident)

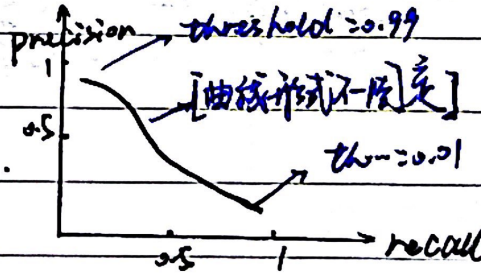
→ 改进阈值 like { 1 if $h_\theta(x) \ge 0.9$; 0 if $h_\theta(x) < 0.9$ ⇒ higher precision. lower recall.

Suppose we don't want to avoid missing too many cases of cancer

→ like { 1 if $h_\theta(x) \ge 0.3$; 0 if < 0.3 ⇒ higher recall. lower precision

More generally, predict $y=1$ if $h_\theta(x) \ge$ threshold

对于大多数分类器，需要权衡 precision 和 recall.



How to compare?

→ F_1 Score $= \frac{2PR}{P+R}$

	Precision	Recall	F_1 Score
Algorithm 1	0.5	0.4	0.664
2	0.7	0.1	0.175
3	0.02	1.0	0.0392

9-5 Data for machine learning

- When feature $x \in \mathbb{R}^{n+1}$ has sufficient information to predict y accurately → 使用大量数据才有意义
- Large data rationale
  - Use a learning algorithm with many parameters (e.g. logistic regression / linear regression with many features; neural network with many hidden units). → 保证低偏差
  - Use a large training set (足够大, ⇒ parameters ⇒ unlikely to overfit) → 保证低方差

10 Support Vector Machines 支持向量机

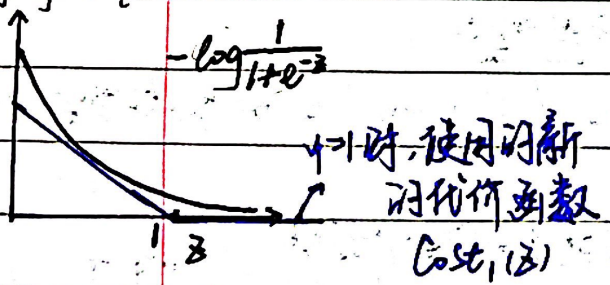
10-1 Optimization objective 优化目标

- Alternative view of logistic regression

Cost of example: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

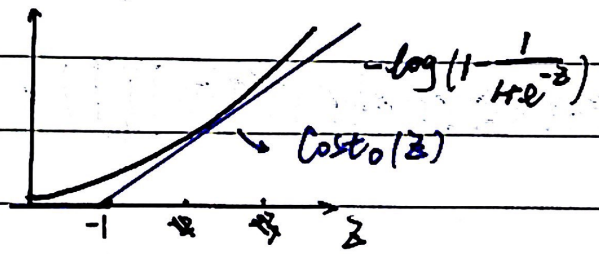
$= -y \log \frac{1}{1+e^{-\theta^T x}} - (1-y)\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)$

- 若 $y=1$ (want $\theta^T x \gg 0$)



$y=1$ 时，使用对新的代价函数 $\text{Cost}_1(z)$

- 若 $y=0$ (want $\theta^T x \ll 0$)



- Support vector machine 对总体优化对象

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

↓

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

(constant)

↓

假设函数形式 $h_\theta(x)$ { 1, 若 $\theta^T x \geq 0$; 0, otherwise } 并不会输出概率

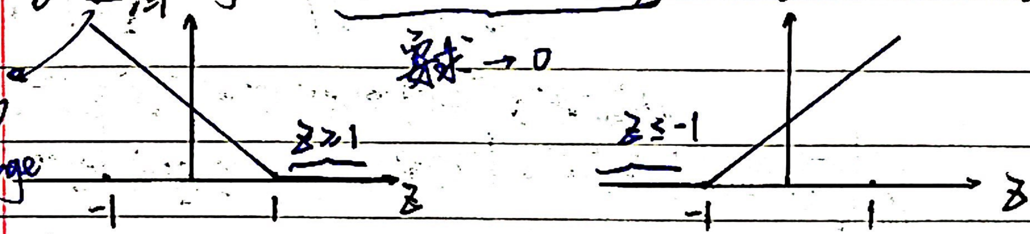
对大间隔的直观理解

10-2 large Margin Intuition（SVM有时被称为大间距分类器）

$$\min_{\theta} C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

要求 → 0

C设置为 very large



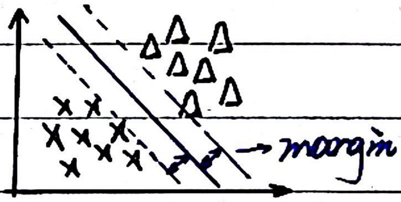
if $y=1$, want $\theta^T x \geq 1$ (not just ≥ 0)

if $y=0$, want $\theta^T x \leq -1$ (not just < 0)

} 存在"安全间隔"

SVM的决策边界会使margin更大 ⇐

会让SVM具有鲁棒性



- C 值设置越大，对离群点（异常点）越敏感（⇒ 可以让C小一点）

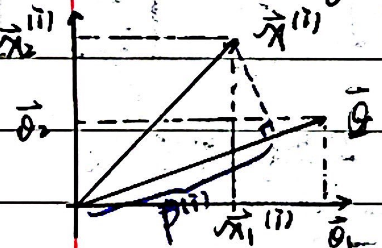
10-3 大间隔分类器的数学原理

- 向量积 $\min_{\theta}\frac{1}{2}\sum_{j=1}^{n}\theta_j^2$ → 看作 $\vec{\theta_j} = \vec{\theta_{1j}} + \vec{\theta_{2j}}$ → $\frac{1}{2}\|\theta\|^2$（向量长度/范数）

s.t. $\theta^T x^{(i)} \geq 1$ if $y^{(i)}=1$

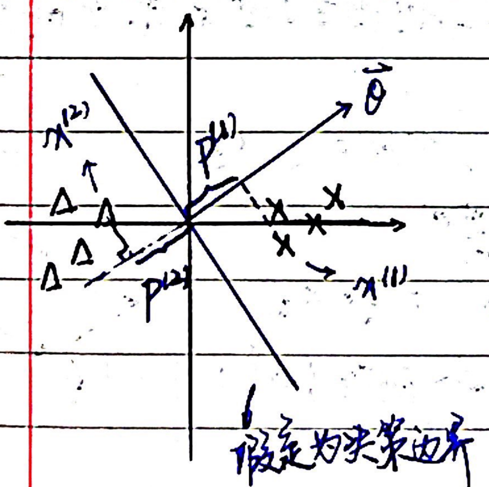
$\theta^T x^{(i)} \leq -1$ if $y^{(i)}=0$

假设 $\theta_0=0$, $n=2$



$$\theta^T x^{(i)} = p^{(i)}\|\theta\| = \vec{\theta_1}\cdot\vec{x_1}^{(i)} + \vec{\theta_2}\vec{x_2}^{(i)}$$

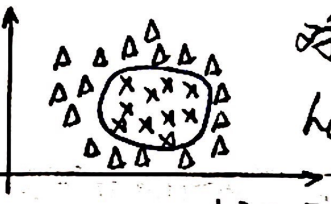
- 参数向量θ事实上是与决策边界垂直的 ⇒ $\theta_1 x_1 + \theta_2 x_2 = 0$



s.t. $p^{(i)}\cdot\|\theta\| \geq 1$, if $y^{(i)}=1$

$p^{(i)}\cdot\|\theta\| \leq -1$, if $y^{(i)}=0$

to $\min_{\theta}\frac{1}{2}\sum_{j=1}^{n}\theta_j^2$ → to make $p^{(i)}$ larger

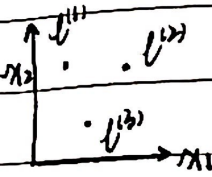


对 Non-linear 问题 $h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0 \\ 0 & \text{otherwise} \end{cases}$

10-4 Kernels I ⇒ 相比于高阶项，是否有更好的特征选择？

选取 · 建立新特征 (Given x, compute new feature depending on proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$) 实际可选择更多

手动?



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

$f_2 = \text{Similarity}(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$

$f_3 = \cdots \rightarrow k(x, l^{(i)})$ ↓ Gaussian kernels

若 $x \approx l^{(1)}$: $f_1 \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$ · σ^2 决定了特征变量减小速度

若 x is far from $l^{(1)}$: $f_1 \approx 0$ 。 σ^2 越小，减小越快。

△ 通过标记点和相似性函数定义新的特征变量 → 训练复杂的非线性边界

10-5 Kernels II · Where / How to get $l^{(1)}, l^{(2)}, l^{(3)}, \cdots$? (选择样本点)

→ Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})$

→ Choose $x^{(1)} = l^{(1)}, x^{(2)} = l^{(2)}, \cdots, x^{(m)} = l^{(m)}$

· Given example x: $f_1 = \text{Similarity}(x, l^{(1)})$, $f_2 = \text{Similarity}(x, l^{(2)}) \cdots$

→ $f = [f_0, f_1, \cdots, f_m]^T$, $f_0 = 1$ $f_m = \text{Sim}(x, l^{(m)})$

· For training example $(x^{(i)}, y^{(i)})$ $x^{(i)} \in \mathbb{R}^{n+1}$

$x^{(i)} \Rightarrow$ $f_1^{(i)} = \text{Sim}(x^{(i)}, l^{(1)})$ ⋮ $f_m^{(i)} = \text{Sim}(x^{(i)}, l^{(m)})$ ⇒ $f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix}$, $f_0^{(i)} = 1$

Hypothesis: Given x, compute feature $f \in \mathbb{R}^{m+1}$

Predict "$y=1$" if $\theta^T f \geq 0$ ($\theta_0 f_0 + \theta_1 f_1 + \theta_2 f_2 + \cdots + \theta_m f_m$)

⇒ $\min_\theta C \sum_{i=1}^{m} y^{(i)} \text{Cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{Cost}_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{m(=n)} \theta_j^2$ → θ 的维度

参数选择 · C ($= \frac{1}{\lambda}$) { large C: lower bias, higher variance; small C: higher bias, lower variance }

· σ^2 { large σ^2: Feature f_i vary more smoothly. higher bias, lower variance; small σ^2: f_i vary less smoothly. lower bias, higher variance. }

10-6 Using an SVM

- Use SVM software package (e.g. Sklearn:) to solve for parameters θ.

Need to specify: } choice of parameter C
choice of kernel (similarity function)

(两个常用核函数)

E.g. · No kernel ("linear kernel")

predict "y=1" if $\theta^T x \geq 0 \rightarrow \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \geq 0$, $x \in \mathbb{R}^{n+1}$

· Gaussian kernel

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

Note: Do perform [feature scaling] before using the Gaussian kernel.

因为 $\|x - l^{(i)}\|^2 \rightarrow (x_1 - l_1)^2 + (x_2 - l_2)^2 + \cdots (x_n - l_n)^2$ 数量

不同特征代表因数值差异可能会很大. 例如 x_1 代表房子面积, x_2 代表房间

· 对于Multi-class classification (i.e. kSVM)

原理和"1-1"一致. one to distinguish $y=i$ from the rest, for $i = 1, 2, \cdots K$.

get $\theta^{(1)} \rightarrow (y=1)$, $\theta^{(2)} \rightarrow (y=2)$, $\cdots$ $\theta^{(k)}$; pick class i with largest $(\theta^{(i)})^T x$

· 逻辑回归 vs. SVM (n表示特征, m表示训练数据)

· if n is large (relative to m), e.g. $n \geq 10000$, $m = 10 \sim 1000$

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

· if n is small, m is intermediate, e.g. $n = 1 \sim 1000$, $m = 10 \sim 10000$

Use SVM with Gaussian kernel

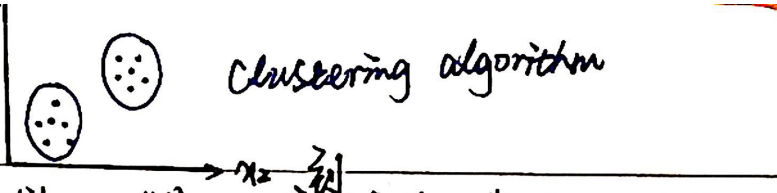
· if n is small, m is large, e.g. $n = 1 \sim 1000$, $m = 50000+$

Add/create more feature, then use logistic regression or SVM without a kernel.

△ Neural network likely to work well for most settings, (these)

but may be slower to train.

11 Clustering 聚类 → 无监督



11-1 Unsupervised learning Introduction

- 无标签，train set：$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ ⟹ 找数据中的结构
- 应用：见 11-3

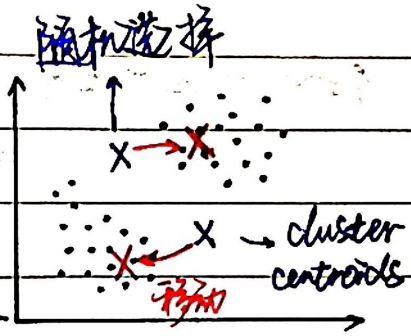
11-2 K-means Algorithm

- K-means 是一个迭代算法 ⇒ { 簇分类(分配) / 移动聚类中心

每次内循环第一步：进行簇分配，根据与两个中心哪个更近来将样本点分配给两个聚类中心之一（遍历每个点）

第二步：移动聚类中心，将两个中心分别移动到其分配点的均值处

然后进行迭代



- Algorithm

Inputs: K (number of clusters)

Training set $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention)

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

Repeat {

for $i = 1$ to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$ （样本点所属簇类）

for $k = 1$ to K

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

}

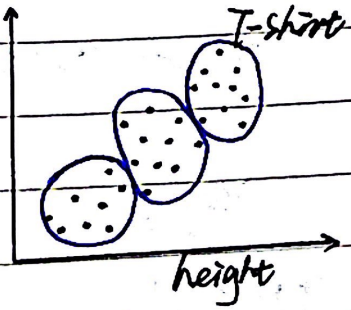
$\min_k \|x^{(i)} - \mu_k\|^2$ 将 k 赋给 $c^{(i)}$

$c^{(i)} = k$

e.g. $x^{(1)}, x^{(5)}, x^{(6)}, x^{(10)} \rightarrow c^{(1)} = 2, c^{(5)} = 2, c^{(6)} = 2, c^{(10)} = 2$

$\Rightarrow \mu_2 = \frac{1}{4}[x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}] \in \mathbb{R}^n$

- K-means 对于 non-separated clusters 也有效



(假设 $x^{(i)}$ 被划分为第5个簇)

e.g. $x^{(i)} \to 5$, 即 $c^{(i)}=5$, $\Rightarrow \mu_{c^{(i)}} = \mu_5$

11-3 Optimization objective

(回顾11-2) · $c^{(i)} = \dots$, $\mu_k = \dots$, $\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned.

优化目标函数: (Distortion, 失真代价函数)

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

$$\min_{\substack{c^{(1)},\dots,c^{(m)} \\ \mu_1,\dots,\mu_K}} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

· 实际上, 11-2中 { 第一步(分配)就是在 minimize J w.r.t. $c^{(1)}, \dots, c^{(m)}$.
第二步(移动)就是在 minimize J w.r.t $\mu_1, \dots, \mu_K$.

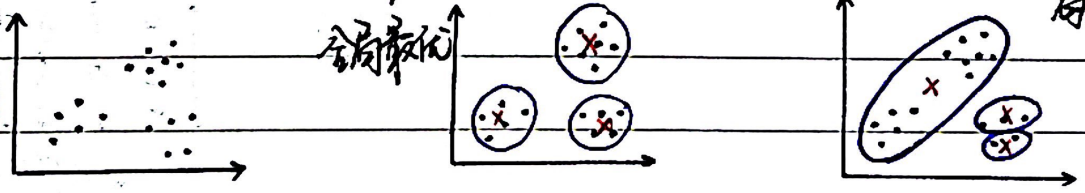
11-4 Random initialization 随机初始化

· (选择样本点作为聚类中心)

Should have $K < m$, randomly pick K training examples.

Set $\mu_1, \dots, \mu_K$ equal to these K examples. ($\mu_1 = x^{(i)}$, $\mu_2 = x^{(j)}$, $\dots$)

· 对于局部最优 (local optima)

e.g. 

⇒ 对 K-means Algorithm 进行多次初始化 (对于K值相对较小的情况 例如 $K=2\sim10$)

For $i=1$ to 100 {

Randomly initialize K-means.

Run K-means. Get $c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K$.

Compute cost function (distortion) → $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

}

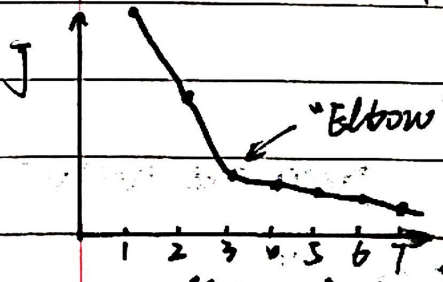
Pick clustering that gave lowest cost $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$

理论上，因为无标签 ⇒ 无法设计Algorithm自动选择K：答案可能不止一个.

一般都是手动选择 ⇒ 观察可视化的图；观察聚类算法的输出. etc.

11-5 Choosing the number of cluster

- Elbow method (肘部法则)



(但大多数情况，并不会出现拐点.)

"Elbow"：表示出现拐点，K>3时曲线下降缓慢

意味着可选择分为3簇

(e.g. 市场分割)

- 下游目的，i.e. 看哪个聚类数量能更好地应用于后续目的

12 Dimensionality Reduction 降维

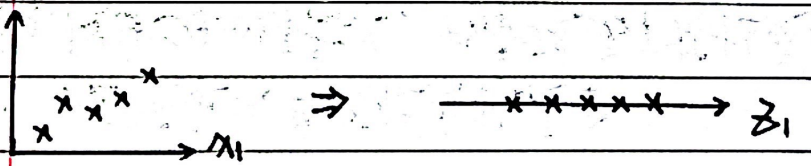
12-1 Motivation I: Data Compression 目标1：数据压缩

⇒ 让数据占用较少的内存或硬盘空间，[如果特征高度相关 → 高冗余度]

对学习算法进行加速

E.g. Reduce data from 2D to 1D

$x^{(1)} \in \mathbb{R}^2 \to z^{(1)} \in \mathbb{R}$ ... $x^{(m)} \in \mathbb{R}^2 \to z^{(m)} \in \mathbb{R}$



Reduce data from 3D to 2D, i.e. 立体转化至平面

12-2 Motivation II: Data Visualization (可视化)

E.g. 各国家特征 (x_1 = GDP, x_2 = Poverty index, x_3 = life expectancy ... $x \in \mathbb{R}^{50}$) ⇒ 将x降维到2维 ⇒ z_1 表示-GDP总值，z_2 表示人均GDP

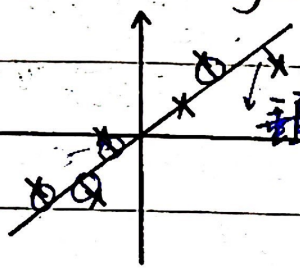
从而成可视化.

(PCA：处理降维问题的算法)

12-3 Principal Component Analysis problem formulation 主成分分析问题规划：

- PCA → Find k vectors $u^{(1)}, \dots, u^{(k)}$，将数据投影到这k个向量所构成的[Reduced from n-dimension to k-dimension]线性空间上，并最小化投影误差
- PCA和linear regression是不同的算法.

数据点到空间的距离



e.g. 二维 → 一维

垂直距离 → 投影距离

12-4 主成分分析问题规划2

- Data preprocessing：Training set：$x^{(1)}, x^{(2)}, \dots, x^{(m)}$

Preprocessing (feature scale /[mean normalization])

$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$ ← Replace each $x_j^{(i)}$ with $x_j - \mu_j$

If different features on different scales (e.g. x_1 = size of house, x_2 = number of bedrooms), scale features to have comparable range of values

- Algorithm：(Reduce data from n-d (dimension) to k-d) n维 → k维

Compute "Covariance matrix"(协方差) → $\Sigma = \frac{1}{m}\sum_{i=1}^{n} (x^{(i)})(x^{(i)})^T$

Compute "eigenvectors" of matrix Σ：[U, S, V] = svd(Sigma)

特征向量

To get：

$U = [u^{(1)} \dots u^{(n)}] \in \mathbb{R}^{n \times n}$ 获取前k个列 $u^{(1)} \to u^{(k)}$

$$z = \underbrace{\begin{bmatrix} | & & | \\ u^{(1)} & \cdots & u^{(k)} \\ | & & | \end{bmatrix}^T}_{U_{reduce}} x = \underbrace{\begin{bmatrix} - & u^{(1)T} & - \\ & \vdots & \\ - & u^{(kT)} & - \end{bmatrix}}_{k \times n} \underbrace{x}_{n \times 1}$$

$n \times k$

$k \times 1$ i.e. z是k维向量

主成分数量选择

12-5 Choosing the number of principal components (K)

- Algorithm: Try PCA with $k=1$ (2, … 迭代直到满足条件)

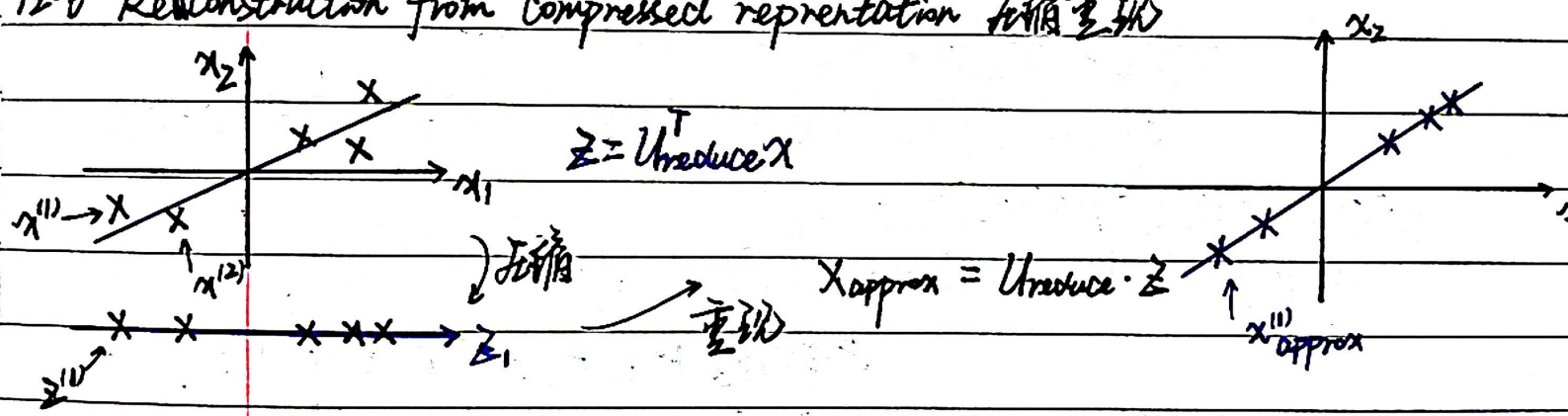
  Compute $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x^{(1)}_{approx}$(投影), …, $x^{(m)}_{approx}$

  Check if $\frac{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}-x^{(i)}_{approx}\|^2}{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}\|^2} \le 0.01$ (= 99% of variance is retained)

- $[U,S,V] = svd(Sigma)$ to get $S=\begin{bmatrix} S_{11} & & 0 \\ & S_{22} & \\ 0 & & \ddots \\ & & & S_{nn} \end{bmatrix}_{n\times n}$

  (for given k) pick smallest of value k for which $\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \ge 0.99$

12-6 Reconstruction from compressed representation 压缩重现



12-7 Advice for applying PCA

- Design
  - Get training set $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$
  - Run PCA to reduce $x^{(i)}$ in dimension to get $z^{(i)}$
  - Train logistic regression on $\{(z^{(1)}, y^{(1)}), \dots, (z^{(m)}, y^{(m)})\}$
  - Test on test set: [Map $x^{(i)}_{test}$ to $z^{(i)}_{test}$]. Run $h_\theta(z)$ on $\{(z^{(1)}_{test}, y^{(1)}_{test}), \dots, (z^{(m)}_{test}, y^{(m)}_{test})\}$ → 此处用映射关系利用用是PCA在训练集中获得的参数

- Bad use of PCA: To prevent overfitting

  ↓

  [减正则化]