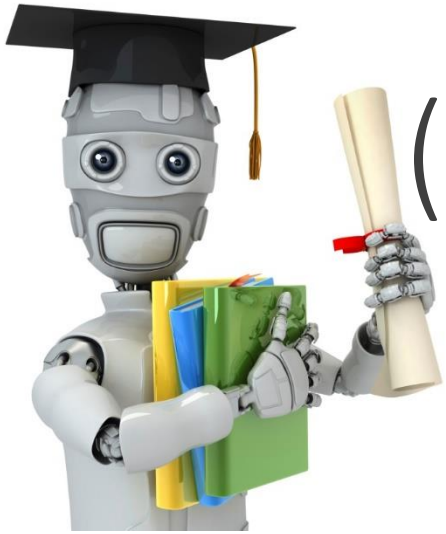


系统设计与应用 (System Structuring & Applications)

Machine Learning

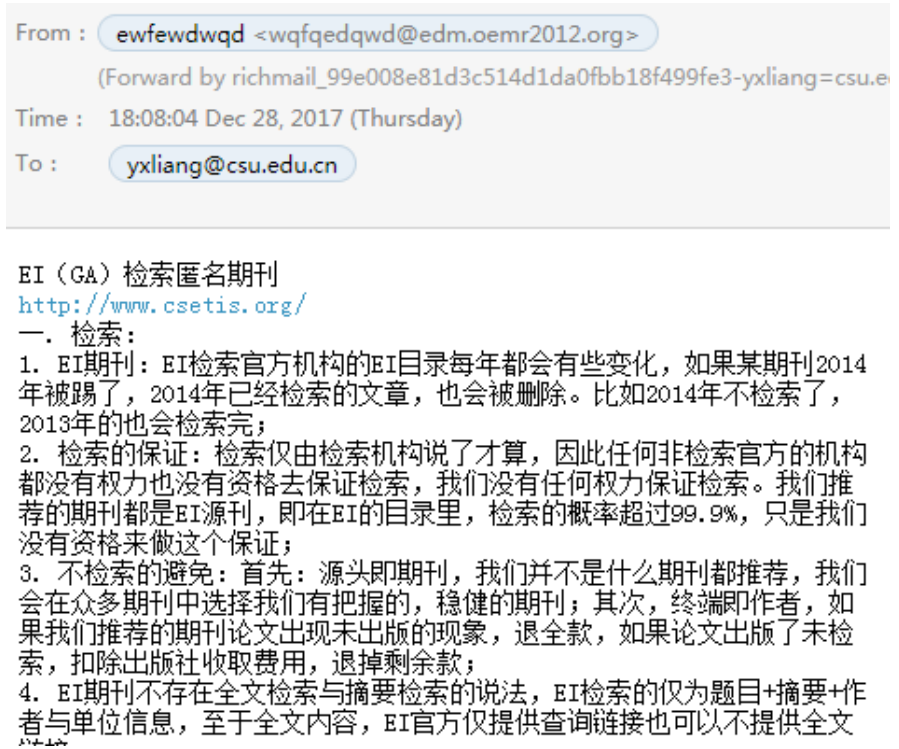
梁毅雄

yxliang@csu.edu.cn

Some materials from Andrew Ng, Hongyi Lee and others

机器学习系统设计

垃圾邮件分类器:



机器学习系统设计

垃圾邮件分类问题属于监督学习典型案例：

- x =邮件文本特征, $y = 1$(垃圾邮件)或者$y = 0$（正常邮件）
- 特征x: Bag of Words (BoW), 对应特征的维度等于words的个数, 若邮件文本中存在该词，则对应的分量为1，否则为0

$$x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- 如何选择words? 一种简单的方案是选择5000-10000个常用字。

机器学习系统设计

如何提高垃圾邮件分类器的准确率？

- 收集更多的数据
- 设计设计更好的特征表示x方法
- 同源同义词合并、拼写校正等
- ...

Chain of assumptions in ML

- Fit training set well on cost function
- Fit validation set well on cost function
- Fit test set well on cost function
- Perform well in real world

错误分析

假设验证集有500个样本，算法准确率仅80%, 即存在100个错分样本，此时可以对这些错分的样本进行详细的分析，如

- 这些错分样本的类型
- 能找到那些方法或者线索（特征）可以减少错误的分类？
- ...

错误分析

Incorrectly labeled examples

x 

y 1 0 1 1 0 1 1

Image	Dog	Great Cat	Blurry	Incorrectly labeled	Comments
…					
98				✓	Labeler missed cat in background
99		✓			
100				✓	Drawing of a cat; Not a real cat.
% of total	8%	43%	61%	6%	

算法评估

- 尽量采用单值评估指标：如平均准确率/错误率等

Algorithm	US	China	India	Other
A	3%	7%	5%	9%
B	5%	6%	5%	10%
C	2%	3%	4%	5%
D	5%	8%	7%	2%
E	4%	5%	2%	4%
F	7%	11%	8%	12%

算法评估

- 对于癌症分类这类样本分布极为不平衡的二分类问题，是否仍然采用（平均）准确率进行评估？
  - 假设采用Logistic Regression来处理癌症分类，发现测试集上的错误率为1%
  - 但若实际癌症只有0.5%，若一简单分类器把所有的样本都判别为正常，则错误率只有0.5%，是否该选择这一简单分类器？

算法评估

$$precision = \frac{TP}{\hat{P}} = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{P} = \frac{TP}{TP + FN}$$

$$specificity = \frac{TN}{N} = \frac{TN}{FP + TN}$$

$$accuracy = \frac{TP + TN}{P + N}$$

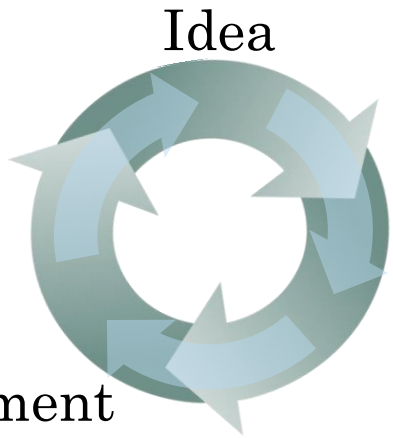
		y		
		1	0	
$\hat{y}$	1	TP	FP	$\hat{P}$
	0	FN	TN	$\hat{N}$
		P	N	

Table 1: Confusion Matrix

算法评估

$$precision = PPV = \frac{TP}{\hat{P}} = \frac{TP}{TP + FP}$$

$$recall = TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$



Classifier	Precision	Recall
A	95%	90%
B	98%	85%

$$F_1 Score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

算法评估

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms

$$\max(\texttt{Accuracy} - \lambda\, \texttt{Running time})$$

$$\max\ \texttt{Accuracy}$$

$$s.t.\ \texttt{Running time} \leq 100ms$$

训练集、验证集与测试集

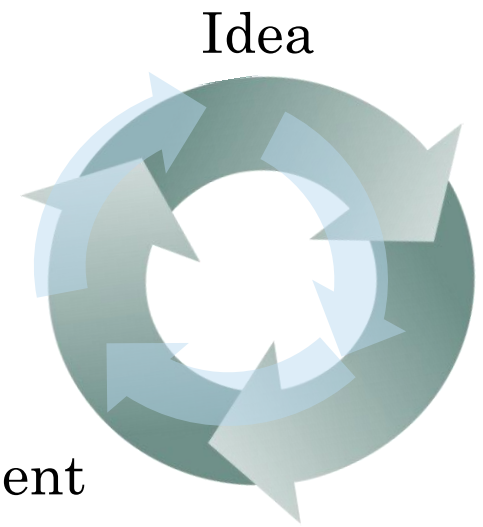
Regions:

- US
- UK
- Other Europe
- South America

验证集

- India
- China
- Other Asia
- Australia

测试集



训练集、验证集

Data from webpages

Data from mobile app

如何选择训练集与验证集?

验证集与测试集

- 如何选择训练集、验证集和测试集？
- 要确保你在训练集上的训练、在验证集上调参、模型选择等工作真正发挥了作用
- Choose a validation set and test set to reflect data you expect to get in the future and consider important to do well on.

快速迭代: Speech recognition example

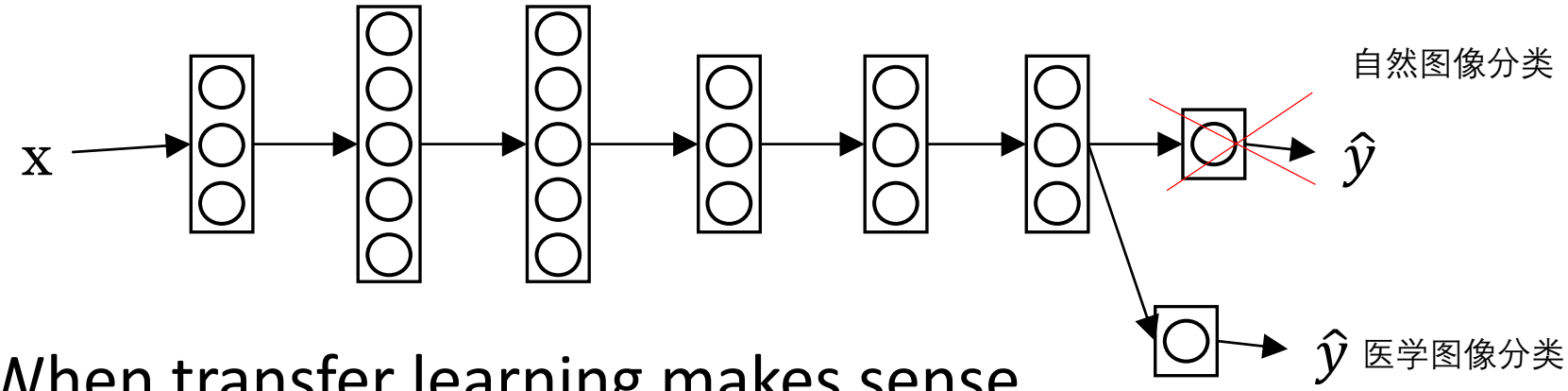
- Noisy background
  - Café noise
  - Car noise
- Accented speech
- Far from microphone
- Young children's speech
- Stuttering
- …

- Set up dev/test set and metric
- Build initial system quickly
- Use Bias/Variance analysis & Error analysis to prioritize next steps.

Guideline:

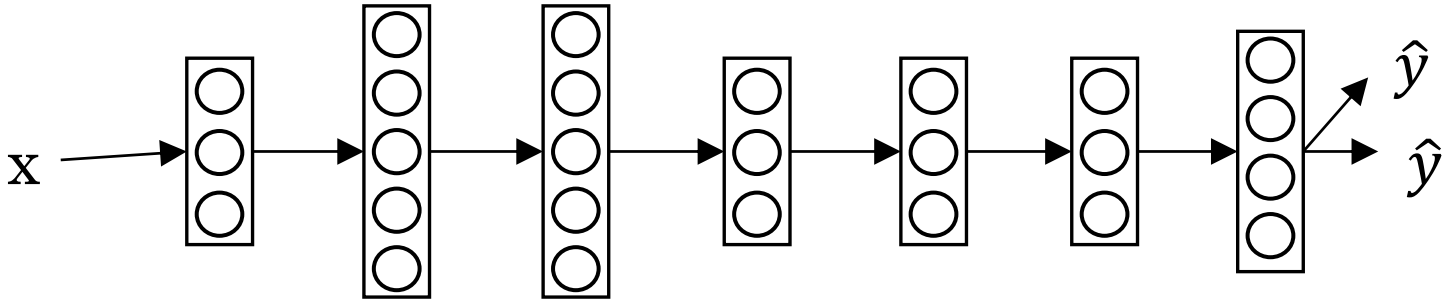
Build your first system quickly, then iterate

迁移学习



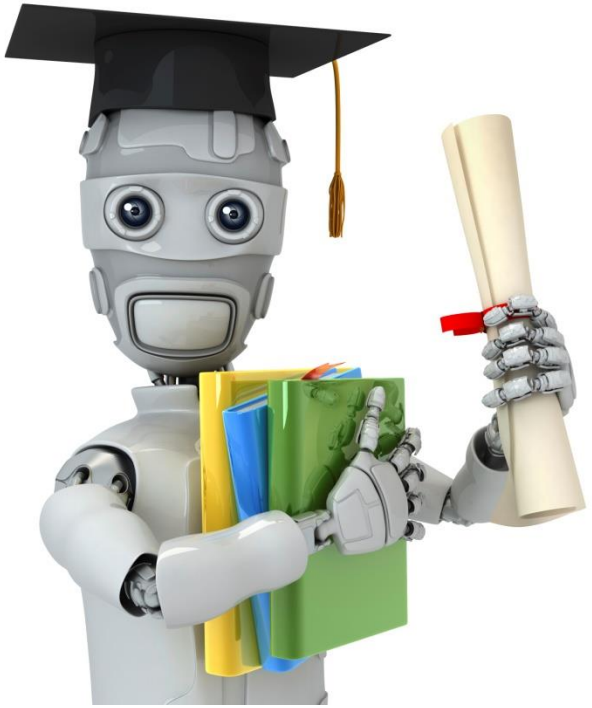
- When transfer learning makes sense
  - Task A and B have the same input x.
  - You have a lot more data for Task A than Task B
  - Low level features from A could be helpful for learning

多任务学习



多任务学习

- When multi-task learning makes sense
  - Training on a set of tasks that could benefit from having shared lower-level features.
  - Usually: Amount of data you have for each task is quite similar
  - Can train a big enough model to do well on all the tasks



Machine Learning

应用

异常检测

异常检测

Aircraft engine features:

x_1 = heat generated

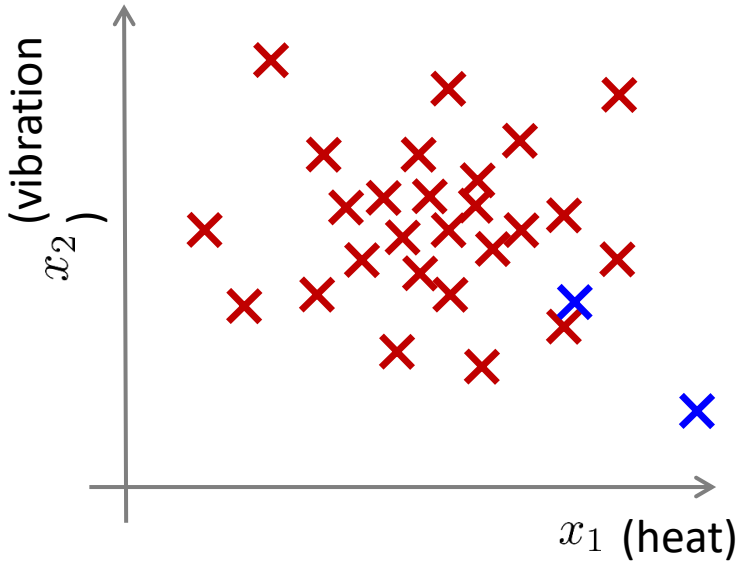
x_2 = vibration intensity

...

Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

New engine: x_{test}

Is x_{test} anomalous?



方法：估计x_{test}出现的概率

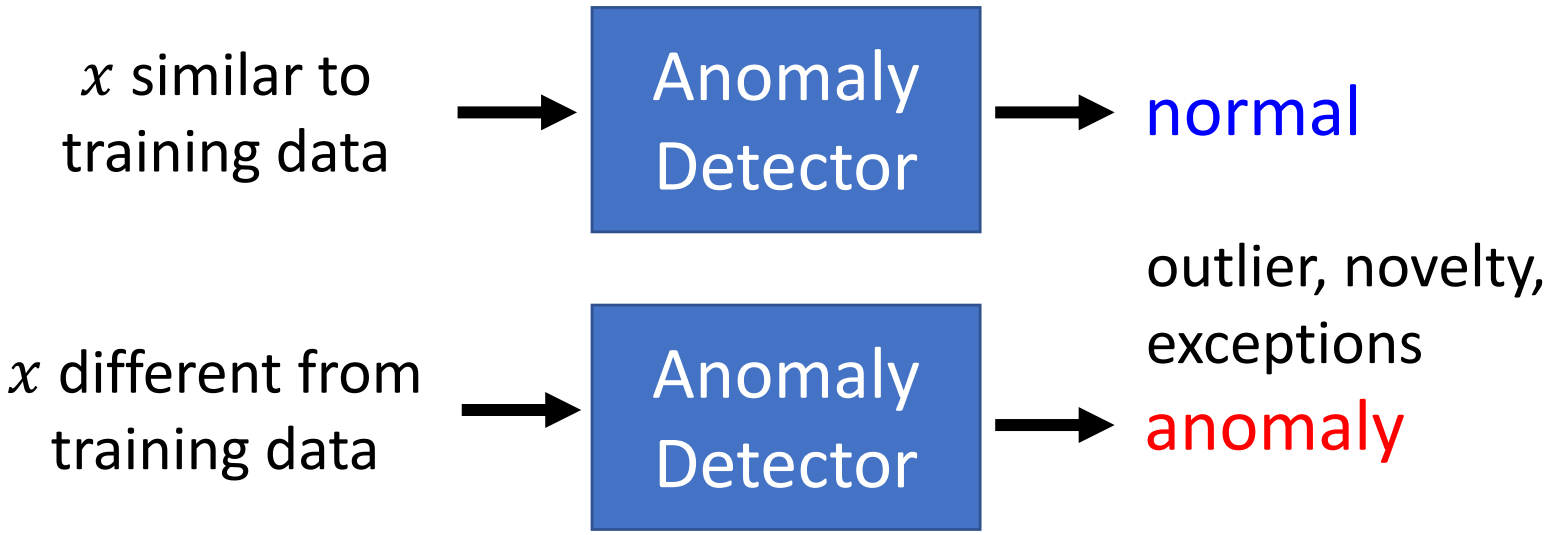
若$p(x_{test}) < \epsilon$ 小概率事件（异常）

若$p(x_{test}) \geq \epsilon$ 正常

常采用高斯模型(或混合高斯模型)

Problem Formulation

- Given a set of training data $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- We want to find a function detecting input is similar to training data or not.



Applications

- Fraud Detection
  - Training data: 正常刷卡行为, x: 盗刷？
  - Ref: https://www.kaggle.com/ntnu-testimon/paysim1/home
  - Ref: https://www.kaggle.com/mlg-ulb/creditcardfraud/home
- Network Intrusion Detection
  - Training data: 正常连接, x: 攻击行为？
  - Ref: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
- Cancer Detection
  - Training data: 正常细胞, x: 癌细胞
  - Ref: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/home

Binary Classification?

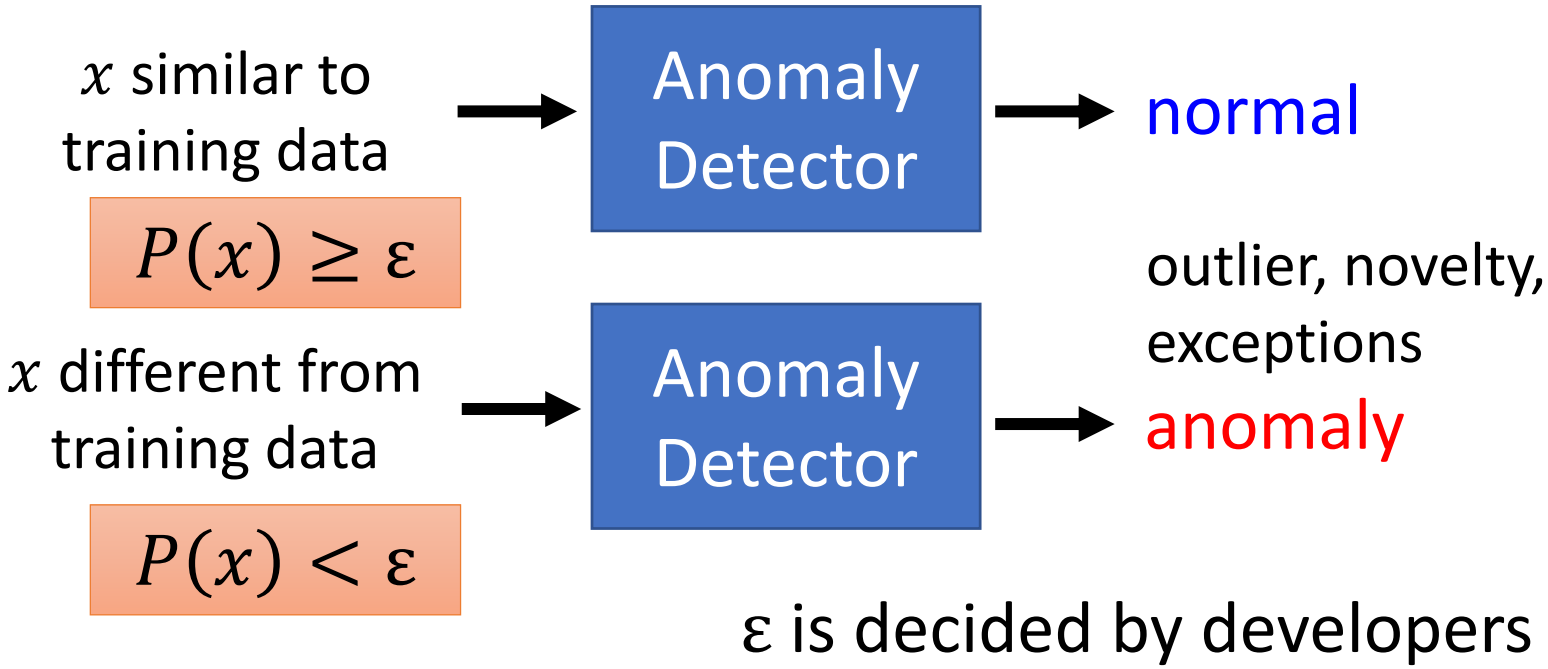
- Given a set of normal data data $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Given anomaly $\{\hat{x}^{(1)}, \hat{x}^{(2)}, \ldots, \hat{x}^{(m)}\}$
- Then training a binary classifier?

Binary Classification?

- Given a set of normal data data $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Given anomaly $\{\hat{x}^{(1)}, \hat{x}^{(2)}, \ldots, \hat{x}^{(m)}\}$
- Then training a binary classifier?
- NO. anomaly cannot be considered as a class
- Even worse, in some cases, it is difficult to find anomaly example

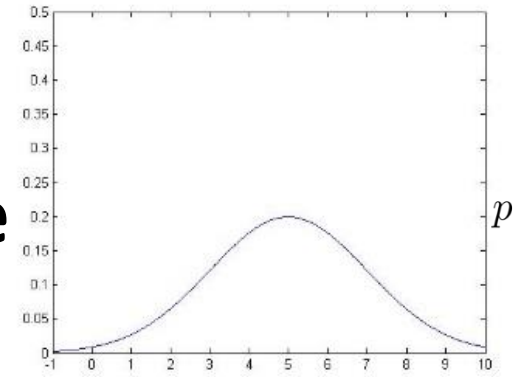
Problem Formulation

- Given a set of training data $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ which are generated from distribution $p(x)$
- We want to find a function detecting input is similar to training data or not.

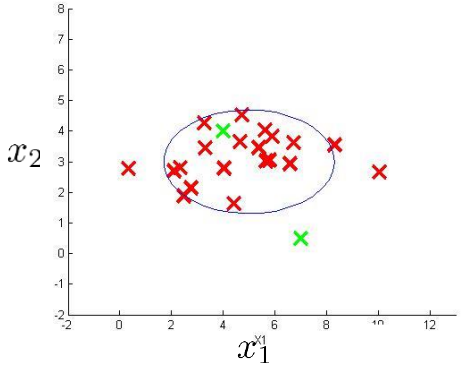


异常检测

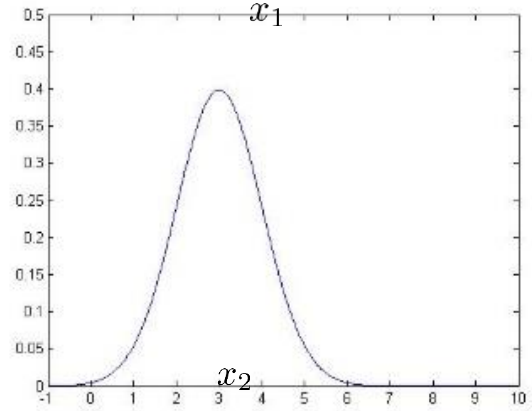
Anomaly detection example



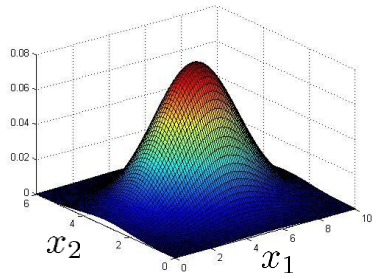
$p(x_1; \mu_1, \sigma_1^2)$



$$\mu_1 = 5, \sigma_1 = 2$$
$$\mu_2 = 3, \sigma_2 = 1$$



$p(x_2; \mu_2, \sigma_2^2)$



$$\varepsilon = 0.02$$
$$p(x_{test}^{(1)}) = 0.0426$$
$$p(x_{test}^{(2)}) = 0.0021$$

异常检测

- 假设有部分带标注的数据($y = 0$ 表示正常, $y = 1$表示异常)和部分无标注的数据
- 用不带标注的数据训练模型
- 将带标注的数据分为验证集和测试集去进行模型选择和评估

异常检测

Aircraft engines motivating example

- 10000 good (normal) engines, 20 flawed engines (anomalous)
  - Training set: 6000 good engines
  - CV set: 2000 good engines ($y = 0$), 10 anomalous ($y = 1$)
  - Test set: 2000 good engines ($y = 0$), 10 anomalous ($y = 1$)
- Alternative:
  - Training set: 8000 good engines
  - CV set: 1000 good engines ($y = 0$), 10 anomalous ($y = 1$)
  - Test set: 1000 good engines ($y = 0$), 10 anomalous ($y = 1$)

异常检测

- Fit model $p(x)$ on training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- On a cross validation/test example x , predict

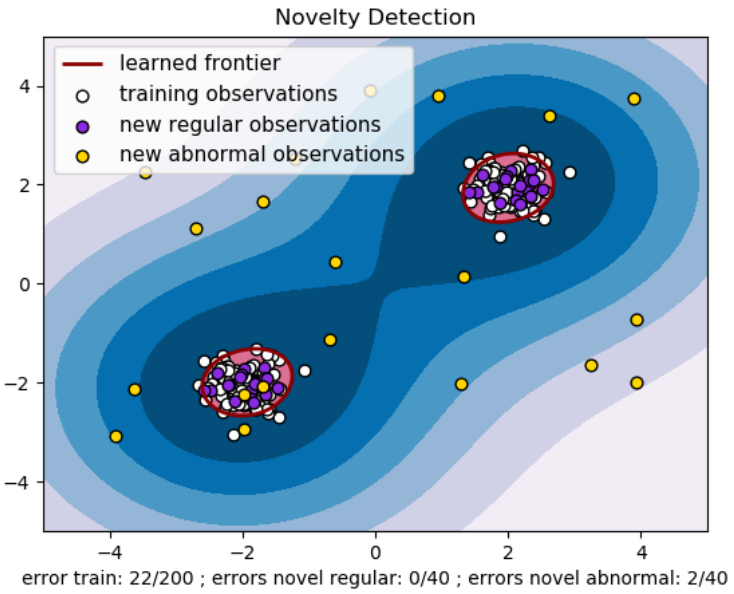
$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

- Possible evaluation metrics:
  - True positive, false positive, false negative, true negative
  - Precision/Recall
  - F1-score
- Can also use cross validation set to choose parameter ε

More …

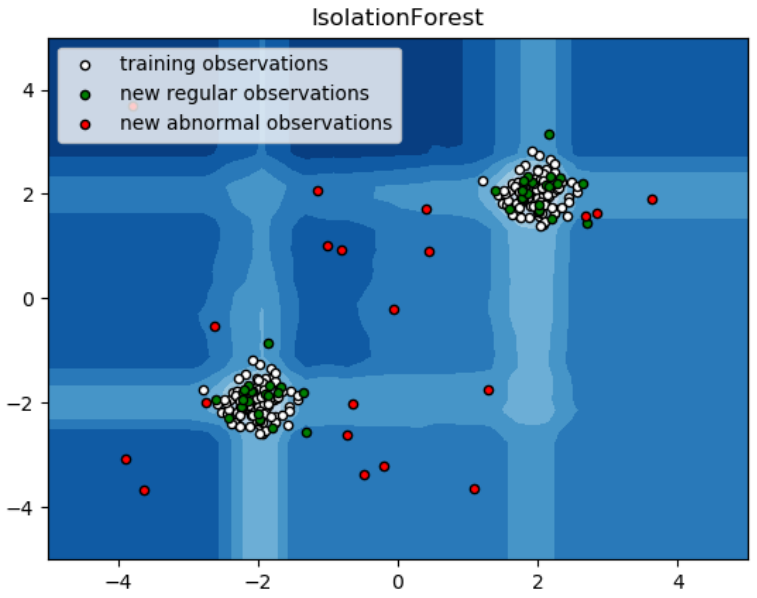
Source of images: https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection

One-class SVM



Ref: https://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection.pdf

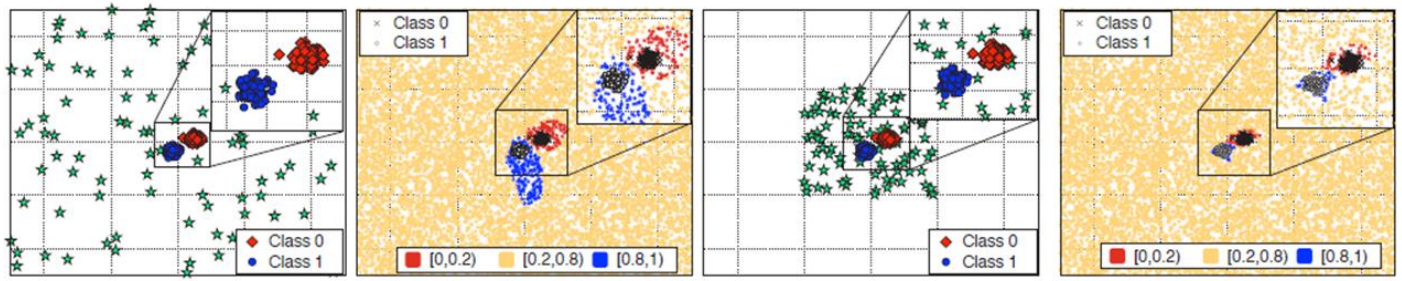
Isolated Forest



Ref: https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf

More ……

- Learn a classifier giving low confidence score to anomaly



Kimin Lee, Honglak Lee, Kibok Lee, Jinwoo Shin, Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples, ICLR 2018

- How can you obtain anomaly?

Generating by Generative Models?

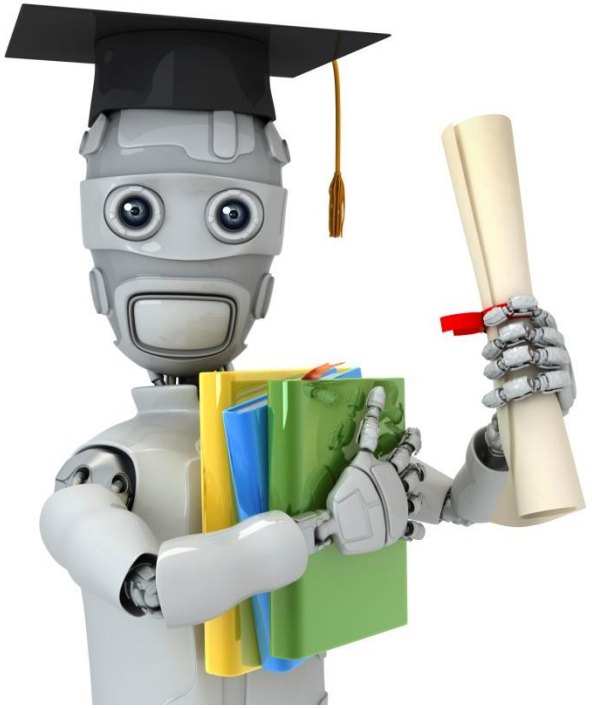
Mark Kliger, Shachar Fleishman, Novelty Detection with GAN, arXiv, 2018

异常检测 vs. 监督学习

Very small number of positive examples ($y = 1$). (0-20 is common).
Large number of negative ($y = 0$) 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.

Large number of positive and negative examples.

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.



Machine Learning

应用

推荐系统

基于内容的推荐系统

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

For each user j, learn a parameter $\theta^{(j)} \in \mathbb{R}^3$. Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$ stars.

基于内容的推荐系统

- $r(i,j) = 1$ if user j has rated movie i (0 otherwise)
- $y^{(i,j)}$ = rating by user j on movie i (if defined)
- $\theta^{(j)}$ = parameter vector 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 $\theta^{(j)}$

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

基于内容的推荐系统

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

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

Gradient descent update:

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

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

协同滤波(Collaborative filtering)

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}, \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

协同滤波(Collaborative filtering)

- 给定$\theta^{(1)},\ldots,\theta^{(n_u)}$, 学习$x^{(i)}$

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

- 给定$\theta^{(1)},\ldots,\theta^{(n_u)}$, 学习$x^{(1)},\ldots,x^{(n_m)}$

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

协同滤波(Collaborative filtering)

- 给定$x^{(1)}, \ldots, x^{(n_m)}$, 可以估计$\theta^{(1)}, \ldots, \theta^{(n_u)}$

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

- 给定$\theta^{(1)}, \ldots, \theta^{(n_u)}$, 可以估计$x^{(1)}, \ldots, x^{(n_m)}$

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

- $\theta \to x \to \theta \to \cdots$

协同滤波(Collaborative filtering)

- 同时最小化$x^{(1)}, \ldots, x^{(n_m)}$和$\theta^{(1)}, \ldots, \theta^{(n_u)}$:

$$J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)}) =$$

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

$$\min_{\substack{x^{(1)}, \ldots, x^{(n_m)} \\ \theta^{(1)}, \ldots, \theta^{(n_u)}}} J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$$

协同滤波(Collaborative filtering)算法

- 用小的随机数初始化$x^{(1)}, \ldots, x^{(n_m)}$和$\theta^{(1)}, \ldots, \theta^{(n_u)}$
- 基于梯度下降法最小化$J(x^{(1)}, \ldots, x^{(n_m)}, \theta^{(1)}, \ldots, \theta^{(n_u)})$:

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

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

协同滤波(Collaborative filtering): 矢量化

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

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

Predicted ratings:

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(n_m)})^T \end{bmatrix} \qquad \Theta = \begin{bmatrix} (\theta^{(1)})^T \\ (\theta^{(2)})^T \\ \vdots \\ (\theta^{(n_u)})^T \end{bmatrix}$$

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

协同滤波(Collaborative filtering)

- 对于每部电影，都会学习到一个特征表示 $x^{(i)} \in \mathbb{R}^n$
- 思考:
  - 如何找到与某部电影最相关的电影?
  - 如何找到与某部电影最相关的5部电影?

协同滤波： Mean normalization

Users who have not rated any movies

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	Eve (5)
Love at last	5	5	0	0	?
Romance forever	5	?	?	0	?
Cute puppies of love	?	4	0	?	?
Nonstop car chases	0	0	5	4	?
Swords vs. karate	0	0	5	?	?

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

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

协同滤波： Mean normalization

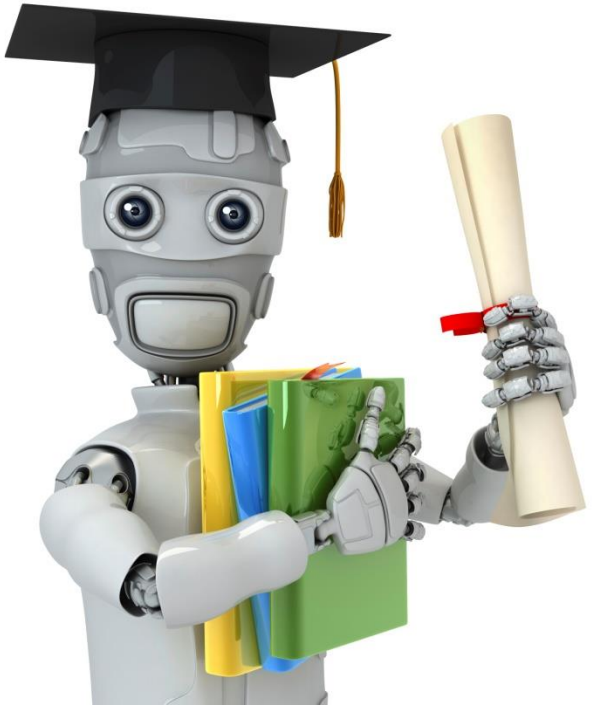
$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad \mu = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.25 \\ 1.25 \end{bmatrix} \quad \rightarrow Y = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 &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

For user j, on movie i predict:

$$(\theta^{(j)})^T(x^{(i)}) + \mu_i$$

User 5 (Eve):

$$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



Machine Learning

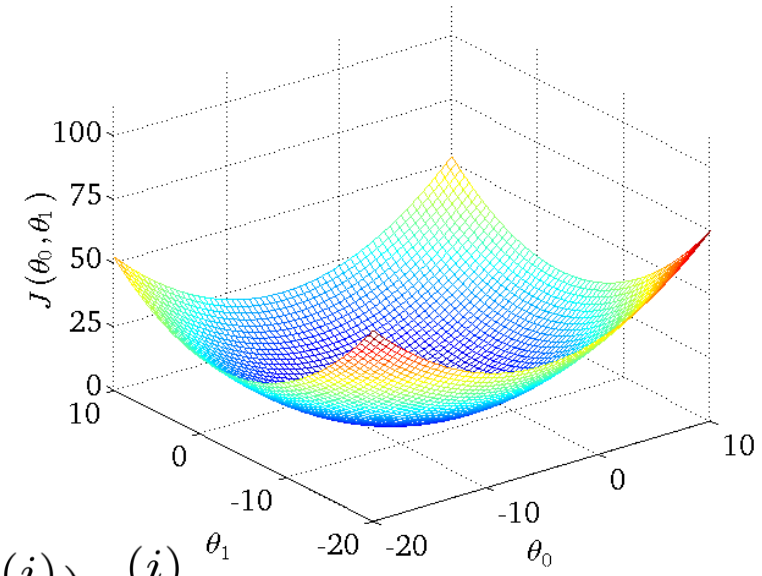
应用

Large Scale Machine Learning

Recap: Linear Regression with Gradient Descent

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

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



Repeat {

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

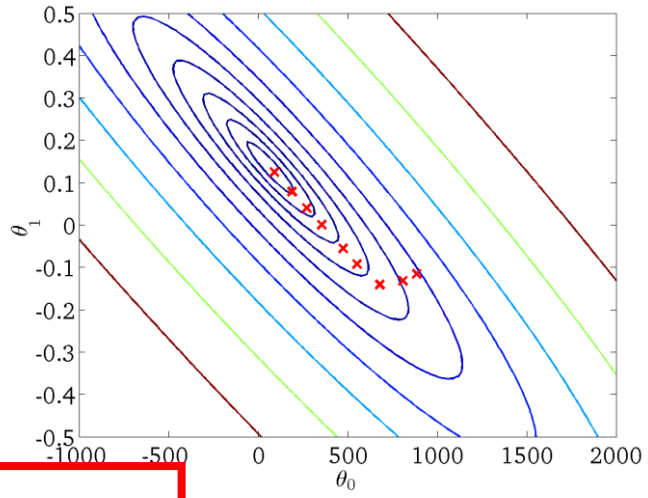
(for every $j = 0, \ldots, n$)

}

Recap: Linear Regression with Gradient Descent

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

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



Repeat {

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

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

}

Batch Gradient Descent $m = 300,000,000$

Stochastic gradient descent

- Randomly shuffle (reorder) training examples

- Repeat {

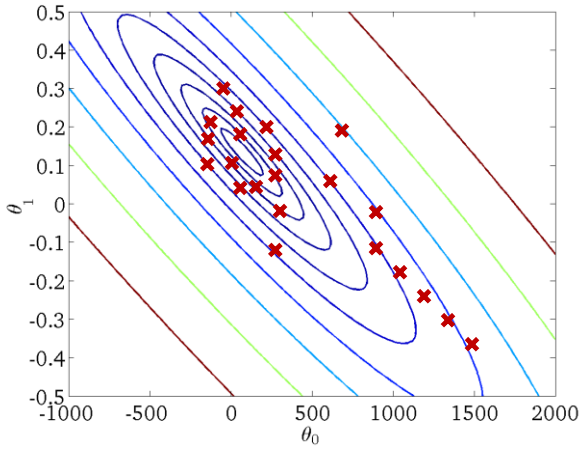
  for $i = 1, \cdots, m$ {

  $\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$

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

  }

  }



Stochastic gradient descent

Batch gradient descent

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

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

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

}

Stochastic gradient descent

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

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

- Randomly shuffle dataset
- Repeat {

  for $i = 1, \cdots, m$ {

  $\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$

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

  }

  }

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

Say $b = 10, m = 1000$
Repeat {
for $i = 1, 11, 21, 31, \ldots, 991\{$

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)} \text{ (for every } j = 0, \ldots, n)$$

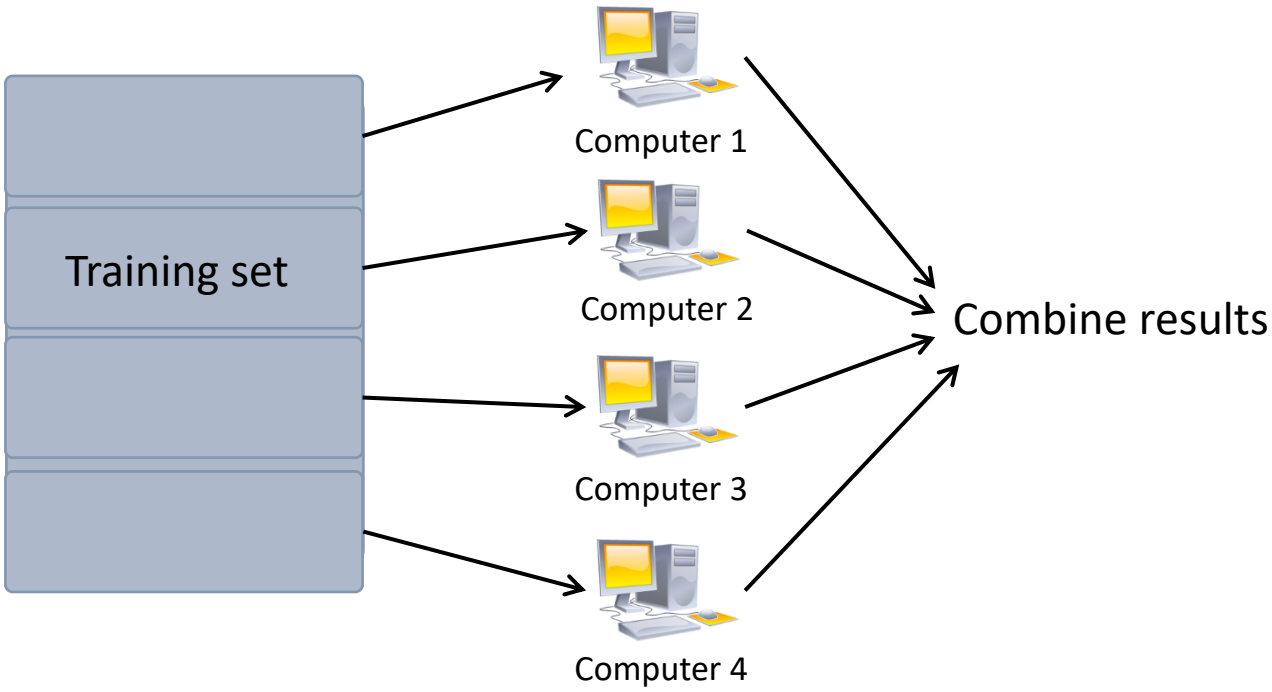
}}

Map-reduce and data parallelism

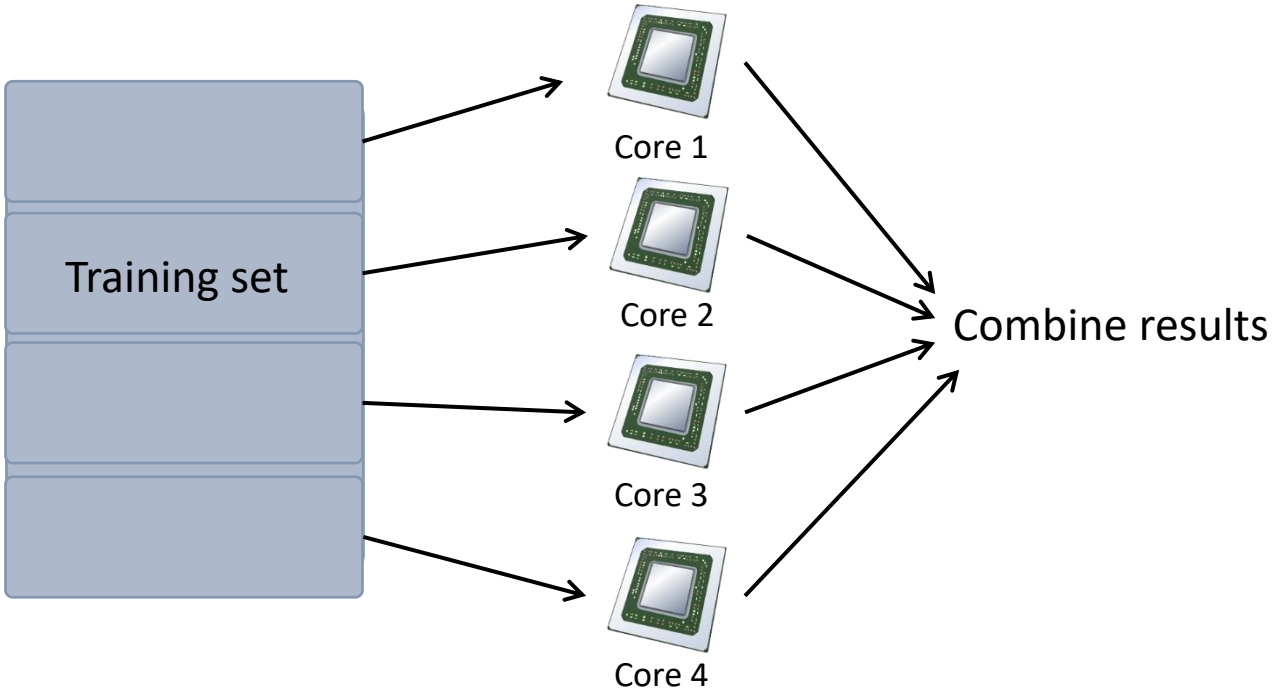
Batch gradient descent: $\theta_j := \theta_j - \alpha \frac{1}{400} \sum_{i=1}^{400} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

- Machine 1: Use $(x^{(1)}, y^{(1)}), \ldots, (x^{(100)}, y^{(100)})$.
  $temp_j^{(1)} = \sum_{i=1}^{100} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$
- Machine 2: Use $(x^{(101)}, y^{(101)}), \ldots, (x^{(200)}, y^{(200)})$.
  $temp_j^{(2)} = \sum_{i=101}^{200} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$
- Machine 3: $(x^{(201)}, y^{(201)}), \ldots, (x^{(300)}, y^{(300)})$.
  $temp_j^{(3)} = \sum_{i=201}^{300} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$
- Machine 4: $(x^{(301)}, y^{(301)}), \ldots, (x^{(400)}, y^{(400)})$.
  $temp_j^{(4)} = \sum_{i=301}^{400} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$

Map-reduce and data parallelism



Map-reduce and data parallelism



Thanks!

Any questions?