CIS 530—Advanced Data Mining

2- Linear Algebra and Matrix

Thomas W. Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy of Fei-Fei Li: http://vision.stanford.edu/teaching/cs131_fall1617/

Outline

Vectors and Matrices
Basic operations
Special matrices

More Matrix Operations
Matrix inverse
Matrix rank
Singular Value Decomposition (SVD)
Use for image compression

Transformation Matrices
Homogeneous coordinates
Translation

Vector

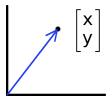
- A column vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$ where
$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$
- A row vector $\mathbf{v}^T \in \mathbb{R}^{1\times n}$ where
$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

Vector

- We'll default to column vectors in this class
$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$
- You'll want to keep track of the orientation of your vectors when programming in MATLAB
- In MATLAB v' means v^T, indicating the transpose operation

Vectors have two main uses



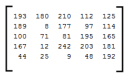
- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin

- Data (pixels, gradients at an image keypoint, etc.) can also be treated as a vector
- Such vectors don't have a geometric interpretation, but calculations like "distance" can still have value

Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is an array of numbers with size by, i.e., m rows and n columns.
$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$
- If $m = n$, we say that $\mathbf{A}$ is square.

Images

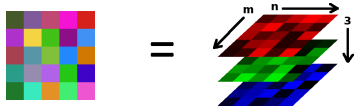


MATLAB represents an image as a matrix of pixel brightnesses

Note that matrix coordinates are NOT Cartesian coordinates. The upper leg corner is [y,x] = (1,1)

Color Images

- Grayscale images have one number per pixel, and are stored as an m × n matrix.
- Color images have 3 numbers per pixel – red, green, and blue brightnesses (RGB)
- Stored as an m × n × 3 matrix



Basic Matrix Operations

- Addition
- Scaling
- Dot product
- Multiplication
- Transpose
- Inverse / pseudoinverse
- Determinant / trace

Matrix Operations

- Addition
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$
  - Can only add a matrix with matching dimensions, or a scalar.
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$
- Scaling
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

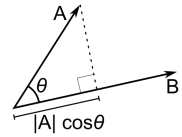
Matrix Operations

- Inner product (dot product) of vectors
  - Multiply corresponding entries of two vectors and add up the result
  - $\mathbf{x}\cdot\mathbf{y}$ is also $|\mathbf{x}||\mathbf{y}|\cos(\textit{the angle between } \mathbf{x} \textit{ and } \mathbf{y})$

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad \text{(scalar)}$$

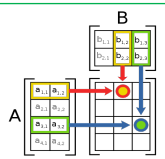
Matrix Operations

- Inner product (dot product) of **vectors**
  - If B is a unit vector, then $\mathbf{A}\cdot\mathbf{B}$ gives the length of A which lies in the direction of B



Matrix Operations

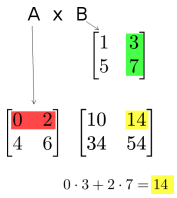
- Multiplication
- The product AB is:



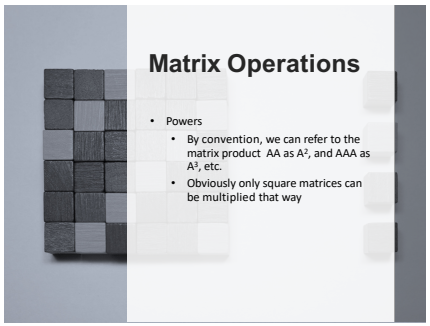
- Each entry in the result is (that row of A) dot product with (that column of B)
- Many uses, which will be covered later

Matrix Operations

- Multiplication example:



– Each entry of the matrix product is made by taking the dot product of the corresponding row in the leg matrix, with the corresponding column in the right one.



Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1

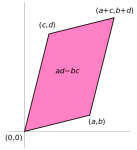
$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:
$$(ABC)^T = C^T B^T A^T$$

Matrix Operations

- Determinant
  - $\det(\mathbf{A})$returns a scalar
  - Represents area (or volume) of the parallelogram described by the vectors in the rows of the matrix
  - For $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\det(\mathbf{A}) = ad - bc$
  - Properties:

$$\det(\mathbf{AB}) = \det(\mathbf{BA})$$
$$\det(\mathbf{A}^{-1}) = \frac{1}{\det(\mathbf{A})}$$
$$\det(\mathbf{A}^T) = \det(\mathbf{A})$$
$$\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A} \text{ is singular}$$



Matrix Operations

- Trace

$\mathrm{tr}(\mathbf{A}) = \text{sum of diagonal elements}$

$\mathbf{tr}\left(\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}\right) = 1 + 7 = 8$

  - Invariant to a lot of transformations, so it's used sometimes in proofs. (Rarely in this class though.)
  - Properties:

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$$
$$\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$$

Special Matrices

- Identity matrix **I**
  - Square matrix, 1's along diagonal, 0's elsewhere
  - **I**×[another matrix] = [that matrix]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Diagonal matrix
  - Square matrix with numbers along diagonal, 0's elsewhere
  - A diagonal × [another matrix] scales the rows of that matrix

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

Special Matrices

- Symmetric matrix

$$\mathbf{A}^T = \mathbf{A} \qquad \begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

- Skew-symmetric matrix

$$\mathbf{A}^T = -\mathbf{A} \qquad \begin{bmatrix} 0 & -2 & -5 \\ 2 & 0 & -7 \\ 5 & 7 & 0 \end{bmatrix}$$

Outline

- Vectors and Matrices
  - Basic operations
  - Special matrices
- More Matrix Operations
  - Matrix inverse
  - Matrix rank
  - Singular Value Decomposition (SVD)
  - Use for image compression
- Transformation Matrices
  - Homogeneous coordinates
  - Translation

Inverse

- Given a matrix **A**, its inverse $\mathbf{A}^{-1}$ is a matrix such that $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
  - E.g. $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$
- Inverse does not always exist. If $\mathbf{A}^{-1}$ exists, **A** is *invertible* or *non–singular*. Otherwise, it's *singular*.
- Useful identities, for matrices that are invertible:

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A}$$
$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$
$$\mathbf{A}^{-T} \triangleq (\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$$

Pseudoinverse

- Say you have the matrix equation AX=B, where A and B are known, and you want to solve for X
- You could use MATLAB to calculate the inverse and pre-multiply by it: $A^{-1}AX=A^{-1}B \rightarrow X=A^{-1}B$
- MATLAB command would be inv(A)*B
- But calculating the inverse for large matrices often brings problems with computer floating-point resolution (because it involves working with very small and very large numbers together).
- Or your matrix might not even have an inverse.

Pseudoinverse

Fortunately, there are workarounds to solve AX=B in these situations. And MATLAB can do them!

Instead of taking an inverse, directly ask MATLAB to solve for X in AX=B, by typing A\B, called "mldivide"

For complete instruction please check:

https://www.mathworks.com/help/matlab/ref/mldivide.html

Pseudoinverse

- MATLAB will try several appropriate numerical methods (including the pseudoinverse if the inverse doesn't exist)
  - If there is no exact solution, it will return the closest one
  - If there are many solutions, it will return the smallest one

Pseudoinverse

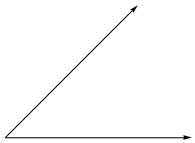
- MATLAB example:

$$AX = B$$
$$A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

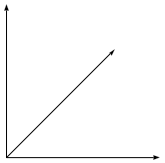
```
>> x = A\B
x =
    1.0000
   -0.5000
```

Linear Independence

Linearly independent set



Not linearly independent



Linear Independence

- Suppose we have a set of vectors $v_1, v_2, \ldots v_n$
- If we can express v_1 as a linear combination of the other vectors $v_2, \ldots v_n$ then v_1 is linearly dependent on the other vectors
  - $v_1 = [0,1], v_2 = [0,2]$
- If no vector is linearly dependent on the rest of the set, the set is linearly independent
  - $v_1 = [1,0,0], v_2 = [0,1,0], v_3 = [0,0,1]$

Matrix Rank

- Column/row rank

$\text{col-rank}(\mathbf{A}) = \text{the maximum number of linearly independent column vectors of } \mathbf{A}$
$\text{row-rank}(\mathbf{A}) = \text{the maximum number of linearly independent row vectors of } \mathbf{A}$

- Column rank always equals row rank
- Matrix rank

$$\mathrm{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$$

$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 2 & 0 & 2 \end{bmatrix}$ **What is the rank of the left matrix and why?**

Matrix Rank

- For transformation matrices, the rank tells you the **dimensions** of the output
- E.g., if rank of **A** is 1, then the transformation

p'=Ap

maps points onto a line.

- Here's a matrix with rank 1:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + y \\ 2x + 2y \end{bmatrix}$$

All points get mapped to the line y=2x

Matrix Rank

If an m x m matrix is rank m, we say it's "full rank"

- Maps an *m x 1* vector uniquely to another *m x 1* vector
- An inverse matrix can be found

If rank < *m*, we say it's "singular"

At least one dimension is getting collapsed. No way to look at the result and tell what the input was

- Inverse does not exist

Inverse also doesn't exist for non-square matrices

Singular Value Decomposition

There are several computer algorithms that can "factorize" a matrix, representing it as the product of some other matrices

The most useful of these is the **Singular Value Decomposition (SVD)**.

Represent any matrix **A** as a product of three matrices: $\mathbf{UEV}^T$

MATLAB command: [U,S,V]=svd(A)

Singular Value Decomposition

$$\mathbf{U\Sigma V^T = A}$$

- Where **U** and **V** are rotation matrices, and **Σ** is a scaling matrix. For example:

$$\underset{U}{\begin{bmatrix}-.40 & .916\\ .916 & .40\end{bmatrix}}\times\underset{\Sigma}{\begin{bmatrix}5.39 & 0\\ 0 & 3.154\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.05 & .999\\ .999 & .05\end{bmatrix}}=\underset{A}{\begin{bmatrix}3 & -2\\ 1 & 5\end{bmatrix}}$$

Singular Value Decomposition

U and **V** are always rotation matrices.

- Geometric rotation may not be an applicable concept, depending on the matrix. So, we call them "unitary" matrices – each column is a unit vector.

Σ is a diagonal matrix

- The number of nonzero entries = rank of **A**
- The algorithm always sorts the entries high to low

$$\underset{U}{\begin{bmatrix}-.39 & -.92\\ -.92 & .39\end{bmatrix}}\times\underset{\Sigma}{\begin{bmatrix}9.51 & 0 & 0\\ 0 & .77 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}=\underset{A}{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}}$$

SVD Applications

- We've discussed SVD in terms of geometric transformation matrices
- But SVD of an image matrix can also be very useful
- To understand this, we'll look at a less geometric interpretation of what SVD is doing
- An **outer-product view**

SVD Applications

$$\underset{U}{\begin{bmatrix}-.39 & -.92\\ -.92 & .39\end{bmatrix}}\times\underset{\Sigma}{\begin{bmatrix}9.51 & 0 & 0\\ 0 & .77 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}=\underset{A}{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}}$$

- Look at how the multiplication works out, left to right:
- Column 1 of **U** gets scaled by the first value from **Σ**.

$$\underset{U\Sigma}{\begin{bmatrix}-3.67 & -.71 & 0\\ -8.8 & .30 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}\quad\underset{A_{partial}}{\begin{bmatrix}1.6 & 2.1 & 2.6\\ 3.8 & 5.0 & 6.2\end{bmatrix}}$$

- The resulting vector gets scaled by row 1 of $\mathbf{V^T}$ to produce a contribuLon to the columns of **A**

SVD Applications

$$\underset{U\Sigma}{\begin{bmatrix}-3.67 & -.71 & 0\\ -8.8 & .30 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}\quad\underset{A_{partial}}{\begin{bmatrix}1.6 & 2.1 & 2.6\\ 3.8 & 5.0 & 6.2\end{bmatrix}}$$

$$+\ \underset{U\Sigma}{\begin{bmatrix}-3.67 & -.71 & 0\\ -8.8 & .30 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}\quad\underset{A_{partial}}{\begin{bmatrix}-.6 & -.1 & .4\\ .2 & 0 & -.2\end{bmatrix}}$$

$$=\quad\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}$$

- Each product of (column i of **U**)·(value i from **Σ**)·(row i of $\mathbf{V^T}$) produces a component of the final **A**.

SVD Applications

$$\underset{U\Sigma}{\begin{bmatrix}-3.67 & -.71 & 0\\ -8.8 & .30 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}\quad\underset{A_{partial}}{\begin{bmatrix}1.6 & 2.1 & 2.6\\ 3.8 & 5.0 & 6.2\end{bmatrix}}\qquad\underset{A}{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}}$$

$$\underset{U\Sigma}{\begin{bmatrix}-3.67 & -.71 & 0\\ -8.8 & .30 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}\quad\underset{A_{partial}}{\begin{bmatrix}-.6 & -.1 & .4\\ .2 & 0 & -.2\end{bmatrix}}$$

We can call those first few columns of **U** the *Principal Components* of the data

They show the major patterns that can be added to produce the columns of the original matrix

The rows of $\mathbf{V^T}$ show how the *principal components* are mixed to produce the columns of the matrix

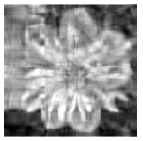
SVD Applications

$$\underset{U}{\begin{bmatrix}-.39 & -.92\\ -.92 & .39\end{bmatrix}}\times\underset{\Sigma}{\begin{bmatrix}9.51 & 0 & 0\\ 0 & .77 & 0\end{bmatrix}}\times\underset{V^T}{\begin{bmatrix}-.42 & -.57 & -.70\\ .81 & .11 & -.58\\ .41 & -.82 & .41\end{bmatrix}}=\underset{A}{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}}$$

We can look at **Σ** to see that the first column has a large effect

while the second column has a much smaller effect in this example

SVD Applications



For this image, using **only the first 10** of 300 principal components produces a recognizable reconstructon

So, SVD can be used for image compression

Outline

- Vectors and Matrices
  - Basic operations
  - Special matrices
- More Matrix Operations
  - Matrix inverse
  - Matrix rank
  - Singular Value Decomposition (SVD)
  - Use for image compression
- Transformation Matrices
  - Homogeneous coordinates
  - Translation

Transformation

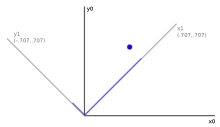
- Matrices can be used to transform vectors in useful ways, through multiplication: x'= Ax
- Simplest is scaling:

$$\begin{bmatrix}s_x & 0\\ 0 & s_y\end{bmatrix}\times\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}s_x x\\ s_y y\end{bmatrix}$$

- (Verify to yourself that the matrix multiplication works out this way)

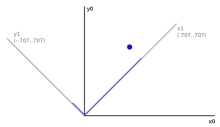
Rotation

- How can you convert a vector represented in frame "0" to a rotated coordinate frame "1"?
- Remember what a vector is:
  [component in direction of the frame's x axis, and component in direction of y axis]



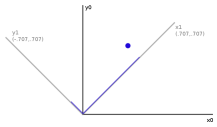
Rotation

- So to rotate it we must produce this vector:
  [component in direction of **new** x axis, component in direction of **new** y axis]
- We can do this easily with ***dot products***!
- New x coordinate is [original vector] **dot** [the new x axis]
- New y coordinate is [original vector] **dot** [the new y axis]



Rotation

- Insight: this is what happens in a matrix*vector multiplication
  - Result x coordinate is:
    [**original vector**] **dot** [**matrix row 1**]
  - So matrix multiplication can rotate a vector p:

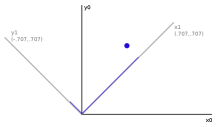


R x p = rotated p'

$$\underset{R}{\begin{bmatrix}.707 & .707\\ -.707 & .707\end{bmatrix}}\quad\underset{p}{\begin{bmatrix}px\\ py\end{bmatrix}}\quad\begin{bmatrix}px'\\ py'\end{bmatrix}$$

Rotation

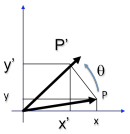
- Suppose we express a point in the new coordinate system which is rotated left
- If we plot the result in the **original** coordinate system, we have rotated the point right



– Thus, rotation matrices can be used to rotate vectors. We'll usually think of them in that sense---- as operators to rotate vectors

2D Rotation Matrix Formula

Counter-clockwise rotation by an angle θ



$$x' = \cos\theta\, x - \sin\theta\, y$$
$$y' = \cos\theta\, y + \sin\theta\, x$$

$$\begin{bmatrix}x'\\ y'\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}$$

$$\mathbf{P' = R\ P}$$

Transformation Matrices

- Multiple transformation matrices can be used to transform a point: $p' = R_2 R_1 S\, p$
- The effect of this is to apply their transformations one ager the other, from **right to left.**
- In the example above, the result is $(R_2 (R_1 (S\, p)))$
- The result is exactly the same if we multiply the matrices first, to form a single transformation matrix:
- $p' = (R_2 R_1 S)\, p$

Homogeneous System

- In general, a matrix multiplication lets us linearly combine components of a vector

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

  - This is sufficient for scale, rotate, skew transformations.
  - But notice, we can't add a constant! ☹

Homogeneous System

- The (somewhat hacky) solution? Stick a "1" at the end of every vector:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+c \\ dx+ey+f \\ 1 \end{bmatrix}$$

- Now we can rotate, scale, and skew like before, AND translate (note how the multiplication works out, above)
- This is called "homogeneous coordinates"

Homogeneous System

- In homogeneous coordinates, the multiplication works out so the rightmost column of the matrix is a vector that gets added.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+c \\ dx+ey+f \\ 1 \end{bmatrix}$$

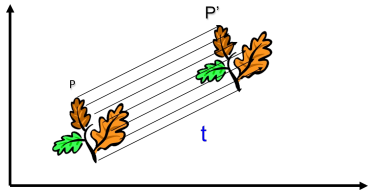
- Generally, a homogeneous transformation matrix will have a bottom row of [0 0 1], so that the result has a "1" at the bottom too.

Homogeneous System

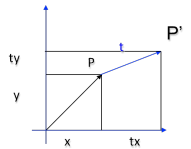
- One more thing we might want: to divide the result by something
  - For example, we may want to divide by a coordinate, to make things scale down as they get farther away in a camera image
  - Matrix multiplication can't actually divide
  - So, **by convention**, in homogeneous coordinates, we'll divide the result by its last coordinate after doing a matrix multiplication

$$\begin{bmatrix} x \\ y \\ 7 \end{bmatrix} \Rightarrow \begin{bmatrix} x/7 \\ y/7 \\ 1 \end{bmatrix}$$

2D Translation

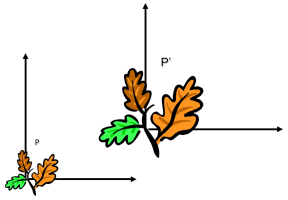


Using Homogeneous Coordinates

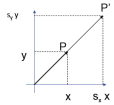


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$
$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$
$$\mathbf{P'} \rightarrow \begin{bmatrix} x+t_x \\ y+t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{T} \cdot \mathbf{P}$$

Scaling

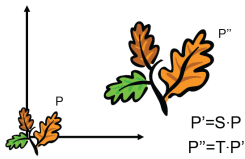


Scaling Equation



$$\mathbf{P} = (x, y) \rightarrow \mathbf{P'} = (s_x x, s_y y)$$
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$
$$\mathbf{P'} = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$
$$\mathbf{P'} \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S'} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

Scaling & Translation



$$\mathbf{P''} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \mathbf{T} \cdot (\mathbf{S} \cdot \mathbf{P}) = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \mathbf{A} \cdot \mathbf{P}$$

Scaling & Translation

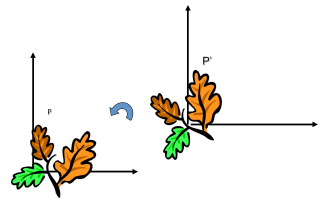
$$\mathbf{P''} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$
$$= \underbrace{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Order Matters

$$\mathbf{P'''} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

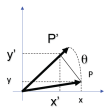
$$\mathbf{P'''} = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

Rotation



Rotation Equation

Counter-clockwise rotation by an angle θ



$$x' = \cos\theta\, x - \sin\theta\, y$$
$$y' = \cos\theta\, y + \sin\theta\, x$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
$$\mathbf{P'} = \mathbf{R}\,\mathbf{P}$$

Rotation Matrix Properties

- Transpose of a rotation matrix produces a rotation in the opposite direction

$$\mathbf{R} \cdot \mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{I}$$
$$\det(\mathbf{R}) = 1$$

- The rows of a rotation matrix are always mutually perpendicular (a.k.a. orthogonal) unit vectors
  - (and so are its columns)

Properties

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

A 2D rotation matrix is 2x2

Note: R belongs to the category of *normal* matrices and satisfies many interesting properties:

$$\mathbf{R} \cdot \mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{I}$$
$$\det(\mathbf{R}) = 1$$

Scaling + Translation + Rotation

$$\mathbf{P'} = (\mathbf{T}\,\mathbf{R}\,\mathbf{S})\,\mathbf{P}$$
$$\mathbf{P'} = \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$
$$= \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$
$$= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} RS & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

This is the form of the general-purpose transformation matrix

CIS 530—Advanced Data Mining

3- Introduction to Data Mining

Thomas W Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy to Prof. Panayiotis Tsaparas

What is data mining?

After years of data mining there is still no unique answer to this question.

A tentative definition: Data mining is the use of efficient techniques for the analysis of very large collections of data and the extraction of useful and possibly unexpected patterns in data.

Why do we need data mining?

- Huge amounts of raw data!
  - In the digital age, TB of data is generated by the second
    - Mobile devices, digital photographs, web documents.
    - Facebook updates, Tweets, Blogs, User-generated content
    - Transactions, sensor data, surveillance data
    - Queries, clicks, browsing
  - Cheap storage has made possible to maintain this data
- Need to analyze the raw data to extract knowledge

Why do we need data mining?

"The data is the computer"

- Large amounts of data can be more powerful than complex algorithms and models
  - Google has solved many Natural Language Processing problems, simply by looking at the data
  - Example: misspellings, synonyms
- Data is power!
  - Today, the collected data is one of the biggest assets of an online company
  - Query logs of Google
  - The friendship and updates of Facebook
  - Tweets and follows of Twitter
  - Amazon transactions

We need a way to harness the collective intelligence

The data is also very complex

Multiple types of data: tables, time series, images, graphs, etc

Spatial and temporal aspects

Interconnected data of different types:

- From the mobile phone we can collect, location of the user, friendship information, check-ins to venues, opinions through twitter, images though cameras, queries to search engines

Example: transaction data

Billions of real-life customers:

- WALMART: 20M transactions per day
- AT&T 300M calls per day
- Credit card companies: billions of transactions per day.

The point cards allow companies to collect information about specific users

Example: document data

Web as a document repository: estimated 50 billions of web pages

Wikipedia: 4 million articles (and counting)

Example: network data

Web: 50 billion pages linked via hyperlinks

Facebook: 500 million users

Twitter: 300 million users

Instant messenger: ~1billion users

Blogs: 250 million blogs worldwide, presidential candidates run blogs

Example: genomic sequences

http://www.1000genomes.org/page.php

Full sequence of 1000 individuals

$3*10^9$ nucleotides per person → $3*10^{12}$ nucleotides

Lots more data in fact: medical history of the persons, gene expression data

Example: environmental data

- Climate data (just an example)
  https://www.ncdc.noaa.gov/cdo-web/datatools
- "a database of temperature, precipitation and pressure records managed by the National Climatic Data Center, Arizona State University and the Carbon Dioxide Information Analysis Center"
- "6000 temperature stations, 7500 precipitation stations, 2000 pressure stations"
  - Spatiotemporal data

Behavioral data

Mobile phones today record a large amount of information about the user behavior

- GPS records position
- Camera produces images
- Communication via phone and SMS
- Text via facebook updates
- Association with entities via check-ins

Amazon collects all the items that you browsed, placed into your basket, read reviews about, purchased.

Google and Bing record all your browsing activity via toolbar plugins. They also record the queries you asked, the pages you saw and the clicks you did.

Data collected for millions of users on a daily basis

So, what is Data?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
  - Examples: eye color of a person, temperature, etc.
  - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object
  - Object is also known as record, point, case, sample, entity, or instance

Attributes / Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Size: Number of objects
Dimensionality: Number of attributes
Sparsity: Number of populated object-attribute pairs

Types of Attributes

- There are different types of attributes
  - Categorical
    - Examples: eye color, zip codes, words, rankings (e.g, good, fair, bad), height in {tall, medium, short}
    - Nominal (no order or comparison) vs Ordinal (order but not comparable)
  - Numeric
    - Examples: dates, temperature, time, length, value, count.
    - Discrete (counts) vs Continuous (temperature)
    - Special case: Binary attributes (yes/no, exists/not exists)

Numeric Record Data

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute
- Such data set can be represented by an n-by-d data matrix, where there are n rows, one for each object, and d columns, one for each attribute

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Categorical Data

- Data that consists of a collection of records, each of which consists of a fixed set of categorical attributes

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	High	No
2	No	Married	Medium	No
3	No	Single	Low	No
4	Yes	Married	High	No
5	No	Divorced	Medium	Yes
6	No	Married	Low	No
7	Yes	Divorced	High	No
8	No	Single	Medium	Yes
9	No	Married	Medium	No
10	No	Single	Medium	Yes

Document Data

- Each document becomes a `term' vector,
  - each term is a component (attribute) of the vector,
  - the value of each component is the number of times the corresponding term occurs in the document.
  - Bag-of-words representation – no ordering

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Transaction Data

- Each record (transaction) is a set of items.

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

- A set of items can also be represented as a binary vector, where each attribute is an item.
- A document can also be represented as a set of words (no counts)

Sparsity: average number of products bought by a customer

Ordered Data

- Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

- Data is a long ordered string

Ordered Data

- Time series
  - Sequence of ordered (over "time") numeric values.

Types of data

- Numeric data: Each object is a point in a multidimensional space
- Categorical data: Each object is a vector of categorical values
- Set data: Each object is a set of values (with or without counts)
  - Sets can also be represented as binary vectors, or vectors of counts
- Ordered sequences: Each object is an ordered sequence of values.
- Graph data

What can you do with the data?

- Suppose that you are the owner of a supermarket and you have collected billions of market basket data. What information would you extract from it and how would you use it?

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Product placement

Catalog creation

Recommendations

- What if this was an online store?

What can you do with the data?

- Suppose you are a search engine and you have a toolbar log consisting of
  - pages browsed,
  - queries,
  - pages clicked,
  - ads clicked

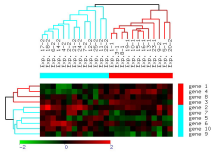
Ad click prediction

Query reformulations

each with a user id and a timestamp. What information would you like to get our of the data?

What can you do with the data?

- Suppose you are biologist who has microarray expression data: thousands of genes, and their expression values over thousands of different settings (e.g. tissues). What information would you like to get out of your data?



Groups of genes and tissues

What can you do with the data?

- Suppose you are a stock broker and you observe the fluctuations of multiple stocks over time. What information would you like to get our of your data?

Clustering of stocks

Correlation of stocks

Stock Value prediction

What can you do with the data?

- You are the owner of a social network, and you have full access to the social graph, what kind of information do you want to get out of your graph?
  - Who is the most important node in the graph?
  - What is the shortest path between two nodes?
  - How many friends two nodes have in common?
  - How does information spread on the network?

Why data mining?

Commercial point of view

- Data has become the key competitive advantage of companies
  - Examples: Facebook, Google, Amazon
- Being able to extract useful information out of the data is key for exploiting them commercially.

Scientific point of view

- Scientists are at an unprecedented position where they can collect TB of information
  - Examples: Sensor data, astronomy data, social network data, gene data
- We need the tools to analyze such data to get a better understanding of the world and advance science

Scale (in data size and feature dimension)

- Why not use traditional analytic methods?
- Enormity of data, curse of dimensionality
- The amount and the complexity of data does not allow for manual processing of the data. We need automated techniques.

What is Data Mining again?

"Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data analyst" (Hand, Mannila, Smyth)

"Data mining is the discovery of models for data" (Rajaraman, Ullman)

- We can have the following types of models
  - Models that explain the data (e.g., a single function)
  - Models that predict the future data instances.
  - Models that summarize the data
  - Models the extract the most prominent features of the data.

What can we do with data mining?

- Some examples:
  - Frequent itemsets and Association Rules extraction
  - Clustering
  - Classification
  - Ranking
  - Exploratory analysis

Frequent Itemsets and Association Rules

- Given a set of records each of which contain some number of items from a given collection;
  - Identify sets of items (itemsets) occurring frequently together
  - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Itemsets Discovered:
{Milk,Coke}
{Diaper, Milk}

Rules Discovered:
{Milk} --> {Coke}
{Diaper, Milk} --> {Beer}

Frequent Item sets: Applications

Text mining: finding associated phrases in text

- There are lots of documents that contain the phrases "association rules", "data mining" and "efficient algorithm"

Recommendations:

- Users who buy this item often buy this item as well
- Users who watched James Bond movies, also watched Jason Bourne movies.
- Recommendations make use of item and user similarity

Association Rule Discovery: Application

- Supermarket shelf management.
  - Goal: To identify items that are bought together by sufficiently many customers.
  - Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
  - A classic rule --
    - If a customer buys diaper and milk, then he is very likely to buy beer.
    - So, don't be surprised if you find six-packs stacked next to diapers!

Clustering Definition

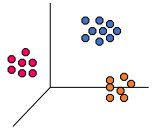
- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
  - Data points in one cluster are more similar to one another.
  - Data points in separate clusters are less similar to one another.
- Similarity Measures?
  - Euclidean Distance if attributes are continuous.
  - Other Problem-specific Measures.

Illustrating Clustering

Euclidean Distance Based Clustering in 3-D space.

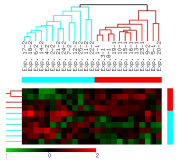
Intracluster distances are minimized

Intercluster distances are maximized



Clustering: Application 1

- Bioinformatics applications:
  - Goal: Group genes and tissues together such that genes are coexpressed on the same tissues



Clustering: Application 2

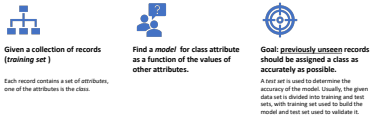
- Document Clustering:
  - Goal: To find groups of documents that are similar to each other based on the important terms appearing in them.
  - Approach: To identify frequently occurring terms in each document. Form a similarity measure based on the frequencies of different terms. Use it to cluster.
  - Gain: Information Retrieval can utilize the clusters to relate a new document or search term to clustered documents.

Clustering of S&P 500 Stock Data

- Observe Stock Movements every day.
- Cluster stocks if they change similarly over time.

	Discovered Clusters	Industry Group
1	Applied-Mat-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

Classification: Definition



Given a collection of records (*training set*)

Each record contains a set of attributes, one of the attributes is the class.

Find a *model* for class attribute as a function of the values of other attributes.

Goal: previously unseen records should be assigned a class as accurately as possible.

A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification Example

categorical, categorical, continuous, class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?

Training Set → Learn Classifier → Model

Test Set → Model

Classification: Application 1

- Ad Click Prediction
  - Goal: Predict if a user that visits a web page will click on a displayed ad. Use it to target users with high click probability.
  - Approach:
    - Collect data for users over a period of time and record who clicks and who does not. The {click, no click} information forms the class attribute.
    - Use the history of the user (web pages browsed, queries issued) as the features.
    - Learn a classifier model and test on new users.

Classification: Application 2

- Fraud Detection
  - Goal: Predict fraudulent cases in credit card transactions.
  - Approach:
    - Use credit card transactions and the information on its account-holder as attributes.
    - When does a customer buy, what does he buy, how often he pays on time, etc
    - Label past transactions as fraud or fair transactions. This forms the class attribute.
    - Learn a model for the class of the transactions.
    - Use this model to detect fraud by observing credit card transactions on an account.

Link Analysis Ranking

Given a collection of web pages that are linked to each other, rank the pages according to importance (authoritativeness) in the graph

- Intuition: A page gains authority if it is linked to by another page.

Application: When retrieving pages, the authoritativeness is factored in the ranking.

Exploratory Analysis

Trying to understand the data as a physical phenomenon, and describe them with simple metrics

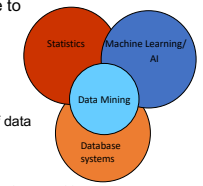
- What does the web graph look like?
- How often do people repeat the same query?
- Are friends in facebook also friends in twitter?

The important thing is to find the right metrics and ask the right questions

It helps our understanding of the world and can lead to models of the phenomena we observe.

Connections of Data Mining with others

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems
- Traditional Techniques may be unsuitable due to
  - Enormity of data
  - High dimensionality of data
  - Heterogeneous, distributed nature of data
  - Emphasis on the use of data



Cultures

- Databases: concentrate on large-scale (non-main-memory) data.
- AI (machine-learning): concentrate on complex methods, small data.
  - In today's world data is more important than algorithms
- Statistics: concentrate on models.

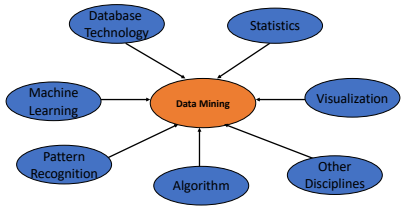
Models vs. Analytic Processing

- To a database person, data-mining is an extreme form of analytic processing – queries that examine large amounts of data.
  - Result is the query answer.
- To a statistician, data-mining is the inference of models.
  - Result is the parameters of the model.

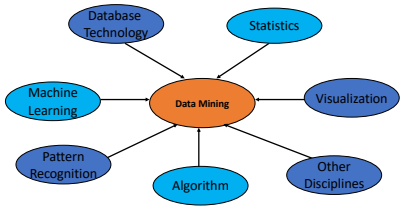
(Way too Simple) Example

- Given a billion numbers, a DB person would compute their average and standard deviation.
- A statistician might fit the billion points to the best Gaussian distribution and report the mean and standard deviation *of that distribution*.

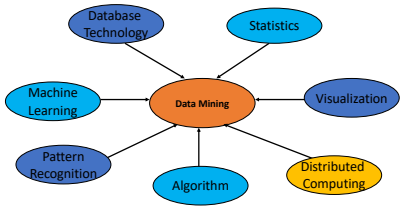
Data Mining: Confluence of Multiple Disciplines



Data Mining: Confluence of Multiple Disciplines



Data Mining: Confluence of Multiple Disciplines



The data analysis pipeline

- Mining is not the only step in the analysis process

- Preprocessing: real data is noisy, incomplete and inconsistent. Data cleaning is required to make sense of the data
  - Techniques: Sampling, Dimensionality Reduction, Feature selection.
  - A dirty work, but it is often the most important step for the analysis.
- Post-Processing: Make the data actionable and useful to the user
  - Statistical analysis of importance
  - Visualization.
- Pre- and Post-processing are often data mining tasks as well

Data Quality

- Examples of data quality problems:
  - Noise and outliers
  - missing values
  - duplicate data

Sampling

- Sampling is the main technique employed for data selection.
  - It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

Sampling …

- The key principle for effective sampling is the following:
  - using a sample will work almost as well as using the entire data sets, if the sample is representative
  - A sample is representative if it has approximately the same property (of interest) as the original set of data

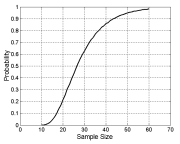
Types of Sampling

Simple Random Sampling	There is an equal probability of selecting any particular item
Sampling without replacement	As each item is selected, it is removed from the population
Sampling with replacement	Objects are not removed from the population as they are selected for the sample. In sampling with replacement, the same object can be picked up more than once
Stratified sampling	Split the data into several partitions; then draw random samples from each partition

Sample Size

8000 points 2000 Points 500 Points

Sample Size	What sample size is necessary to get at least one object from each of 10 groups.



CIS 530—Advanced Data Mining

4- Pre- and Post-Processing

Thomas W Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy to Prof. Panayiotis Tsaparas

What is Data Mining?

Data mining is the use of efficient techniques for the analysis of very large collections of data and the extraction of useful and possibly unexpected patterns in data.

"Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data analyst" (Hand, Mannila, Smyth)

"Data mining is the discovery of models for data" (Rajaraman, Ullman)

- We can have the following types of models
  - Models that explain the data (e.g., a single function)
  - Models that predict the future data instances.
  - Models that summarize the data
  - Models the extract the most prominent features of the data.

Why do we need data mining?

Really huge amounts of complex data generated from multiple sources and interconnected in different ways

- Scientific data from different disciplines
  - Weather, astronomy, physics, biological microarrays, genomics
- Huge text collections
  - The Web, scientific articles, news, tweets, Facebook postings.
- Transaction data
  - Retail store records, credit card records
- Behavioral data
  - Mobile phone data, query logs, browsing behavior, ad clicks
- Networked data
  - The Web, Social Networks, IM networks, email network, biological networks.
- All these types of data can be combined in many ways
  - Facebook has a network, text, images, user behavior, ad transactions.

We need to analyze this data to extract knowledge

- Knowledge can be used for commercial or scientific purposes.
- Our solutions should scale to the size of the data

The data analysis pipeline

- Mining is not the only step in the analysis process

Data Preprocessing → Data Mining → Result Post-processing

- Preprocessing: real data is noisy, incomplete and inconsistent. Data cleaning is required to make sense of the data
  - Techniques: Sampling, Dimensionality Reduction, Feature selection.
  - A dirty work, but it is often the most important step for the analysis.
- Post-Processing: Make the data actionable and useful to the user
  - Statistical analysis of importance
  - Visualization.
- Pre- and Post-processing are often data mining tasks as well

Data Quality

- Examples of data quality problems:
  - Noise and outliers
  - Missing values
  - Duplicate data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	10000K	Yes
6	No	NULL	60K	No
7	Yes	Divorced	220K	NULL
8	No	Single	85K	Yes
9	No	Married	90K	No
9	No	Single	90K	No

A mistake or a millionaire?

Missing values

Inconsistent duplicate entries

Sampling

Sampling is the main technique employed for data selection.

- It is often used for both the preliminary investigation of the data and the final data analysis.

Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.

- Example: What is the average height of a person in UMD?
  - We cannot measure the height of everybody

Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

- Example: We have 1M documents. What fraction has at least 100 words in common?
  - Computing number of common words for all pairs requires 10^{12} comparisons
- Example: What fraction of tweets in a year contain the word "Greece"?
  - 300M tweets per day, if 100 characters on average, 86.5TB to store all tweets

Sampling …

- The key principle for effective sampling is the following:
  - using a sample will work almost as well as using the entire data sets, if the sample is representative
  - A sample is representative if it has approximately the same property (of interest) as the original set of data
  - Otherwise, we say that the sample introduces some bias
  - What happens if we take a sample from the university campus to compute the average height of a person at UMD?

Types of Sampling

- Simple Random Sampling
  - There is an equal probability of selecting any particular item
- Sampling without replacement
  - As each item is selected, it is removed from the population
- Sampling with replacement
  - Objects are not removed from the population as they are selected for the sample.
  - In sampling with replacement, the same object can be picked up more than once. This makes analytical computation of probabilities easier
  - e.g., we have 100 people, 51 are women P(W) = 0.51, 49 men P(M) = 0.49. If I pick two persons what is the probability P(W,W) that both are women?
    - Sampling with replacement: $P(W,W) = 0.51^2$
    - Sampling without replacement: P(W,W) = 51/100 * 50/99

Types of Sampling

- Stratified sampling
  - Split the data into several groups; then draw random samples from each group.
    - Ensures that both groups are represented.
  - Example 1. I want to understand the differences between legitimate and fraudulent credit card transactions. 0.1% of transactions are fraudulent. What happens if I select 1000 transactions at random?
    - I get 1 fraudulent transaction (in expectation). Not enough to draw any conclusions. Solution: sample 1000 legitimate and 1000 fraudulent transactions

Probability Reminder: If an event has probability p of happening and I do N trials, the expected number of times the event occurs is pN

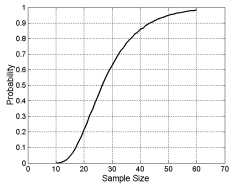
  - Example 2. I want to answer the question: Do web pages that are linked have on average more words in common than those that are not? I have 1M pages, and 1M links, what happens if I select 10K pairs of pages at random?
    - Most likely I will not get any links. Solution: sample 10K random pairs, and 10K links

Sample Size

8000 points 2000 Points 500 Points

Sample Size

- What sample size is necessary to get at least one object from each of 10 groups.



A data mining challenge

You have N integers, and you want to sample one integer uniformly at random. How do you do that?

The integers are coming in a stream: you do not know the size of the stream in advance, and there is not enough memory to store the stream in memory. You can only keep a constant number of integers in memory

How do you sample?

- Hint: if the stream ends after reading n integers the last integer in the stream should have probability 1/n to be selected.

Reservoir Sampling:

- Standard interview question for many companies

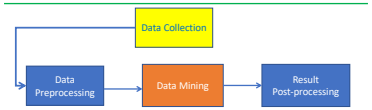
Reservoir sampling

- Algorithm: With probability 1/n select the n-th item of the stream and replace the previous choice.
- Claim: Every item has probability 1/N to be selected after N items have been read.
- Proof
  - What is the probability of the n-th item to be selected?
    - $\frac{1}{n}$
  - What is the probability of the n-th items to survive for N-n rounds?
    - $\left(1-\frac{1}{n+1}\right)\left(1-\frac{1}{n+2}\right)\cdots\left(1-\frac{1}{N}\right)$

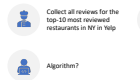
A (detailed) data preprocessing example

- Suppose we want to mine the comments/reviews of people on Yelp and Foursquare.

Data Collection



- Today there is an abundance of data online
  - Facebook, Twitter, Wikipedia, Web, etc…
- We can extract interesting information from this data, but first we need to collect it
  - Customized crawlers, use of public APIs
  - Additional cleaning/processing to parse out the useful parts
  - Respect of crawling etiquette



Mining Task

Example data

- I heard so many good things about this place so I was pretty juiced to try it. I'm from Cali and I heard Shake Shack is comparable to IN-N-OUT and I gotta say, Shake Shake wins hands down. Surprisingly, the line was short and we waited about 10 MIN. to order. I ordered a regular cheeseburger, fries and a black/white shake. So yummerz. I love the location too! It's in the middle of the city and the view is breathtaking. Definitely one of my favorite places to eat in NYC.
- I'm from California and I must say, Shake Shack is better than IN-N-OUT, all day, err'day.
- Would I pay $15+ for a burger here? No. But for the price point they are asking for, this is a definite bang for your buck (though for some, the opportunity cost of waiting in line might outweigh the cost savings) Thankfully, I came in before the lunch swarm descended and I ordered a shake shack (the special burger with the patty + fried cheese & portabella topping) and a coffee milk shake. The beef patty was very juicy and snugly packed within a soft potato roll. On the downside, I could do without the fried portabella-thingy, as the crispy taste conflicted with the juicy, tender burger. How does shake shack compare with in-and-out or 5-guys? I say a very close tie, and I think it comes down to personal affliations. On the shake side, true to its name, the shake was well churned and very thick and luscious. The coffee flavor added a tangy taste and complemented the vanilla shake well. Situated in an open space in NYC, the open air sitting allows you to munch on your burger while watching people zoom by around the city. It's an oddly calming experience, or perhaps it was the food coma I was slowly falling into. Great place with food at a great price.

First Cut

- Do simple processing to "normalize" the data (remove punctuation, make into lower case, clear white spaces, other?)
- Break into words, keep the most popular words

the 27514	the 16710	the 16010	the 14241
and 14508	and 9139	and 9504	and 8237
i 13088	a 8583	i 7966	a 8182
a 12152	i 8415	to 6524	i 7001
to 10672	to 7003	a 6370	to 6727
of 8702	in 5363	it 5169	of 4874
ramen 8518	it 4606	of 5159	you 4515
was 8274	of 4365	is 4519	it 4308
is 6835	is 4340	sauce 4020	is 4016
it 6802	burger 432	in 3951	was 3791
in 6402	was 4070	this 3519	pastrami 3748
for 6145	for 3441	was 3453	in 3508
but 5254	but 3284	for 3327	for 3424
that 4540	shack 3278	you 3220	sandwich 2928
you 4366	shake 3172	that 2769	that 2728
with 4181	that 3005	but 2590	but 2715
pork 4115	you 2985	food 2497	on 2247
my 3841	my 2514	on 2350	this 2099
this 3487	line 2389	my 2311	my 2064
wait 3184	this 2242	cart 2236	with 2040
not 3016	fries 2240	chicken 2220	not 1655
we 2984	on 2204	with 2195	your 1622
at 2980	are 2142	rice 2049	so 1610
on 2922	with 2095	so 1825	have 1585

First Cut

- Do simple processing to "normalize" the data (remove punctuation, make into lower case, clear white spaces, other?)
- Break into words, keep the most popular words

the 27514	the 16710	the 16010	the 14241
and 14508	and 9139	and 9504	and 8237
i 13088	a 8583	i 7966	a 8182
a 12152	i 8415	to 6524	i 7001
to 10672	to 7003	a 6370	to 6727
of 8702	in 5363	it 5169	of 4874
ramen 8518	it 4606	of 5159	you 4515
was 8274	of 4365	is 4519	it 4308
is 6835	is 4340	sauce 4020	is 4016
it 6802	burger 432	in 3951	was 3791
in 6402	was 4070	this 3519	pastrami 3748
for 6145	for 3441	was 3453	in 3508
but 5254	but 3284		
that 4540	shack 3278		
you 4366	shake 3172		
with 4181	that 3005	but 2590	but 2715
pork 4115	you 2985	food 2497	on 2247
my 3841	my 2514	on 2350	this 2099
this 3487	line 2389	my 2311	my 2064
wait 3184	this 2242	cart 2236	with 2040
not 3016	fries 2240	chicken 2220	not 1655
we 2984	on 2204	with 2195	your 1622
at 2980	are 2142	rice 2049	so 1610
on 2922	with 2095	so 1825	have 1585

Most frequent words are stop words

Second cut

- Remove stop words
- Stop-word lists can be found online.

a,about,above,after,again,against,all,am,an,and,any,are,aren't,as,at,be,be
cause,been,before,being,below,between,both,but,by,can't,cannot,could,could
n't,did,didn't,do,does,doesn't,doing,don't,down,during,each,few,for,from,f
urther,had,hadn't,has,hasn't,have,haven't,having,he,he'd,he'll,he's,her,he
re,here's,hers,herself,him,himself,his,how,how's,i,i'd,i'll,i'm,i've,if,in
,into,is,isn't,it,it's,its,itself,let's,me,more,most,mustn't,my,myself,no,
nor,not,of,off,on,once,only,or,other,ought,our,ours,ourselves,out,over,own
,same,shan't,she,she'd,she'll,she's,should,shouldn't,so,some,such,than,tha
t,that's,the,their,theirs,them,themselves,then,there,there's,these,they,th
ey'd,they'll,they're,they've,this,those,through,to,too,under,until,up,very
,was,wasn't,we,we'd,we'll,we're,we've,were,weren't,what,what's,when,when's
,where,where's,which,while,who,who's,whom,why,why's,with,won't,would,would
n't,you,you'd,you'll,you're,you've,your,yours,yourself,yourselves,

Second cut

- Remove stop words
- Stop-word lists can be found online.

ramen 8572	burger 4340	sauce 4023	pastrami 3782
pork 4152	shack 3291	food 2507	sandwich 2934
wait 3195	shake 3221	cart 2239	place 1480
good 2867	line 2397	chicken 2238	good 1341
place 2361	fries 2260	rice 2052	get 1251
noodles 2279	good 1920	hot 1835	katz's 1223
ippudo 2261	burgers 1643	white 1782	just 1214
buns 2251	wait 1508	line 1755	like 1207
broth 2041	just 1412	good 1629	meat 1168
like 1902	cheese 1307	lamb 1422	one 1071
just 1896	like 1204	halal 1343	deli 984
get 1641	food 1175	just 1338	best 965
time 1613	get 1162	get 1332	go 961
one 1460	place 1159	one 1222	ticket 955
really 1437	one 1118	like 1096	food 896
go 1366	long 1013	place 1052	sandwiches 813
food 1296	go 995	go 965	can 812
bowl 1272	time 951	can 878	beef 768
can 1256	park 887	night 832	order 720
great 1172	can 860	time 794	pickles 699
best 1167	best 849	long 792	time 662
		people 790	

Second cut

- Remove stop words
- Stop-word lists can be found online.

ramen 8572	burger 4340	sauce 4023	pastrami 3782
pork 4152	shack 3291	food 2507	sandwich 2934
wait 3195	shake 3221	cart 2239	place 1480
good 2867	line 2397	chicken 2238	good 1341
place 2361	fries 2260	rice 2052	get 1251
noodles 2279	good 1920	hot 1835	katz's 1223
ippudo 2261	burgers 1643	white 1782	just 1214
buns 2251	wait 1508	line 1755	like 1207
broth 2041	just 1412	good 1629	meat 1168
like 1902	cheese 1307	lamb 1422	one 1071
just 1896	like 1204	halal 1343	deli 984
get 1641	food 1175	just 1338	best 965
time 1613	get 1162	get 1332	go 961
one 1460	place 1159	one 1222	ticket 955
really 1437	one 1118	like 1096	food 896
go 1366	long 1013	place 1052	sandwiches 813
food 1296	go 995	go 965	can 812
bowl 1272			
can 1256			
great 1172			
best 1167	best 849	long 792	time 662
		people 790	

Commonly used words in reviews, not so interesting

IDF

- Important words are the ones that are unique to the document (differentiating) compared to the rest of the collection
  - All reviews use the word "like". This is not interesting
  - We want the words that characterize the specific restaurant
- Document Frequency $DF(w)$: fraction of documents that contain word w.

$$DF(w) = \frac{D(w)}{D}$$

$D(w)$: num of docs that contain word w
D: total number of documents

- Inverse Document Frequency $IDF(w)$:

$$IDF(w) = \log\left(\frac{1}{DF(w)}\right)$$

- Maximum when unique to one document : $IDF(w) = \log(D)$
- Minimum when the word is common to all documents: $IDF(w) = 0$

TF-IDF

The words that are best for describing a document are the ones that are important for the document, but also unique to the document.

TF(w,d): term frequency of word w in document d

- Number of times that the word appears in the document
- Natural measure of importance of the word for the document

IDF(w): inverse document frequency

- Natural measure of the uniqueness of the word w

TF-IDF(w,d) = TF(w,d) × IDF(w)

Third cut

- Ordered by TF-IDF

ramen 3057.4176194	fries 806.08537330	lamb 985.655290756243	pastrami 1931.94250908298 6
akamaru 2353.24196	custard 729.607519	halal 686.038812717726	katz's 1120.62356508209 4
noodles 1579.68242	shakes 628.4738038	53rd 375.685771863491	rye 1004.28925735888 2
broth 1414.7133955	shroom 515.7790608	gyro 305.809092298788	corned 906.113544700399 2
miso 1252.60629058	burger 457.2646379	pita 304.984759446376	pickles 640.487221580035 4
hirata 709.1962086	crinkle 398.347221	cart 235.902194557873	reuben 515.779060830666 1
hakata 591.7643688	burgers 366.624854	platter 139.4599030800	matzo 430.583412389887 1
shiromaru 587.1591	madison 350.939350	chicken/lamb 135.85252	sally 428.110484707471 2
noodle 581.8446147	shackburger 292.42	carts 120.274374158359	harry 226.323810772916 4
tonkotsu 529.59457	'shroom 287.823136	hilton 84.298747332422	mustard 216.079238853014 6
ippudo 504.5275695	portobello 239.806	lamb/chicken 82.893063	cutter 209.535243462458 1
buns 502.296134008	custards 211.83782	yogurt 70.007865236554	carnegie 198.655512713779 3
ippudo's 453.60926	concrete 195.16992	52nd 67.5963923222322	katz 194.387844446609 7
modern 394.8391629	bun 186.9621782983	6th 60.7930175345658	knish 184.206807439524 1
egg 367.3680056967	milkshakes 174.996	4am 55.4517744447956	sandwiches 181.415707218 8
shoyu 352.29551922	concretes 165.7861	yellow 54.447026520667	brisket 131.945865389878 4
chashu 347.6903490	portabello 163.483	tzatziki 52.9594571388	fries 131.613054313392 7
karaka 336.1774235	shack's 159.334353	lettuce 51.3230168022	salami 127.621117258549 3
kakuni 276.3102111	patty 152.22603588	sammy's 50.6568720458	knishes 124.339595021678 1
ramens 262.4947006	ss 149.66803104461	sw 50.5668577816893 3	delicatessen 117.488967607 2
bun 236.5122638036	patties 148.068287	platters 49.9065970003	deli's 117.431839742696 1
wasabi 232.3667512	cam 105.9496067806	falafel 49.4796995212	carver 115.129254649702 1
dama 221.048168927	milkshake 103.9720	sober 49.2211422635451	brown's 109.441778045519 2
brulee 201.1797390	lamps 99.011158998	moma 48.1589121730374	matzoh 108.22149937072 1

Third cut

TF-IDF takes care of stop words as well

We do not need to remove the stopwords since they will get IDF(w) = 0

Decisions, decisions…

- When mining real data you often need to ask:
  - What data should we collect? How much? For how long?
  - Should we throw out some data that does not seem to be useful?

An actual review: AAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAA AAA

  - Too frequent data (stop words), too infrequent (errors?), erroneous data, missing data, outliers
  - How should we weight different pieces of data?
- Most decisions are application dependent. Some information may be lost but we can usually live with it (most of the times)
- We should make our decisions clear since they affect our findings.
- Dealing with real data is hard…

Exploratory analysis of data

- Summary statistics: numbers that summarize properties of the data
  - Summarized properties include frequency, location and spread
    - Examples: location - mean; spread - standard deviation
  - Most summary statistics can be calculated in a single pass through the data

Frequency and Mode

The frequency of an attribute value is the percentage of time the value occurs in the data set

- For example, given the attribute 'gender' and a representative population of people, the gender 'female' occurs about 50% of the time.

The mode of an attribute is the most frequent attribute value

The notions of frequency and mode are typically used with categorical data

Percentiles

- For continuous data, the notion of a percentile is more useful.
- Given an ordinal or continuous attribute x and a number p between 0 and 100, the p^{th} percentile is a value x_p of x such that $p\%$ of the observed values of x are less than x_p.
- For instance, the 50th percentile is the value $x_{50\%}$ such that 50% of all values of x are less than $x_{50\%}$.

Measures of Location: Mean and Median

- The mean is the most common measure of the location of a set of points.
- However, the mean is very sensitive to outliers.
- Thus, the median or a trimmed mean is also commonly used.

$$\text{mean}(x) = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

Example

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	10000K	Yes
6	No	NULL	60K	No
7	Yes	Divorced	220K	NULL
8	No	Single	85K	Yes
9	No	Married	90K	No
10	No	Single	90K	No

Mean: 1090K

Trimmed mean (remove min, max): 105K

Median: (90+100)/2 = 95K

Measures of Spread: Range and Variance

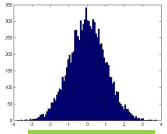
- Range is the difference between the max and min
- The variance or standard deviation is the most common measure of the spread of a set of points.

$$\text{var}(x) = \frac{1}{m}\sum_{i=1}^{m}(x - \bar{x})^2$$

$$\sigma(x) = \sqrt{\text{var}(x)}$$

Normal Distribution

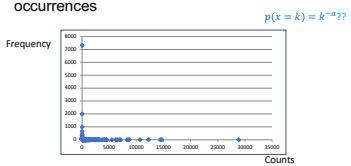
- $\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$



- An important distribution that characterizes many quantities and has a central role in probabilities and statistics.
  - Appears also in the central limit theorem
- Fully characterized by the mean μ and standard deviation σ

Not everything is normally distributed

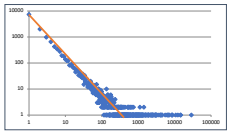
- Plot of y number of words with x number of occurrences



- If this was a normal distribution, we would not have a count as large as 28K

Power-law distribution

- We can understand the distribution of words if we take the log-log plot

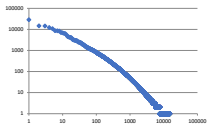


- Linear relationship in the log-log space for:

$$p(x = k) = k^{-a}$$

Zipf's law

- Power laws can be detected by a linear relationship in the log-log space for the rank-frequency plot

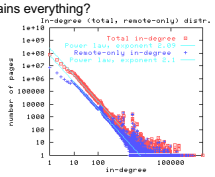


- $f(r)$: Frequency of the r-th most frequent word

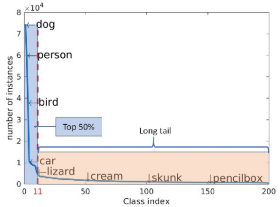
$$f(r) = r^{-\beta}$$

Power-laws are everywhere

- Incoming and outgoing links of web pages, number of friends in social networks, number of occurrences of words, file sizes, city sizes, income distribution, popularity of products and movies
  - Signature of human activity?
  - A mechanism that explains everything?
  - Rich get richer process



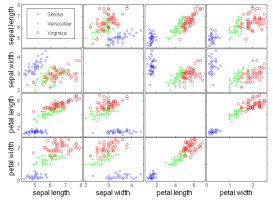
The Long Tail



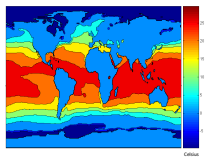
Post-processing

- Visualization
  - The human eye is a powerful analytical tool
  - If we visualize the data properly, we can discover patterns
  - Visualization is the way to present the data so that patterns can be seen
    - E.g., histograms and plots are a form of visualization
    - There are multiple techniques (a field on its own)

Scatter Plot Array of Iris Attributes



Contour Plot Example: SST Dec, 1998



Meaningfulness of Answers

A big data-mining risk is that you will "discover" patterns that are meaningless.

Statisticians call it Bonferroni's principle: (roughly) if you look in more places for interesting patterns than your amount of data will support, you are bound to find crap.

The Rhine Paradox: a great example of how not to conduct scientific research.

Rhine Paradox – (1)

Joseph Rhine was a parapsychologist in the 1950's who hypothesized that some people had Extra-Sensory Perception (ESP).

He devised (something like) an experiment where subjects were asked to guess 10 hidden cards – red or blue.

He discovered that almost 1 in 1000 had ESP – they were able to get all 10 right!

Rhine Paradox – (2)

He told these people they had ESP and called them in for another test of the same type.

Alas, he discovered that almost all of them had lost their ESP.

What did he conclude?

- Answer on next slide.

Rhine Paradox – (3)

- He concluded that you shouldn't tell people they have ESP; it causes them to lose it.
- The facts behind…

$$\begin{aligned} P(\text{at least one wins}) &= 1 - P(\text{no one wins}) \\ &= 1 - \left(1 - \frac{1}{2}^{10}\right)^{1000} \\ &\approx 0.6235762 \end{aligned}$$

CIS 530—Advanced Data Mining

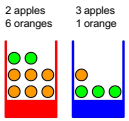
5- Probability Theory

Thomas W Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy to Prof. Sargur N. Srihari

Probabilities of Interest

- Marginal Probability
  - What is the probability of an apple?
- Conditional Probability
  - Given that we have an orange what is the probability that we chose the blue box?
- Joint Probability
  - What is the probability of orange AND blue box?

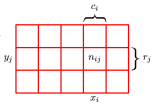


Box is random variable B (has values *r* or *b*)
Fruit is random variable F (has values *o* or *a*)

Let $p(B=r)=4/10$ and $p(B=b)=6/10$

Sum Rule of Probability Theory

- Consider two random variables
  - X can take on values $x_i, i=1,..,M$
  - Y can take on values $y_j, j=1,..,L$
  - N trials sampling both X and Y
  - No. of trials with $X=x_i$ and $Y=y_j$ is n_{ij}



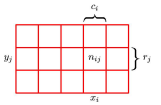
Joint Probability $p(X=x_i, Y=y_j)=\frac{n_{ij}}{N}$

$p(X=x_i)=\frac{c_i}{N}$

Since $c_i=\sum_j n_{ij}$ $\quad p(X=x_i)=\sum_{j=1}^{3} p(X=x_i,Y=y_j)$

Product Rule of Probability Theory

- Consider only those instances for which $X=x_i$
- Then fraction of those instances for which $Y=y_j$ is written as $p(Y=y_j|X=x_i)$
- Called conditional probability
- Relationship between joint and conditional probability:

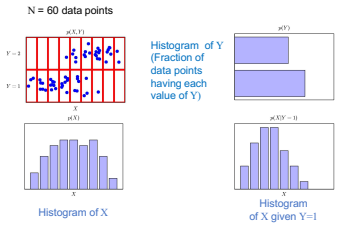


$$p(Y=y_j|X=x_i)=\frac{n_{ij}}{c_i}$$

$$p(X=x_i,Y=y_j)=\frac{n_{ij}}{N}=\frac{n_{ij}}{c_i}\times\frac{c_i}{N}$$

$$=p(Y=y_j|X=x_i)\,p(X=x_i)$$

Joint Distribution over Two Variables



Fractions would equal the probability as N →oo

Bayes Theorem

- Think about relation between mid-term score and final grade:
  - Everyone wants to know, if my mid-term is in the range [80-90], what is the probability of getting A?
  - Assume the range of score is represented by random variable X, and final grade is by Y
  - So, we are modeling: $p(Y=A|X\in[80,90])$
- Below is something we may know from the course records of year 2018:
  - $p(Y=A)=0.3;$
  - $p(X\in[80,90])=0.2$
  - $p(X\in[80,90]|Y=A)=0.1$

Bayes Theorem

- From the product rule together with the symmetry property $p(X,Y)=p(Y,X)$ we get

$$p(Y|X)=\frac{p(X|Y)p(Y)}{p(X)}$$

Normalization constant to ensure sum of conditional probability equal to 1 over all values of Y

- which is called Bayes' theorem
- Sum and product rule for $p(X)$

$$p(X)=\sum_Y p(X|Y)p(Y)$$

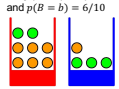
Bayes rule applied to Fruit Problem

- Probability that box is red given that fruit picked is orange

$$p(B=r\,|\,F=o)=\frac{p(F=o\,|\,B=r)p(B=r)}{p(F=o)}$$

$$=\frac{\frac{3}{4}\times\frac{4}{10}}{\frac{9}{20}}=\boxed{\frac{2}{3}=0.66}$$

Let $p(B=r)=4/10$ and $p(B=b)=6/10$



The *a posteriori* probability of 0.66 is different from the *a priori* probability of 0.4

- Probability that fruit is orange
  - From sum and product rules

$$p(F=o)=p(F=o,B=r)+p(F=o,B=b)$$
$$=p(F=o\,|\,B=r)p(B=r)+p(F=o\,|\,B=b)p(B=b)$$

$$=\frac{6}{8}\times\frac{4}{10}+\frac{1}{4}\times\frac{6}{10}=\boxed{\frac{9}{20}=0.45}$$

The *marginal* probability of 0.45 is lower than average probability of 7/12=0.58

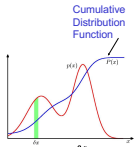
Independent Variables

- If $p(X,Y)=p(X)p(Y)$ then X and Y are said to be independent
- Why?
- From product rule $\quad p(Y|X)=\frac{p(X,Y)}{P(X)}=p(Y)$
- In fruit example if each box contained same fraction of apples and oranges then $p(F|B)=p(F)$

Probability Densities Function (PDF)

- Continuous Variables
  - If probability that x falls in interval $(x, x+\delta x)$ is given by $p(x)dx$ for $\delta x\to 0$
  - then $p(x)$ is a Probability Density Function (PDF) of x
- Probability x lies in interval (a,b) is

$$p(x\in(a,b))=\int_a^b p(x)dx$$



$$P(z)=\int_{-\infty}^{z} p(x)dx$$

Probability that x lies in Interval $(-oo,z)$ is

Several Variables

- If there are several continuous variables $x_1,\ldots,x_D$ denoted by vector x then we can define a joint probability density

$$p(\mathbf{x})=p(x_1,x_2,\ldots,x_D)$$

- Multivariate probability density must satisfy

$$\int_{-\infty}^{\infty} p(\mathbf{x})d\mathbf{x}=1 \qquad p(\mathbf{x})\geq 0$$

Sum, Product, Bayes for Continuous

- Rules apply for continuous, or combinations of discrete and continuous variables

Marginalization $p(x)=\int p(x,y)dy$

Product $p(x,y)=p(y|x)p(x)$

Bayes $p(y|x)=\frac{p(x|y)p(y)}{p(x)}$

- Formal justification of sum, product rules for continuous variables requires measure theory

Expectation: an Example

- Assume the final grade is represented by a random variable X
- Blow is the marginal probability of X in class
  - $p(X=A)=0.3; p(X=B)=0.4; p(X=C)=0.3$
- We further assume there is a mapping f between final grade and future salary level
  - $f(A)=100K; f(B)=90K; f(C)=80K$
- Question: what is the expected salary level of this class?
  - $100K*0.3+90K*0.4+80K*0.3=90K$

Expectation of $f(x)$

- Expectation $E[f]$ is *average* value of some function $f(x)$ under the probability distribution $p(x)$
- For a discrete distribution: $E[f]=\sum p(x)f(x)$
- For a continuous distribution: $E[f]=\int p(x)f(x)dx$
- If there are N points drawn from a PDF, then expectation can be approximated as: $E[f]=\frac{1}{N}\sum f(x)$

Variance of $f(x)$ and x

- Measures how much variability there is in $f(x)$ around its mean value $E[f(x)]$
- Variance of $f(x)$ is denoted as

$$var[f]=E[f(x)-E[f(x)]]^2$$

- Expanding the square, we have

$$var[f]=E[f^2(x)]-E[f(x)]^2$$

- Variance of the variable x itself

$$var[x]=E[x^2]-E[x]^2$$

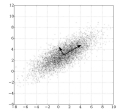
Covariance

- For two random variables x and y covariance is defined as

$$cov[x,y]=E_{xy}(x-E[x])(y-E[y])$$
$$=E_{xy}[xy]-E[x]E[y]$$

- Expresses how x and y vary together
- If x and y are independent, then their covariance vanishes
- If we consider covariance of components of vector x with each other then we denote it as

$$var[x]=cov[x,x]$$

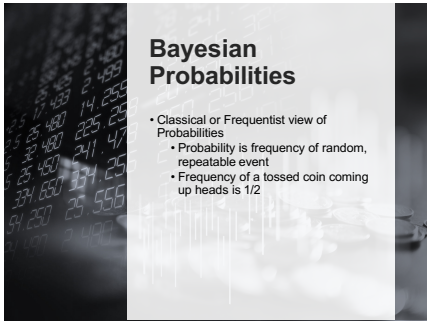


Why??

Covariance Matrix

- If x and y are two vectors of random variables, then the covariance is a matrix
- Example:
  - assume random variable vectors $x = [x_1, x_2, x_3, x_4]; y = [y_1, y_2, y_3, y_4]$

$$cov(x,y) = \begin{pmatrix} cov(x_1,y_1) & cov(x_1,y_2) & cov(x_1,y_3) & cov(x_1,y_4) \\ cov(x_2,y_1) & cov(x_2,y_2) & cov(x_2,y_3) & cov(x_2,y_4) \\ cov(x_3,y_1) & cov(x_3,y_2) & cov(x_3,y_3) & cov(x_3,y_4) \\ cov(x_4,y_1) & cov(x_4,y_2) & cov(x_4,y_3) & cov(x_4,y_4) \end{pmatrix}$$



Bayesian Probabilities

Bayesian View

- Probability is a quantification of uncertainty
- Degree of belief in propositions that do not involve random variables

Examples of uncertain events as probabilities:

- Whether Shakespeare's plays were written by Francis Bacon
- Whether moon was once in its own orbit around the sun
- Whether Thomas Jefferson had a child by one of his slaves
- Whether a signature on a check is genuine

Examples of Uncertain Events

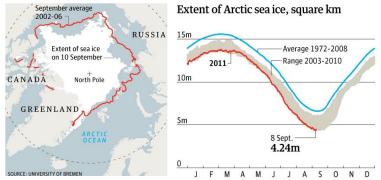
Probability that Mr. M was the murderer of Mrs. M given the evidence

Whether Arctic ice cap will disappear by end of century

- We have some idea of how quickly polar ice is melting
- Revise it on the basis of fresh evidence (satellite observations)
- Assessment will affect actions we take (to reduce greenhouse gases)

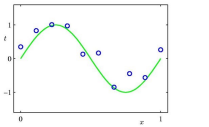
All can be achieved by general Bayesian interpretation

Arctic Ice (2011)



Bayesian Approach

- Quantify uncertainty around choice of parameters
  - E.g., $\mathbf{w}$ is vector of parameters in curve fitting



$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Uncertainty before observing data expressed by $p(\mathbf{w})$

Bayesian Approach

- Given observed data $D = \{t_1, .. t_N\}$
- Uncertainty in $\mathbf{w}$ after observing D, by Bayes rule:

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)}$$

- Quantity $p(D|\mathbf{w})$ can be viewed as a function of $\mathbf{w}$
- Represent how probable the data set is for different parameters $\mathbf{w}$ Called **Likelihood function**
- Not a probability distribution over $\mathbf{w}$

Role of Likelihood Function

- Uncertainty in $\mathbf{w}$ expressed as

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \quad \text{where} \quad p(D) = \int p(D|\mathbf{w})p(\mathbf{w})d\mathbf{w}$$

- Denominator is normalization factor
- Involves marginalization over $\mathbf{w}$
- Bayes theorem in words

$$posterior \propto likelihood \times prior$$

Role of Likelihood Function

- Likelihood Function plays central role in both Bayesian and frequentist paradigms
  - Frequentist: $\mathbf{w}$ is a **fixed** parameter determined by an estimator; error bars on estimate are obtained from possible data sets $\mathcal{D} = \{D_1, .. D_L\}$
  - Bayesian: there is a single data set D; uncertainty is expressed as **probability distribution** over $\mathbf{w}$

Maximum Likelihood Approach

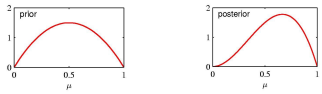
- In frequentist setting, $\mathbf{w}$ is considered to be a fixed parameter
  - $\mathbf{w}$ is set to value that maximizes likelihood function $p(D|\mathbf{w})$
- In ML, negative log of likelihood function is called error function, since maximizing likelihood is equivalent to minimizing error

Maximum Likelihood Approach

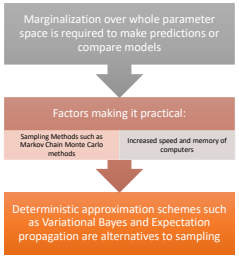
- Bootstrap approach to creating L data sets
  - From N data points new data sets are created by drawing n points at random with replacement
  - Repeat L times to generate L data sets
  - Accuracy of parameter estimate can be evaluated by variability of predictions between different bootstrap sets

Bayesian vs. Frequentist Approach

- Inclusion of prior knowledge arises naturally
- Coin Toss Example
  - Fair looking coin is tossed three times and lands Head each time
  - Classical MLE of the probability of landing heads is 1 implying all future tosses will land Heads
  - Bayesian approach with reasonable prior will lead to less extreme conclusion



Practicality of Bayesian Approach

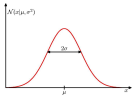


The Gaussian Distribution

- For single real-valued variable x

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

- Parameters:
  - Mean μ, variance σ^2
  - Standard deviation σ
  - Precision = $\frac{1}{\sigma^2}$
  - $E[x] = \mu$
  - $var[x] = \sigma^2$

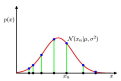


Maximum of a distribution is its mode
For a Gaussian, mode coincides with its mean

Likelihood Function for Gaussian

- Given N observations $x_i, i = 1, .., n$
- Independent and identically distributed (i.i.d.)
- Probability of data set is given by likelihood function

$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu,\sigma^2).$$



Data: black points
Likelihood= product of blue values. Pick mean and variance to maximize this product

- Log-likelihood function is

$$\ln p(\mathbf{x}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

Likelihood Function for Gaussian

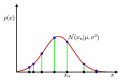
- Log-likelihood function is

$$\ln p(\mathbf{x}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

- Maximum likelihood solutions are given by:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\sigma^2_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2$$



Data: black points
Likelihood= product of blue values. Pick mean and variance to maximize this product

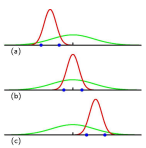
Bias in Maximum Likelihood

- Maximum likelihood systematically underestimates variance

$$E[\mu_{ML}] = \mu$$

$$E[\sigma^2_{ML}] = \left(\frac{N-1}{N}\right)\sigma^2$$

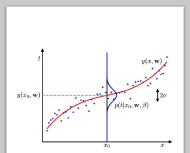
- Not an issue as N increases
- Problem is related to overfitting problem



Averaged across three data sets
mean is correct
Variance is underestimated
because it is estimated relative
to sample mean and not true mean

Curve Fitting (Probabilistic)

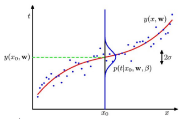
- Goal is to predict for target variable t given a new value of the input variable x
- Given N input values $\mathbf{x} = (x_1, .., x_N)^T$ and corresponding target values $\mathbf{t} = (t_1, .., t_N)^T$



Gaussian conditional distribution
for t given x.
Mean is given by
polynomial function $y(x, \mathbf{w})$
Precision given by β

Curve Fitting (Probabilistic)

- Assume given value of x, value of t has a Gaussian distribution with mean equal to $y(x, \mathbf{w})$ of polynomial curve



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right)$$

Gaussian conditional distribution
for t given x.
Mean is given by
polynomial function $y(x, \mathbf{w})$
Precision given by β

Curve Fitting with Maximum Likelihood

- Likelihood Function is

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n|y(x_n, \mathbf{w}), \beta^{-1}\right).$$

- Logarithm of the Likelihood function is

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$

- To find maximum likelihood solution for polynomial coefficients $\mathbf{w}_{ML}$
- Maximize w.r.t. $\mathbf{w}$

Curve Fitting with Maximum Likelihood

- Can omit last two terms – don't depend on $\mathbf{w}$
- Can replace $\beta/2$ with ½
  - since it is constant w.r.t. $\mathbf{w}$
- Minimize negative log-likelihood
- Identical to sum-of-squares error function

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$

Precision Parameter with MLE

- Maximum likelihood can also be used to determine β of Gaussian conditional distribution
- Maximizing likelihood w.r.t. β gives

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\}^2.$$

- First determine parameter vector $\mathbf{w}_{ML}$ governing the mean and subsequently use this to find precision β_{ML}

Predictive Distribution

- Knowing parameters $\mathbf{w}$ and β
- Predictions for new values of x can be made using

- Instead of a point estimate we are now giving a probability distribution over t

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right).$$

CIS 530—Advanced Data Mining

6- Decision Trees

Thomas W Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy to Prof. Panayiotis Tsaparas

Catching tax-evasion

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tax-return data for year 2011

A new tax return for 2012
Is this a cheating tax return?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

An instance of the classification problem: learn a method for discriminating between records of different classes (cheaters vs non-cheaters)

What is classification?

- **Classification** is the task of *learning* ***a target function*** f that maps attribute set x to one of the predefined class labels y

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(categorical, categorical, continuous, class)

One of the attributes is the class attribute
In this case: Cheat

Two class labels (or classes): Yes (1), No (0)

Figure 4.2. Classification as the task of mapping an input attribute set x into its class label y.

Why classification?

- The target function f is known as a classification model
- Descriptive modeling: Explanatory tool to distinguish between objects of different classes (e.g., understand why people cheat on their taxes)
- Predictive modeling: Predict a class of a previously unseen record

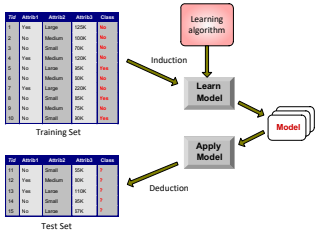
Examples of Classification Tasks

Predicting	Predicting tumor cells as benign or malignant
Classifying	Classifying credit card transactions as legitimate or fraudulent
Categorizing	Categorizing news stories as finance, weather, entertainment, sports, etc.
Identifying	Identifying spam email, spam web pages, adult content
Understanding	Understanding if a web query has commercial intent or not

General approach to classification

- Training set consists of records with known class labels
- Training set is used to build a classification model
- A labeled test set of previously unseen data records is used to evaluate the quality of the model.
- The classification model is applied to new records with unknown class labels

Illustrating Classification Task



Evaluation of classification models

- Counts of test records that are correctly (or incorrectly) predicted by the classification model
- **Confusion matrix**

Actual Class \ Predicted Class	Class = 1	Class = 0
Class = 1	f_{11}	f_{10}
Class = 0	f_{01}	f_{00}

$$\text{Accuracy} = \frac{\#\text{ correct predictions}}{\text{total }\#\text{ of predictions}} = \frac{f_{11}+f_{00}}{f_{11}+f_{10}+f_{01}+f_{00}}$$

$$\text{Error rate} = \frac{\#\text{ wrong predictions}}{\text{total }\#\text{ of predictions}} = \frac{f_{10}+f_{01}}{f_{11}+f_{10}+f_{01}+f_{00}}$$

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Decision Tree

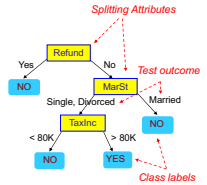
- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution

Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(categorical, categorical, continuous, class)

Training Data

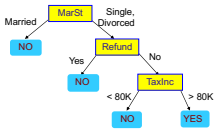


Model: Decision Tree

Another Example of Decision Tree

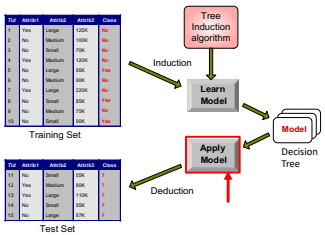
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(categorical, categorical, continuous, class)



There could be more than one tree that fits the same data!

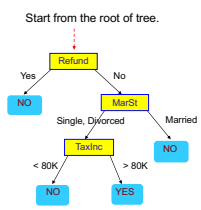
Decision Tree Classification Task



Apply Model to Test Data

Test Data

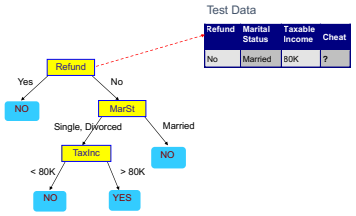
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

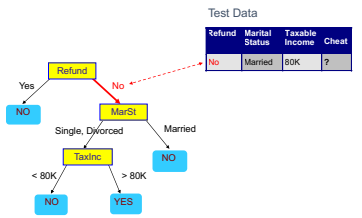
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



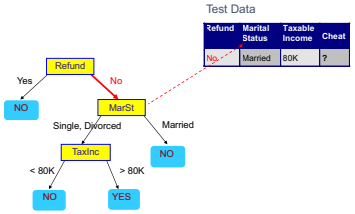
Apply Model to Test Data

Test Data

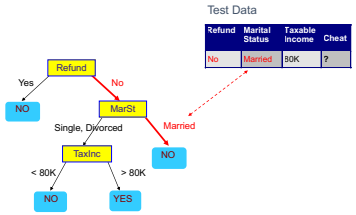
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



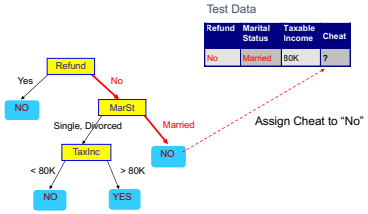
Apply Model to Test Data



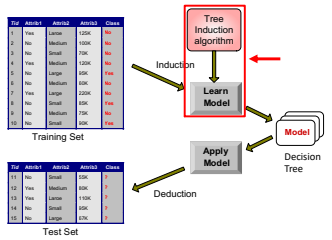
Apply Model to Test Data



Apply Model to Test Data



Decision Tree Classification Task



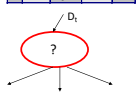
Tree Induction

- Finding the best decision tree is NP-hard
- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ,SPRINT

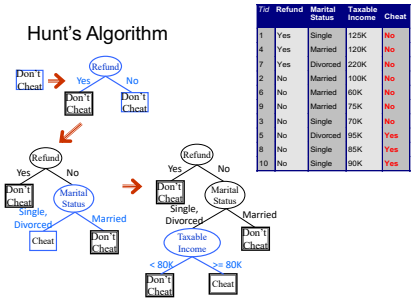
General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- General Procedure:
  1. If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  2. If D_t contains records with the same attribute values, then t is a leaf node labeled with the majority class y_t
  3. If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
  4. If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.
- Recursively apply the procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt's Algorithm



Tree Induction

- How to **Classify** a leaf node
  - Assign the majority class
  - If leaf is empty, assign the default class – the class that has the highest popularity.
- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

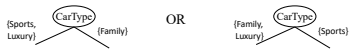
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- Multi-way split: Use as many partitions as distinct values.

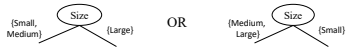
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



Splitting Based on Ordinal Attributes

- Multi-way split: Use as many partitions as distinct values.

- Binary split: Divides values into two subsets – respects the order. Need to find optimal partitioning.

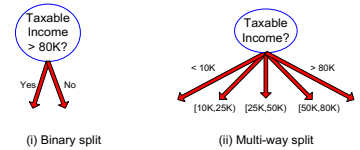


- What about this split?

Splitting Based on Continuous Attributes

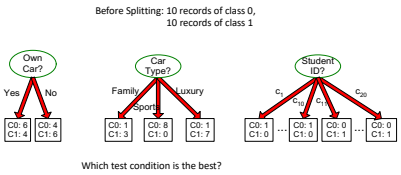
- Different ways of handling:
- Discretization to form an ordinal categorical attribute
  - Static – discretize once at the beginning
  - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
- Binary Decision: $(A < v)$ or $(A \geq v)$
  - consider all possible splits and finds the best cut
  - can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split (ii) Multi-way split

How to determine the Best Split



Which test condition is the best?

How to determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

- Ideas?

Measuring Node Impurity

- $p(i|t)$: fraction of records associated with node t belonging to class i

$$\text{Entropy}(t) = -\sum_{i=1}^{c} p(i \mid t) \log p(i \mid t)$$

- Used in ID3 and C4.5

$$\text{Gini}(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

- Used in CART, SLIQ, SPRINT.

$$\text{Classification error}(t) = 1 - \max_i [p(i \mid t)]$$

Gain

- ***Gain of an attribute split:*** compare the impurity of the parent node with the average impurity of the child nodes

$$\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

- Maximizing the gain ⇔ Minimizing the weighted average impurity measure of children nodes
- If $I()$ = **Entropy(),** then Δ_{info} is called **information gain**

Example

C1	0
C2	6

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1
Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0
Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0
Error = 1 – max (0, 1) = 1 – 1 = 0

C1	1
C2	5

P(C1) = 1/6 P(C2) = 5/6
Gini = 1 – (1/6)² – (5/6)² = 0.278
Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65
Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	2
C2	4

P(C1) = 2/6 P(C2) = 4/6
Gini = 1 – (2/6)² – (4/6)² = 0.444
Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92
Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

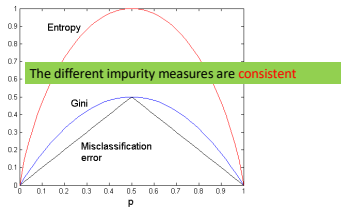
Impurity measures

All of the impurity measures take value zero (minimum) for the case of a pure node where a single value has probability 1

All of the impurity measures take maximum value when the class distribution in a node is uniform.

Comparison among Splitting Criteria

For a 2-class problem:



The different impurity measures are consistent

Categorical Attributes

- For binary values split in two
- For multivalued attributes, for each distinct value, gather counts for each class in the dataset
  - Use the count matrix to make decisions

Multi-way split

CarType	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split (find best partition of values)

CarType	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

CarType	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Continuous Attributes

- Use Binary Decisions based on one value
- Choices for the splitting value
  - Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
  - Class counts in each of the partitions, A < v and A ≥ v
- Exhaustive method to choose best v
  - For each v, scan the database to gather count matrix and compute the impurity index
  - Computationally Inefficient! Repetition of work.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Continuous Attributes

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing impurity
  - Choose the split position that has the least impurity

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income	60	70	75	85	90	95	100	120	125	220
Sorted Values →	55	65	72	80	87	92	97	110	122	172	230
Split Positions →	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	0.420

Splitting based on impurity

- Impurity measures favor attributes with large number of values (splits)
- A test condition with large number of outcomes may not be desirable
- # of records in each partition is too small to make predictions

Splitting based on INFO

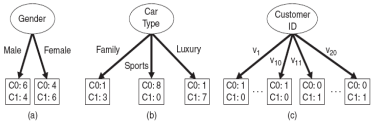


Figure 4.12. Multiway versus binary splits.

Gain Ratio

- Splitting using information gain

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions
n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5
- Designed to overcome the disadvantage of impurity

Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)

Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets

Example: C4.5

- Simple depth-first construction.
- Uses Information Gain
- Sorts Continuous Attributes at each node.
- Needs entire data to fit in memory.
- Unsuitable for Large Datasets.
  - Needs out-of-core sorting.
- You can download the software from: http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz

Other Issues

- Data Fragmentation
- Expressiveness

Data Fragmentation

01 Number of instances gets smaller as you traverse down the tree

02 Number of instances at the leaf nodes could be too small to make any statistically significant decision

03 You can introduce a lower bound on the number of items per leaf node in the stopping criterion.

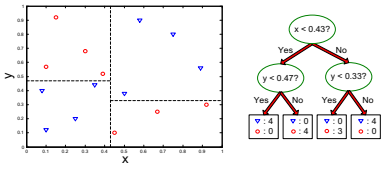
Expressiveness

A classifier defines a function that discriminates between two (or more) classes.

The expressiveness of a classifier is the class of functions that it can model, and the kind of data that it can separate

- When we have discrete (or binary) values, we are interested in the class of boolean functions that can be modeled
- If the data-points are real vectors we talk about the decision boundary that the classifier can model

Decision Boundary

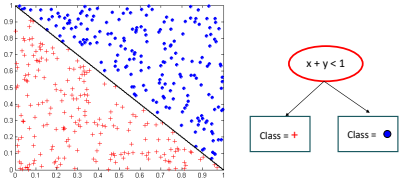


- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Expressiveness

- Decision tree provides expressive representation for learning discrete-valued function
- But they do not generalize well to certain types of Boolean functions
- Example: parity function:
  - Class = 1 if there is an even number of Boolean attributes with truth value = True
  - Class = 0 if there is an odd number of Boolean attributes with truth value = True
  - For accurate modeling, must have a complete tree
- Less expressive for modeling continuous variables
  - Particularly when test condition involves only a single attribute at-a-time

Oblique Decision Trees

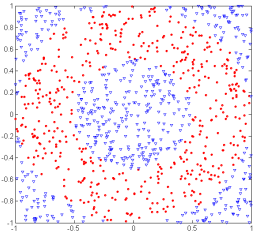


- Test condition may involve multiple attributes
- More expressive representation
- Finding optimal test condition is computationally expensive

Practical Issues of Classification

- Underfitting and Overfitting
- Evaluation

Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

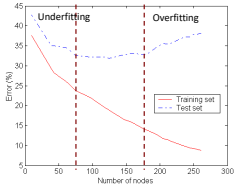
$0.5 \le \text{sqrt}(x_1^2+x_2^2) \le 1$

Triangular points:

$\text{sqrt}(x_1^2+x_2^2) < 0.5$ or

$\text{sqrt}(x_1^2+x_2^2) > 1$

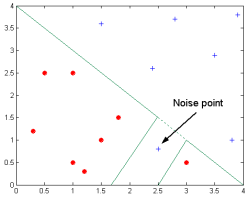
Underfitting and Overfitting



Underfitting: when model is too simple, both training and test errors are large

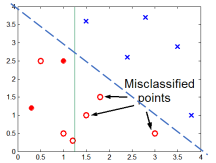
Overfitting: when model is too complex it models the details of the training set and fails on the test set

Underfitting and Overfitting



Decision boundary is distorted by noise point

Overfitting due to Insufficient Examples



Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region

Insufficient number of training records in the region causes the decision tree to predict the test examples using other training records that are irrelevant to the classification task

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
  - The model does not generalize well
- Need new ways for estimating errors

Estimating Generalization Errors

- Re-substitution errors: error on training ($\sum e(t)$)
- Generalization errors: error on testing ($\sum e'(t)$)

- Methods for estimating generalization errors:
- Optimistic approach: $e'(t) = e(t)$
- Pessimistic approach:
- For each leaf node: $e'(t) = (e(t) + 0.5)$
  - Total errors: $e'(T) = e(T) + N \times 0.5$ (N: number of leaf nodes)
  - Penalize large trees
- For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances)
  - Training error = 10/1000 = 1
  - Generalization error = (10 + 30×0.5)/1000 = 2.5%

- Using validation set:
  - Split data into training, validation, test
  - Use validation dataset to estimate generalization error
  - Drawback: less data for training.

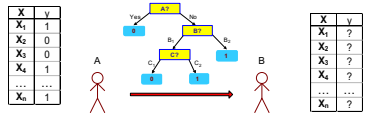
Occam's Razor

GIVEN TWO MODELS OF SIMILAR GENERALIZATION ERRORS, ONE SHOULD PREFER THE SIMPLER MODEL OVER THE MORE COMPLEX MODEL

FOR COMPLEX MODELS, THERE IS A GREATER CHANCE THAT IT WAS FITTED ACCIDENTALLY BY ERRORS IN DATA

THEREFORE, ONE SHOULD INCLUDE MODEL COMPLEXITY WHEN EVALUATING A MODEL

Minimum Description Length (MDL)



- Cost(Model,Data) = Cost(Data|Model) + Cost(Model)
  - Search for the least costly model.
- Cost(Data|Model) encodes the misclassification errors.
- Cost(Model) encodes the decision tree
  - node encoding (number of children) plus splitting condition encoding.

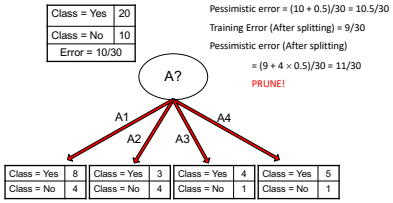
How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features (e.g., using χ^2 test)
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

How to Address Overfitting…

- Post-pruning
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If generalization error improves after trimming, replace sub-tree by a leaf node.
  - Class label of leaf node is determined from majority class of instances in the sub-tree
  - Can use MDL for post-pruning

Example of Post-Pruning



Training Error (Before splitting) = 10/30

Pessimistic error = (10 + 0.5)/30 = 10.5/30

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

= (9 + 4 × 0.5)/30 = 11/30

PRUNE!

Model Evaluation

Metrics for Performance Evaluation	How to evaluate the performance of a model?
Methods for Performance Evaluation	How to obtain reliable estimates?
Methods for Model Comparison	How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Metrics for Performance Evaluation…

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	Class=Yes	Class=No
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of classifying class j example as class i

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	+	-
	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%
Cost = 3910

Model M_2	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%
Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

Accuracy is proportional to cost if
1. C(Yes|No)=C(No|Yes) = q
2. C(Yes|Yes)=C(No|No) = p

N = a + b + c + d

Accuracy = (a + d)/N

Cost	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	p	q
	Class=No	q	p

Cost = p (a + d) + q (b + c)
= p (a + d) + q (N – a – d)
= q N – (q – p)(a + d)
= N [q – (q-p) × Accuracy]

Precision-Recall

Count	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

$$\text{Precision (p)} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

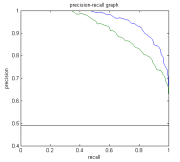
$$\text{Recall (r)} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

$$\text{F-measure (F)} = \frac{1}{\left(\frac{1/r+1/p}{2}\right)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c} = \frac{2TP}{2TP+FP+FN}$$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

Precision-Recall plot

Usually for parameterized models, it controls the precision/recall tradeoff



Model Evaluation

Metrics for Performance Evaluation	• How to evaluate the performance of a model?
Methods for Performance Evaluation	• How to obtain reliable estimates?
Methods for Model Comparison	• How to compare the relative performance among competing models?

Methods for Performance Evaluation

How to obtain a reliable estimate of performance?

Performance of a model may depend on other factors besides the learning algorithm:

Class distribution
Cost of misclassification
Size of training and test sets

Methods of Estimation

- Holdout
  - Reserve **2/3** for training and **1/3** for testing
- Random subsampling
  - One sample may be biased -- Repeated holdout
- Cross validation
  - Partition data into **k** disjoint subsets
  - **k**-fold: train on **k-1** partitions, test on the remaining one
  - Leave-one-out: **k=n**
  - Guarantees that each record is used the same number of times for training and testing
- Bootstrap
  - Sampling with replacement
  - ~63% of records used for training, ~27% for testing

Dealing with class imbalance

If the class we are interested in is very rare, then the classifier will ignore it.

The class imbalance problem

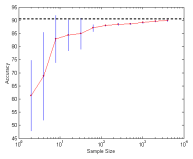
Solution:

We can modify the optimization criterion by using a cost sensitive metric

We can balance the class distribution:

- Sample from the larger class so that the size of the two classes is the same
- Replicate the data of the class of interest so that the classes are balanced -> over-fitting issues

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve
- Effect of small sample size:
  - Bias in the estimate
  - Variance of estimate

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- **ROC** curve plots TPR (on the y-axis) against FPR (on the x-axis)

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

Fraction of positive instances predicted correctly

$$FPR = \frac{FP}{FP+TN}$$

Fraction of negative instances predicted incorrectly

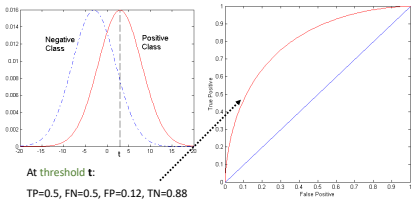
	PREDICTED CLASS		
Actual		Yes	No
	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

ROC (Receiver Operating Characteristic)

- Performance of a classifier represented as a point on the **ROC** curve
- Changing some parameter of the algorithm, sample distribution or cost matrix changes the location of the point

ROC Curve

- 1-dimensional data set containing 2 classes (***positive*** and ***negative***)
- any points located at x > t is classified as ***positive***



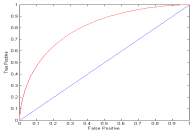
At threshold **t**:

TP=0.5, FN=0.5, FP=0.12, TN=0.88

ROC Curve

(TP,FP):

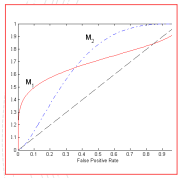
- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal



- Diagonal line:
  - Random guessing
- Below diagonal line
  - Prediction is opposite of the true class

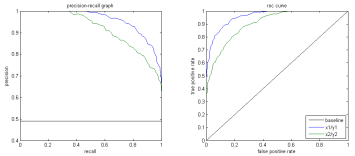
		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Using ROC for Model Comparison



- No model consistently outperform the other
  - M1 is better for small FPR
  - M2 is better for large FPR
- Area Under the ROC curve (AUC)
  - Ideal: Area = 1
  - Random guess: Area = 0.5

ROC curve vs Precision-Recall curve



Area Under the Curve (AUC) as a single number for evaluation

CIS 530—Advanced Data Mining

7- Similarity and Distance

Thomas W Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy to Prof. Panayiotis Tsaparas

Similarity and Distance

- For many different problems we need to quantify how close two objects are.
- Examples:
  - For an item bought by a customer, find other similar items
  - Group together the customers of site so that similar customers are shown the same ad.
  - Group together web documents so that you can separate the ones that talk about politics and the ones that talk about sports.
  - Find all the near-duplicate mirrored web documents.
  - Find credit card transactions that are very different from previous transactions.
- To solve these problems, we need a definition of similarity, or distance.
  - The definition depends on the type of data that we have

Similarity

- Numerical measure of how alike two data objects are.
  - A function that maps pairs of objects to real values
  - Higher when objects are more alike.
- Often falls in the range [0,1], sometimes in [-1,1]

- Desirable properties for similarity
  1. s(p, q) = 1 (or maximum similarity) only if p = q. (Identity)
  2. s(p, q) = s(q, p) for all p and q. (Symmetry)

Similarity between sets

- Consider the following documents

apple releases new ipod | apple releases new ipad | new apple pie recipe

- Which ones are more similar?

- How would you quantify their similarity?

Similarity: Intersection

- Number of words in common

apple releases new ipod | apple releases new ipad | new apple pie recipe

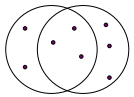
- Sim(D,D) = 3, Sim(D,D) = Sim(D,D) =2
- What about this document?

Vefa releases new book with apple pie recipes

- Sim(D,D) = Sim(D,D) = 3

Jaccard Similarity

- The Jaccard similarity (Jaccard coefficient) of two sets S_1, S_2 is the size of their intersection divided by the size of their union.
  - JSim $(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.



3 in intersection.
8 in union.
Jaccard similarity = 3/8

- Extreme behavior:
  - JSim(X,Y) = 1, iff X = Y
  - JSim(X,Y) = 0 iff X,Y have not elements in common
- JSim is symmetric

Similarity: Intersection

- Number of words in common

apple releases new ipod | apple releases new ipad | new apple pie recipe | Vefa releases new book with apple pie recipes

- JSim(D,D) = 3/5
- JSim(D,D) = JSim(D,D) = 2/6
- JSim(D,D) = JSim(D,D) = 3/9

Similarity between vectors

Documents (and sets in general) can also be represented as vectors

document	Apple	Microsoft	Obama	Election
D1	10	20	0	0
D2	30	60	0	0
D2	0	0	10	20

How do we measure the similarity of two vectors?

How well are the two vectors aligned?

Example

document	Apple	Microsoft	Obama	Election
D1	1/3	2/3	0	0
D2	1/3	2/3	0	0
D2	0	0	1/3	2/3

Documents D1, D2 are in the "same direction"

Document D3 is orthogonal to these two

Cosine Similarity

Figure 2.16. Geometric illustration of the cosine measure.

- Sim(X,Y) = cos(X,Y): The cosine of the angle between X and Y
- If the vectors are aligned (correlated) angle is zero degrees and cos(X,Y)=1
- If the vectors are orthogonal (no common coordinates) angle is 90 degrees and cos(X,Y) = 0
- Cosine is commonly used for comparing documents, where we assume that the vectors are normalized by the document length.

Cosine Similarity - math

- If d_1 and d_2 are two vectors, then
  $\cos(d_1, d_2) = (d_1 \bullet d_2) / \|d_1\| \, \|d_2\|$,
  where $\bullet$ indicates vector dot product and $\| d \|$ is the length of vector d.
- Example:

d_1 = 3 2 0 5 0 0 0 2 0 0
d_2 = 1 0 0 0 0 0 0 1 0 2

$d_1 \bullet d_2$ = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$\|d_1\|$ = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42) $^{0.5}$ = 6.481

$\|d_2\|$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)$^{0.5}$ = (6) $^{0.5}$ = 2.245

$\cos(d_1, d_2)$ = .3150

Similarity between vectors

document	Apple	Microsoft	Obama	Election
D1	10	20	0	0
D2	30	60	0	0
D2	0	0	10	20

cos(D1,D2) = 1
cos(D1,D3) = cos(D2,D3) = 0

Distance

- Numerical measure of how different two data objects are
  - A function that maps pairs of objects to real values
  - Lower when objects are more alike
- Minimum distance is 0, when comparing an object with itself.
- Upper limit varies

Distance Metric

- A distance function d is a distance metric if it is a function from pairs of objects to real numbers such that:
  1. d(x,y) $\geq$ 0. (non-negativity)
  2. d(x,y) = 0 iff x = y. (identity)
  3. d(x,y) = d(y,x). (symmetry)
  4. d(x,y) $\leq$ d(x,z) + d(z,y) (triangle inequality).

Distances for real vectors

- Vectors $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$

- L_p norms or Minkowski distance:
  $L_p(x,y) = [|x_1 - y_1|^p + \cdots + |x_d - y_d|^p]^{1/p}$

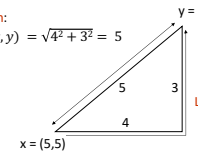
- L_2 norm: Euclidean distance:
  $L_2(x,y) = \sqrt{|x_1 - y_1|^2 + \cdots + |x_d - y_d|^2}$

- L_1 norm: Manhattan distance:
  $L_1(x,y) = |x_1 - y_1| + \cdots + |x_d - y_d|$

- L_∞ norm:
  $L_\infty(x,y) = \max\{|x_1 - y_1|, \ldots, |x_d - y_d|\}$
  - The limit of L_p as p goes to infinity.

L_p norms are known to be distance metrics

Example of Distances

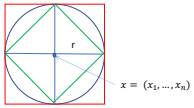


L_2-norm:
$dist(x,y) = \sqrt{4^2 + 3^2} = 5$

L_1-norm:
$dist(x,y) = 4 + 3 = 7$

L_∞-norm:
$dist(x,y) = \max\{3,4\} = 4$

Example



Green: All points y at distance $L_1(x,y) = r$ from point x

Blue: All points y at distance $L_2(x,y) = r$ from point x

Red: All points y at distance $L_\infty(x,y) = r$ from point x

L_p distances for sets

- We can apply all the L_p distances to the cases of sets of attributes, with or without counts, if we represent the sets as vectors
  - E.g., a transaction is a 0/1 vector
  - E.g., a document is a vector of counts.

Similarities into distances

- Jaccard distance:

  JDist(x,y) = 1 – JSim(x,y)

- Jaccard Distance is a metric
- Cosine distance:

  Dist(x,y) = 1−cos(x,y)

- Cosine distance is a metric

Why Jaccard Distance is a Distance Metric?

- JDist(x,x) = 0
  - since JSim(x,x) = 1
- JDist(x,y) = JDist(y,x)
  - by symmetry of intersection
- JDist(x,y) $\geq$ 0
  - since intersection of x,y cannot be bigger than the union.
- Triangle inequality:
  - Follows from the fact that JSim(x,y) is the probability of randomly selected element from the union of x and y to belong to the intersection

Hamming Distance

- Hamming distance is the number of positions in which bit-vectors differ.
- Example
  - p_1 = 10101, p_2 = 10011, $d(p_1, p_2)$ = 2 because the bit-vectors differ in the 3rd and 4th positions, the L_1 norm for the binary vectors
- Hamming distance between two vectors of categorical attributes is the number of positions in which they differ.
- Example:
  - x = (married, low income, cheat), y = (single, low income, not cheat), d(x,y) = 2

Why Hamming Distance is a Distance Metric?

- d(x,x) = 0 since no positions differ.
- d(x,y) = d(y,x) by symmetry of "different from."
- d(x,y) $\geq$ 0 since strings cannot differ in a negative number of positions.
- Triangle inequality: changing x to z and then to y is one way to change x to y.
- For binary vectors if follows from the fact that L_1 norm is a metric

Distance between strings

- How do we define similarity between strings?

weird	wierd
intelligent	unintelligent
Athena	Athina

- Important for recognizing and correcting typing errors and analyzing DNA sequences.

Edit Distance for strings

- The edit distance of two strings is the number of inserts and deletes of characters needed to turn one into the other.
- Example: x = abcde ; y = bcduve.
  - Turn x into y by deleting a, then inserting u and v after d.
  - Edit distance = 3.
- Minimum number of operations can be computed using dynamic programming
- Common distance measure for comparing DNA sequences

Why Edit Distance is a Distance Metric?

- d(x,x) = 0 because 0 edits suffice.
- d(x,y) = d(y,x) because insert/delete are inverses of each other.
- d(x,y) $\geq$ 0: no notion of negative edits.
- Triangle inequality: changing *x* to *z* and then to *y* is one way to change *x* to *y*. The minimum is no more than that

Variant Edit Distances

- Allow insert, delete, and mutate.
  - Change one character into another.
- Minimum number of inserts, deletes, and mutates also forms a distance measure.

- Same for any set of operations on strings.
  - Example: substring reversal or block transposition OK for DNA sequences
  - Example: character transposition is used for spelling

Distances between distributions

- We can view a document as a distribution over the words

document	Apple	Microsoft	Obama	Election
D1	0.35	0.5	0.1	0.05
D2	0.4	0.4	0.1	0.1
D2	0.05	0.05	0.6	0.3

- KL-divergence (Kullback-Leibler) for distributions P,Q

$$D_{KL}(P\|Q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

- KL-divergence is asymmetric. We can make it symmetric by taking the average of both sides
- JS-divergence (Jensen-Shannon)

$$JS(P,Q) = \frac{1}{2} D_{KL}(P\|Q) + \frac{1}{2} D_{KL}(Q\|P)$$

CIS 530—Advanced Data Mining

8- Min Hashing and Locality Sensitive Hashing

Thomas W Gyeera, Assistant Professor
Computer and Information Science
University of Massachusetts Dartmouth

Courtesy to Prof. Panayiotis Tsaparas

Why is similarity important?

- We saw many definitions of similarity and distance
- How do we make use of similarity in practice?
- What issues do we have to deal with?

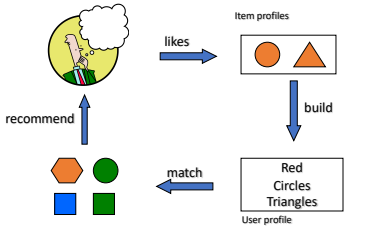
An important problem

- Recommendation systems
  - When a user buys an item (initially books) we want to recommend other items that the user may like
  - When a user rates a movie, we want to recommend movies that the user may like
  - When a user likes a song, we want to recommend other songs that they may like
- A big success of data mining
- Exploits the long tail

Recommendation systems

- Content-based:
  - Represent the items into a feature space and recommend items to customer C similar to previous items rated highly by C
  - Movie recommendations: recommend movies with same actor(s), director, genre, …
  - Websites, blogs, news: recommend other sites with "similar" content

Plan of action



Limitations of content-based approach

- Finding the appropriate features
  - e.g., images, movies, music
- Overspecialization
  - Never recommends items outside user's content profile
  - People might have multiple interests
- Recommendations for new users
  - How to build a profile?

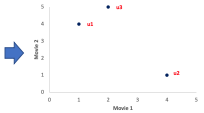
Recommendation Systems (II)

- Collaborative Filtering (user-user)
  - Consider user c
  - Find set D of other users whose ratings are "similar" to c's ratings
  - Estimate user's ratings based on ratings of users in D

Recommendation Systems (II)

- Collaborative Filtering (user-user)

	m1	m2
u1	1	4
u2	4	1
u3	2	5



Recommendation Systems (III)

- Collaborative Filtering (item-item)
  - For item s, find other similar items
  - Estimate rating for item based on ratings for similar items
  - Can use same similarity metrics and prediction functions as in user-user model
- In practice, it has been observed that item-item often works better than user-user

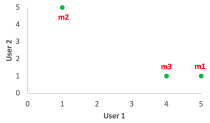
Recommendation Systems (III)

- Collaborative Filtering (item-item)

	u1	u2
m1	5	1
m2	1	5
m3	4	1

$$\cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|}$$

	m1	m2	m3
m1	1.000000	0.384615	0.998868
m2	0.384615	1.000000	0.428086
m3	0.998868	0.428086	1.000000



Pros and cons of collaborative filtering

- Works for any kind of item
  - No feature selection needed
- New user problem
- New item problem
- Sparsity of rating matrix
  - Cluster-based smoothing?

Another important problem

- Find duplicate and near-duplicate documents from a web crawl.
- Why is it important:
  - Identify mirrored web pages, and avoid indexing them, or serving them multiple times
  - Find replicated news stories and cluster them under a single story.
  - Identify plagiarism
- What if we wanted exact duplicates?

Finding similar items

- Both the problems we described have a common component
  - We need a quick way to find highly similar items to a query item
  - OR, we need a method for finding all pairs of items that are highly similar.
- Also known as the Nearest Neighbor problem, or the All Nearest Neighbors problem
- We will examine it for the case of near-duplicate web documents.

Main issues

- What is the right representation of the document when we check for similarity?
  - E.g., representing a document as a set of characters will not do (why?)
- When we have billions of documents, keeping the full text in memory is not an option.
  - We need to find a shorter representation
- How do we do pairwise comparisons of billions of documents?
  - If exact match was the issue, it would be ok, can we replicate this idea?

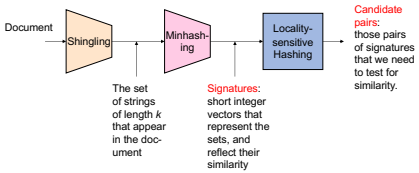
Three Essential Techniques

1. Shingling: convert documents, emails, etc., to sets.
2. Minhashing: convert large sets to short signatures, while preserving similarity.
3. Locality-Sensitive Hashing (LSH): focus on pairs of signatures likely to be similar.

Motivating problem

- Find duplicate and near-duplicate documents from a web crawl.
- If we wanted exact duplicates, we could do this by hashing
  - We will see how to adapt this technique for near duplicate documents

The Big Picture

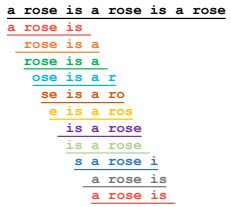


Shingles

- A k-shingle (or k-gram) for a document is a sequence of k characters that appears in the document.
- Example: document = abcab. k=2
  - Set of 2-shingles = {ab, bc, ca}.
  - Option: regard shingles as a bag, and count ab twice.
- Represent a document by its set of k-shingles.

Shingling

- Shingle: a sequence of k contiguous characters



Working Assumption

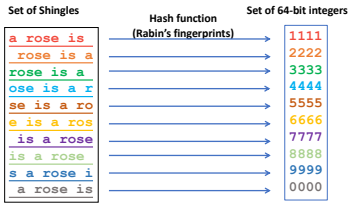
- Documents that have lots of shingles in common have similar text, even if the text appears in different order.
- Careful: you must pick *k* large enough, or most documents will have most shingles.
  - Extreme case *k = 1*: all documents are the same
  - *k* = 5 is OK for short documents; *k* = 10 is better for long documents.
- Alternative ways to define shingles:
  - Use words instead of characters
  - Anchor on stop words (to avoid templates)

Shingles: Compression Option

- To compress long shingles, we can hash them to (say) 4 bytes.
- Represent a doc by the set of hash values of its *k*-shingles.
- From now on we will assume that shingles are integers
  - Collisions are possible, but very rare

Fingerprinting

- Hash shingles to 64-bit integers



Basic Data Model: Sets

- Document: A document is represented as a set shingles (more accurately, hashes of shingles)
- Document similarity: Jaccard similarity of the sets of shingles.
  - Common shingles over the union of shingles
  - *Sim* $(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.
- Although we use the documents as our driving example the techniques we will describe apply to any kind of sets.
  - E.g., similar customers or items.

Signatures

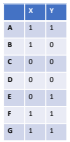
- Problem: shingle sets are too large to be kept in memory.
- Key idea: "hash" each set S to a small signature Sig (S), such that:
  1. Sig (S) is small enough that we can fit a signature in main memory for each set.
  2. Sim (S_1, S_2) is (almost) the same as the "similarity" of Sig (S_1) and Sig (S_2). (signature preserves similarity).
- Warning: This method can produce false negatives, and false positives (if an additional check is not made).
  - False negatives: Similar items deemed as non-similar
  - False positives: Non-similar items deemed as similar

From Sets to Boolean Matrices

- Represent the data as a boolean matrix M
  - Rows = the universe of all possible set elements
    - In our case, shingle fingerprints take values in $[0 \ldots 2^{64}\text{-}1]$
  - Columns = the sets
    - In our case, documents, sets of shingle fingerprints
  - M(r,S) = 1 in row r and column S, if and only if r is a member of S.
- Typical matrix is sparse.
  - We do not really materialize the matrix

Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}
- Sim(X,Y) = $\frac{3}{5}$



Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}
- Sim(X,Y) = $\frac{3}{5}$

At least one of the columns has value 1

Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}
- Sim(X,Y) = $\frac{3}{5}$

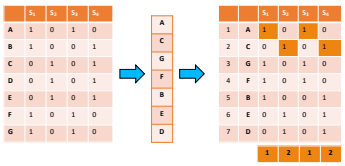
Both columns have value 1

Minhashing

- Pick a random permutation of the rows (the universe U).
- Define "hash" function for set S
  - h(S) = the index of the first row (in the permuted order) in which column S has 1.
  - OR
  - h(S) = the index of the first element of S in the permuted order.
- Use k (e.g., k = 100) independent random permutations to create a signature.

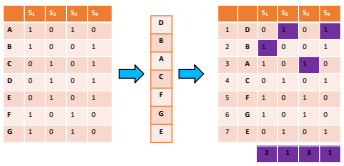
Example of minhash signatures

- Input matrix



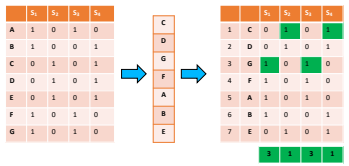
Example of minhash signatures

- Input matrix



Example of minhash signatures

- Input matrix



Example of minhash signatures

- Input matrix

	S_1	S_2	S_3	S_4
A	1	0	1	0
B	1	0	0	1
C	0	1	0	1
D	0	1	0	1
E	0	1	0	1
F	1	0	1	0
G	1	0	1	0

≈

Signature matrix

	S_1	S_2	S_3	S_4
h_1	1	2	1	2
h_2	2	1	3	1
h_3	3	1	1	1

- Sig(S) = vector of hash values
  - e.g., $Sig(S_2) = [2,1,1]$
- Sig(S,i) = value of the i-th hash function for set S
  - E.g., $Sig(S_2,3) = 1$

Hash function Property

$$Pr(h(S_1) = h(S_2)) = Sim(S_1,S_2)$$

- where the probability is over all choices of permutations.

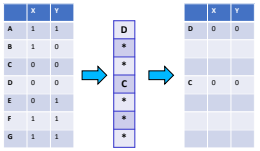
- Why?
  - The first row where one of the two sets has value 1 belongs to the union.
    - Recall that union contains rows with at least one 1.
  - We have equality if both sets have value 1, and this row belongs to the intersection

Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}

Rows C,D could be anywhere they do not affect the probability

- Union = {A,B,E,F,G}
- Intersection = {A,F,G}

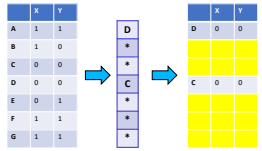


Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}

The * rows belong to the union

- Union = {A,B,E,F,G}
- Intersection = {A,F,G}

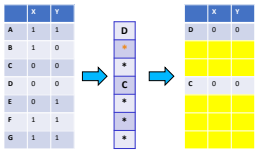


Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}

The question is what is the value of the first * element

- Union = {A,B,E,F,G}
- Intersection = {A,F,G}

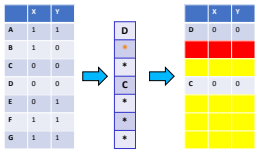


Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}

If it belongs to the intersection, then h(X) = h(Y)

- Union = {A,B,E,F,G}
- Intersection = {A,F,G}



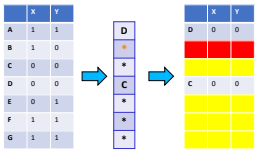
Example

- Universe: U = {A,B,C,D,E,F,G}
- X = {A,B,F,G}
- Y = {A,E,F,G}

Every element of the union is equally likely to be the * element

$$Pr(h(X) = h(Y)) = \frac{|\{A,F,G\}|}{|\{A,B,E,F,G\}|} = \frac{3}{5} = Sim(X,Y)$$

- Union = {A,B,E,F,G}
- Intersection = {A,F,G}



Similarity for Signatures

- The similarity of signatures is the fraction of the hash functions in which they agree.

	S_1	S_2	S_3	S_4
A	1	0	1	0
B	1	0	0	1
C	0	1	0	1
D	0	1	0	1
E	0	1	0	1
F	1	0	1	0
G	1	0	1	0

≈

Signature matrix

S_1	S_2	S_3	S_4
1	2	1	2
2	1	3	1
3	1	3	1

	Actual	Sig
(S_1, S_2)	0	0
(S_1, S_3)	3/5	2/3
(S_1, S_4)	1/7	0
(S_2, S_3)	0	0
(S_2, S_4)	3/4	1
(S_3, S_4)	0	0

Zero similarity is preserved
High similarity is well approximated

- With multiple signatures we get a good approximation

Is it now feasible?

- Assume a billion rows
- Hard to pick a random permutation of 1 billion…
- **Even representing a random permutation requires 1 billion entries!!!**
- How about accessing rows in permuted order? ☹

Being more practical

- Instead of permuting the rows we will apply a hash function that maps the rows to a new (possibly larger) space
  - The value of the hash function is the position of the row in the new order (permutation).
  - Each set is represented by the smallest hash value among the elements in the set
- The space of the hash functions should be such that if we select one at random each element (row) has equal probability to have the smallest value
  - Min-wise independent hash functions

Algorithm – One set, one hash function

Computing **Sig(S,i)** for a single column S and single hash function h_i

In practice only the rows (shingles) that appear in the data

for each row **r**
  compute $h_i(r)$
  if column **S** that has **1** in row **r**
  if $h_i(r)$ is a smaller value than **Sig(S,i)** **then**
    Sig(S,i) = $h_i(r)$;

$h_i(r)$ = index of row r in permutation

S contains row r

Find the row r with minimum index

Sig(S,i) will become the smallest value of $h_i(r)$ among all rows (shingles) for which column **S** has value **1** (shingle belongs in S); *i.e.*, $h_i(r)$ gives the min index for the **i**-th permutation

Algorithm – All sets, k hash functions

Pick k=100 hash functions $(h_1,…,h_k)$

In practice this means selecting the hash function parameters

for each row **r**
  for each hash function h_i
    compute $h_i(r)$
  for each column **S** that has **1** in row **r**
    if $h_i(r)$ is a smaller value than **Sig(S,i)** **then**
      Sig(S,i) = $h_i(r)$;

Compute $h_i(r)$ only once for all sets

Algorithm – All sets, k hash functions

x	Row	S1	S2	h(x)	g(x)
0	A	1	0	1	3
1	B	0	1	2	0
2	C	1	1	3	2
3	D	1	0	4	4
4	E	0	1	0	1

$h(x) = x+1 \bmod 5$
$g(x) = 2x+3 \bmod 5$

h(Row)	Row	S1	S2
0	E	0	1
1	A	1	0
2	B	0	1
3	C	1	1
4	D	1	0

g(Row)	Row	S1	S2
0	B	0	1
1	E	0	1
2	C	1	0
3	A	1	1
4	D	1	0

	Sig1	Sig2
$h(0) = 1$ $g(0) = 3$	1 3	- -
$h(1) = 2$ $g(1) = 0$	1 3	2 0
$h(2) = 3$ $g(2) = 2$	1 2	2 0
$h(3) = 4$ $g(3) = 4$	1 2	2 0
$h(4) = 0$ $g(4) = 1$	1 2	0 0

Implementation

- Often, data is given by column, not row.
  - E.g., columns = documents, rows = shingles.
- If so, sort matrix once so it is by row.
- And always compute $h_i(r)$ only once for each row.

Finding similar pairs

- Problem: Find all pairs of documents with similarity at least t = 0.8
- While the signatures of all columns may fit in main memory, comparing the signatures of all pairs of columns is quadratic in the number of columns.
- Example: 10^6 columns implies $5*10^{11}$ column-comparisons.
- At 1 microsecond/comparison: 6 days.

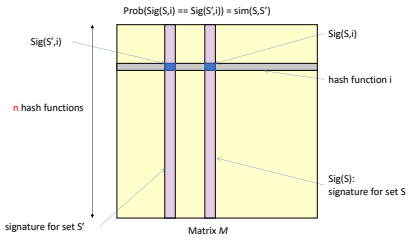
Locality-Sensitive Hashing

- What we want: a function f(X,Y) that tells whether or not X and Y is a candidate pair: a pair of elements whose similarity must be evaluated.
- A simple idea: X and Y are a candidate pair if they have the same min-hash signature.
  - Easy to test by hashing the signatures.
  - Similar sets are more likely to have the same signature.
  - Likely to produce many false negatives.
    - Requiring full match of signature is strict, some similar sets will be lost.

! Multiple levels of Hashing!

- Improvement: Compute multiple signatures; candidate pairs should have at least one common signature.
  - Reduce the probability for false negatives.

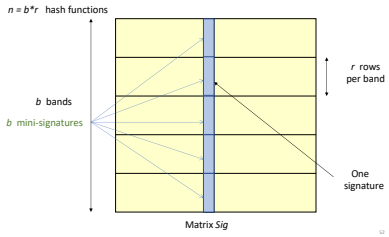
Signature matrix reminder



Partition into Bands – (1)

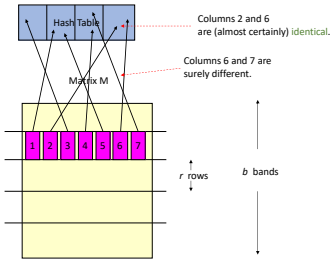
- Divide the signature matrix Sig into *b* bands of *r* rows.
  - Each band is a mini-signature with *r* hash functions.

Partition into Bands – (1)



Partition into Bands – (2)

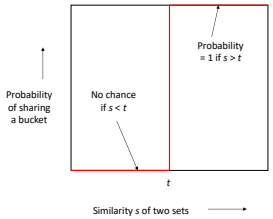
- Divide the signature matrix Sig into *b* bands of *r* rows.
  - Each band is a mini-signature with r hash functions.
- For each band, hash the mini-signature to a hash table with *k* buckets.
  - Make *k* as large as possible so that mini-signatures that hash to the same bucket are almost certainly identical.



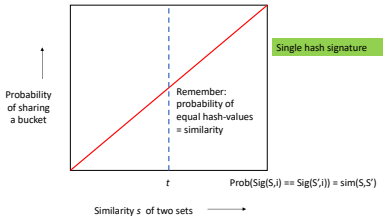
Partition into Bands – (3)

- Divide the signature matrix Sig into *b* bands of *r* rows.
  - Each band is a mini-signature with r hash functions.
- For each band, hash the mini-signature to a hash table with *k* buckets.
  - Make *k* as large as possible so that mini-signatures that hash to the same bucket are almost certainly identical.
- Candidate column pairs are those that hash to the same bucket for at least 1 band.
- Tune *b* and *r* to catch most similar pairs, but few non-similar pairs.

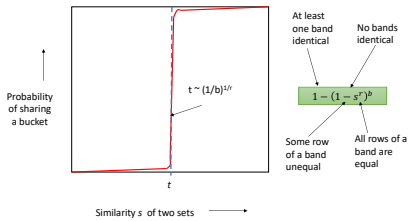
Analysis of LSH – What we want



What One Band of One Row Gives You



What *b* Bands of *r* Rows Gives You



Example: *b* = 20; *r* = 5

s	1-(1-s^r)^b
.2	.006
.3	.047
.4	.186
.5	.470
.6	.802
.7	.975
.8	.9996

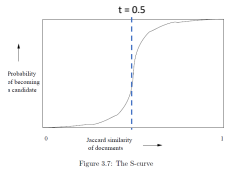


Figure 3.7: The S-curve

Suppose S_1, S_2 are 80% Similar

- We want all 80%-similar pairs. Choose 20 bands of 5 integers/band.
- Probability S_1, S_2 identical in one particular band:
  $(0.8)^5 = 0.328$.
- Probability S_1, S_2 are not similar in any of the 20 bands:
  $(1\text{-}0.328)^{20} = 0.00035$
  i.e., about 1/3000-th of the 80%-similar column pairs are false negatives.
- Probability S_1, S_2 are similar in at least one of the 20 bands:
  1-0.00035 = 0.999

Suppose S_1, S_2 Only 40% Similar

- Probability S_1, S_2 identical in any one particular band:
  $(0.4)^5 = 0.01$.
- Probability S_1, S_2 identical in at least 1 of 20 bands:
  $\leq 20 * 0.01 = 0.2$.
- But false positives much lower for similarities << 40%.