

Preparación de datos para Data Mining

Outline: Preparación de datos

- Entendiendo los datos
- Limpieza de datos
 - Meta-datos
 - Valores faltantes
 - Formatos de fechas
 - Conversiones de varios a numéricos
 - Discretización
 - Normalización
- Pre-selección de variables. “Falsos Predictores”
- Clases desbalanceadas

Entendiendo los datos : Relevancia

- Qué datos tenemos disponibles?
- Los datos son relevantes?
- Hay otros datos relevantes disponibles?
- Qué período de tiempo cubren los datos?
- Quién es el experto en esos datos?

Entendiendo los datos : Calidad

- Número de registros (instancias)
 - Menos datos, menos confiables los resultados
- Número de variables (campos)
 - *Rule of thumb (clásica): por cada variable, 10 o más registros*
 - Si hay demasiadas, se puede usar selección o extracción de variables
- Número de targets
 - Si es muy desbalanceado se puede intentar corregir

Limpieza de Datos: Adquisición

- Los datos pueden estar en DBs
 - ODBC, JDBC, protocolos diversos
- Los datos pueden estar en texto plano
 - Ancho fijo
 - Delimitados: tab, “,” , espacio, otros?
 - Ej. C4.5. Weka usa “arff”, con coma.
- Precaución: Verificar el número de campos leídos

Limpieza de Datos: Adquisición

- Ejemplo en R:

```
>datos<-read.csv("datos.txt")
```

```
>summary(datos)
```

```
#veo con editor y corrijo
```

```
>datos<-read.table("datos.txt",sep="\t",head=T)
```

```
>summary(datos)
```

```
#el año debería ser factor? pasarlo!
```

Limpieza de Datos: Meta-datos

- **Información relevante (datos) sobre los datos**
- **Tipo de variable:**
 - binaria, nominal (categórica), ordinal, numérica, ...
 - Para nominales: tablas de translación
- **Uso de la variable:**
 - input : entradas de los modelos
 - output
 - id/auxiliar: Se leen, pero no se usan al modelar
 - Para ignorar.
 - weight : variables que dan peso estadístico

Limpieza de Datos: Formato

En general es necesario convertir los datos a un formato estándar con campos numéricos (ej. arff o csv)

- Puntos a ver:
 - Valores faltantes
 - Formato de fechas
 - Discretización de datos numéricos
 - Limpieza de errores y outliers

Limpieza de Datos: Valores Faltantes

- Los datos faltantes se marcan de distintas formas:
 - <empty field> “0” “.” “999” “NA” ...
- Unificar el código.
- Qué se hace con los valores faltantes?

Limpieza de Datos: Valores Faltantes, 2

■ Alternativas

- Ignorar los *registros* con datos faltantes
- Ignorar las *variables* con datos faltantes
- Tratar NA como un valor particular (qué valor?)
- Imputation (llenado):
 - Llenar con medias o medianas (medias por clases?)
 - Predecir el NA con un método de ML.

Limpieza de Datos: Valores Faltantes, Ejemplo

- Llenado con la media

```
>datos[sample(100,10),2]<-(-1) #genera  
faltas
```

```
>datos[datos[,2]==-1,]
```

```
>media<-mean(datos[,2]) (bien? Que pasa con  
los -1?)
```

```
>datos[datos[,2]==-1,2]<-media
```

Limpieza de Datos: Formato de fechas

- Necesitamos que las fechas estén en un formato uniforme y coherente
- Un formato util es YYYYMM o YYYYMMDD
 - O hasta YYYYMMDDHHMMSS
- Problema con las fechas YYYYMMDD:
 - YYYYMMDD no conserva distancias
 - 20090201 - 20090131 != 20090131 - 20090130

Limpieza de Datos: Formato de fechas, 2

- Para preservar intervalos:
 - Unix system date: Número de segundos desde 1970
 - Número de días desde 1/1/1960 (SAS)
- Problemas:
 - valores no entendibles, propensos a errores

Limpieza de Datos: Formato de fechas, 4: KSP

$$\text{KSP Date} = \text{YYYY} + \frac{\text{days_starting_Jan_1} - 0.5}{365 + 1_if_leap_year}$$

- Conserva intervalos razonablemente
- Valores obvios: 30 junio, 30 setiembre, etc
- Se puede agregar la hora al mismo formato

Conversión: Nominales a Numéricos

- Muy pocos modelos numéricos pueden trabajar con datos nominales directamente
- Muchos (neural nets, regresión lineal, K-NN) aceptan sólo datos numéricos
- En esos casos es necesario convertir los datos nominales a numéricos
- Se usan diferentes estrategias para datos binarios, ordinales y nominales con muchos valores distintos

Conversión: Binarios a Numéricos

- Binarias

- Ej. Género=M, F

- Convertir a 0, 1 (o -1,1)

- ej. Gender = M \rightarrow Gender_0_1 = 0

- Gender = F \rightarrow Gender_0_1 = 1

- En R: `Gender_0_1[Gender=="M"]<-0`

Conversión: Ordinales a Numéricos

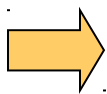
- Variables ordinales (ej. Calidades, calificaciones con letras) se convierten a números preservando el orden natural (y la escala si se la conoce)
 - A → 4.0
 - A- → 3.7
 - B+ → 3.3
 - B → 3.0

Para que siga teniendo sentido el “<” y “>”

Conversión: Nominal, pocos valores

- Atributos nominales, sin orden, con un número chico de valores posibles (*rule of thumb* < 20)
- ej. Color=Rojo, Azul, Amarillo, ..., Verde
 - Se convierte cada valor a una variables binaria, que es 1 si el atributo toma ese valor y 0 en todos los otros casos (uno-de-c)

ID	Color	...
371	rojo	
433	azul	



ID	C_rojo	C_verde	C_azul	...
371	1	0	0	
433	0	0	1	

Conversión: Nominal, muchos valores

- Ejemplo:
 - US State Code (50 valores)
 - “Código de Profesión” (~7,000 valores posibles, pero solo pocos muy frecuentes o de interés)
- Usualmente se ignoran valores únicos
- Se pueden usar grupos (regiones en lugar de estados)
- Dejar como categoría los más frecuentes, agrupar los otros en un grupo “otros”
- Crear variables binarias para las categorías seleccionadas

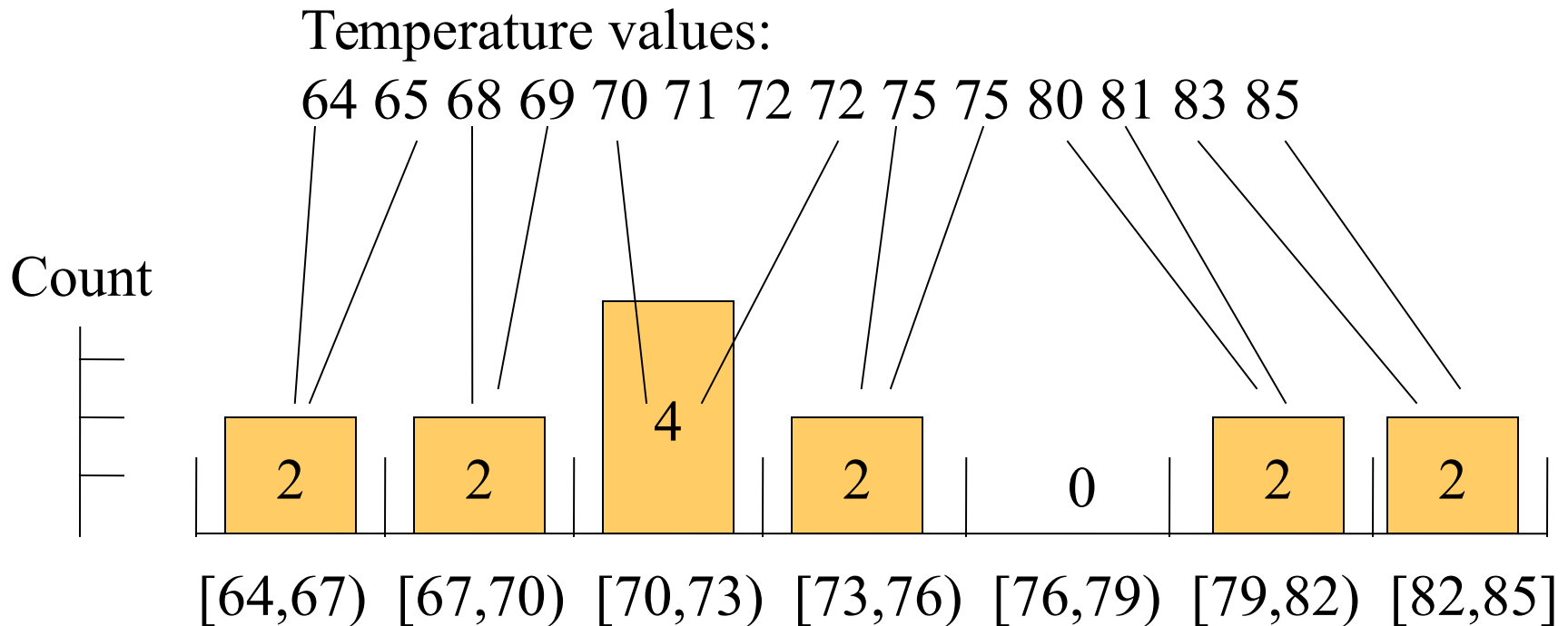
Conversión: Variables cíclicas

- Ejemplos:
 - Horas (igual distancia de 23-24 que de 24-1)
 - Rumbos (350-360-10 grados)
- Q: Que se puede hacer en estos casos?
- A: Usar 2 variables, tipo coordenadas x-y sobre un círculo
 - `>hora_x<-cos(2*PI*hora/12)`

Limpieza: Discretización

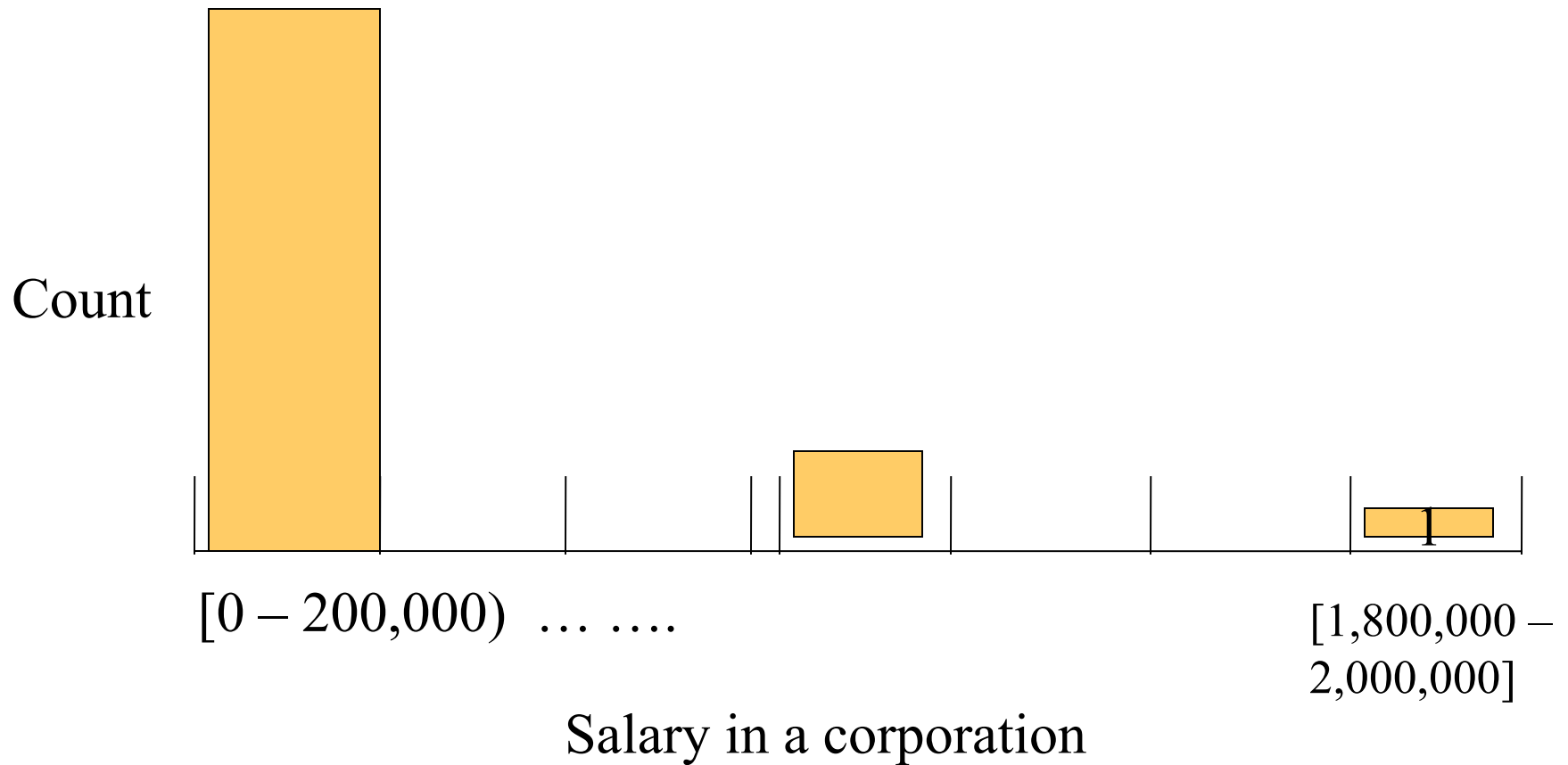
- Algunos métodos requieren valores discretos (Naïve Bayes, CHAID)
- La discretización es útil para generar resúmenes de datos
- También conocido como “binning”

Discretization: Equal-width



Equal Width, bins $\text{Low} \leq \text{value} < \text{High}$

Discretization: Equal-width may produce clumping



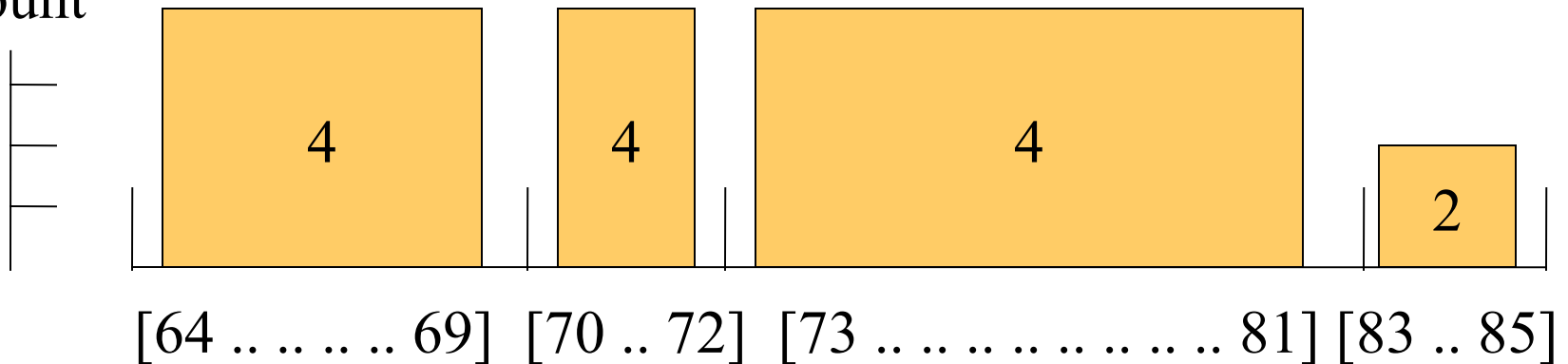
What can we do to get a more even distribution?

Discretization: Equal-height

Temperature values:

64 65 68 69 70 71 72 72 75 75 80 81 83 85

Count



Equal Height = 4, except for the last bin

Discretización: Ventajas de Equal-height

- Generalmente preferible porque evita los grupos, es uniforme.
- En la práctica se usa “almost-equal” binning. Evita los grupos y da breaks simples
- Otras consideraciones en bins:
 - Suele ser útil crear bins separados para valores especiales (ej. 0)
 - Usar breakpoints simples (ej. Valores redondos)

Discretización: consideraciones

- Anchos iguales es simple, y muchas veces funciona
 - En distribuciones “raras” puede funcionar muy mal
- Alturas iguales suele ser mejor
- Para clasificación, buscar bins con información de clases puede ser mejor todavía
 - C4.5 discretiza “de-facto”
 - Naïve Bayes con bins con MI
- Muchos muchos otros métodos...

Normalización

- Normalización min-max

$$v' = \frac{v - \min A}{\max A - \min A} (\text{new}_{\max} A - \text{new}_{\min} A) + \text{new}_{\min} A$$

- Normalización z-score

$$v' = \frac{v - \text{mean} A}{\text{stand_dev} A}$$

- Normalización por escala de décadas

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(| \quad |) < 1$$

Outliers y Errores

- Outliers son valores que, se cree, están fuera de rango.
 - Normalmente, cualquier valor fuera de 2 veces la distancia inter-cuartil, desde los cuartiles 1 y 3.
- Aproximaciones al problema:
 - No hacer nada
 - Forzar límites (inferior y superior)
 - Hacer binning (pero no de igual ancho)
 - otros...

Outliers y Errores

- Ejemplo en R

```
>a<-rnorm(1000)
```

```
>a[1]<-5
```

```
>iqr<-quantile(a)[4]-quantile(a)[2]
```

```
>out<-quantile(a)[4]+2*iqr
```

```
>a[a>out]
```

```
>boxplot(a)
```

Limpieza: pre-selección de variables útiles

Remover campos con ninguna variación (siempre) o poca variación (a veces)

- Examinar el número de valores distintos que toma una variable
 - *Rule of thumb: sacar variables que toman “casi siempre” el mismo valor*
- Sacar variables con poca variabilidad (rango) es siempre peligroso, las pequeñas diferencias pueden ser muy importantes para clasificar!

Falsos Predictores

- Los falsos predictores son campos correlacionados con el target, pero que no sirven para predecir.
- Usualmente relacionados a hechos posteriores en el tiempo a la asignación del target
- Si no hay suficientes meta-datos, un falso predictor se puede confundir con un buen predictor
- Ejemplo: La nota final de un curso predice perfectamente quien aprueba el curso

Falsos Predictores: detectar “sospechosos”

- Construir un árbol de decisión
 - Considerar como sospechosa a cualquier variable que identifique casi completamente a una clase al tope del árbol
- Usar tablas de contingencia de las variables nominales con la clase (`table()`)
- Calcular correlaciones entre predictores y salidas
- Chequear “sospechosos” usando conocimiento del campo o a un experto

Clases desbalanceadas

- En algunos casos las clases tiene muy diferentes frecuencias de aparición
 - Attrition: 97% permanecen, 3% renuncian (mensual)
 - Diagnóstico médico: 95% sanos, 5% enfermos
 - eCommerce: 99% no compran, 1% compra
 - Seguridad: >99.99% de la gente no es terrorista
- Clasificar con la clase mayoritaria da un bajo error, pero no es informativa

Clases desbalanceadas

- Estrategias simples:
 - Sobre-sampear la clase minoritaria
 - Sub-sampear la clase mayoritaria
 - Utilizar una estrategia inteligente para considerar solo las partes que sirven de la clase mayoritaria (descartar casos demasiado obvios)
 - Muchos otros

Resumen

Si entra basura, sale
basura

Preparar los datos adecuadamente
es fundamental para obtener
buenos resultados.