Minería de Datos

Tema 2 - Reducción de Dimensionalidad

Nombre: Alejandro pérez Belando

---

1. Introducción

La complejidad de la mayoría de algoritmos de aprendizaje depende de la cantidad de variables (dimensión del conjunto de entrada) y del número de instancias. Es por eso que a medida que aumentamos la dimensionalidad, las clasificaciones se vuelven más difíciles y la precisión de las consultas disminuye.

En conjuntos más pequeños, los modelos simples son más robustos, se facilita la extracción de conocimiento y las representaciones son más sencillas.

Para reducir la dimensionalidad de un problema, podemos tener dos métodos: *extracción de características* y *selección de características*

2. Extracción de características

Se basa en encontrar una transformación desde un espacio de dimensión d a otro de dimensión k ($d > k$). Hacen uso de todas las dimensiones originales y existen dos variantes:

- **Supervisados**: buscan maximizar la discriminación entre clases.
- **No supervisados**: buscan minimizar la pérdida de información.

Algunas técnicas: Análisis Lineal Discriminante (LDA), Análisis Discriminante Generalizado (GDA), Análisis de componentes principales (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) o Uniform Manifold Approximation and Projection (UMAP).

3. Selección de características

Se basan en encontrar, de las d dimensiones originales, qué dimensiones k aportan la mayor cantidad de información de acuerdo con un determinado criterio. Solo se selecciona un subconjunto de las dimensiones originales y se trabaja con ellas sin realizar transformaciones.

Subconjunto óptimo: es el subconjunto mínimo que permite construir una hipótesis consistente. Se puede realizar sobre los datos de entrenamiento o sobre todo el conjunto de datos.

Dicho de otra forma, sea F el conjunto de todas las características y C el conjunto de clases: G es el subconjunto óptimo (de características) tal que $P(C|G)$ es lo más próxima posible a $P(C|F)$.
Donde: $P(C|F) \equiv$ probabilidad de predecir las clases a partir de las características originales y $P(C|G) \equiv$ probabilidad de predecir las clases a partir del subconjunto óptimo.

3.1. Generación de subconjuntos

Se generan distintos subconjuntos de características candidatos a ser el subconjunto óptimo. Se pueden emplear distintas estrategias de búsqueda:

- **Exhaustiva**: se barre todo el espacio de características y es la única forma de garantizar encontrar el subconjunto óptimo. Sin embargo, solo es posible para pocas características.
- **Heurística**: se usa cuando se dispone de información sobre qué subconjunto es el más prometedor. No garantizan que el subconjunto sea óptimo, aunque sí se suele encontrar una buena solución en un tiempo razonable.
- **Aleatoria**: se parte de una configuración inicial formada por un conjunto finito de posibles subconjuntos. Se va modificando la configuración inicial para dirigir la búsqueda aunque o garantiza una solución óptima.

Dirección de búsqueda: es importante definirla antes de la estrategia:

- **Hacia adelante (forward)**: se comienza con el conjunto vacío y se va añadiendo una característica no seleccionada cada vez.
- **Hacia atrás (backward)**: se comienza con todas las características y se va eliminando una característica cada vez.
- **Bidireccional**: se comienza por los dos extremos del espacio de búsqueda y se realiza de forma paralela las dos búsquedas explicadas antes.

3.2. Evaluación de subconjuntos

Los métodos evalúan cada variable de acuerdo con una determinada medida y establecen un ranking del que seleccionan las características mejor clasificadas (ya no hay fase de búsqueda de subconjuntos).

Estas técnicas son muy rápidas y fácilmente escalables, pero puede ser difícil establecer un umbral de corte.

Las **medidas para evaluar las características** se clasifican en:

- **Medidas de evaluación por pares**: evalúan la dependencia de las variables con la variable objetivo. Hacen un análisis univariable (no se puede evaluar la interacción de grupos de variables respecto a la variable objetivo). Estas medidas se agrupan en:
  - **Basadas en correlación**: miden la correlación entre cada variable y la objetivo.
  - **Basadas en incertidumbre**: se apoyan en medidas típicas de la teoría de la información (como la ganancia de información).
  - **Basadas en test de hipótesis**: para obtener los *p-values*.
  - **Basadas en el poder discriminativo**: usando un modelo de una variable y estimando su error.
- **Medidas de evaluación simultáneas**: realizan la evaluación de manera simultánea para todas las variables. Pueden hacer un análisis multivariable (permiten evaluar la interacción y discrepancia entre variables).

Algoritmo Relief

Algoritmo Relief

Entrada: Dataset **D**, Número de iteraciones T
Salida: Pesos de las características, **W**
$\mathbf{W} \leftarrow 0$
para $t = 1$ to T **hacer**
    para cada instancia $x_i \in \mathbf{D}$ **hacer**
        Encontrar el "nearest hit" , nh y el "nearest miss" nm de x_i
        para cada variable j **hacer**
            Actualizar su peso: $w_j = w_j - \frac{1}{T} \cdot \left(|x_{ij} - nh_j|^2 - |x_{ij} - nm_j|^2\right)$
        fin para
    fin para
fin para
Normalizar los pesos W
devolver W

Donde: *nearest hit* es el elemento más cercano de la misma clase y *nearest miss* el elemento más cercano de la otra clase.

Tenemos varios **tipos de métodos de evaluación de subconjuntos**: basados en *filtros*, en *envoltura (wrappers)* y *empotrados (embedded)*

3.2.1. Métodos basados en filtros

Evalúan la relevancia de las características teniendo solo en cuenta las propiedades intrínsecas de las mismas (sin tener en cuenta la tarea que van a realizar con los datos).

Son independientes de la técnica de regresión o clasificación a realizar.

Medidas de relevancia:

- **Ratio de inconsistencias**: una inconsistencia aparece cuando dos características iguales pertenecen a clases distintas.
- **Medidas basadas en distancias**: se centran en encontrar el subconjunto de características que mejor separe las diferentes clases.

Algunos **algoritmos representativos**: FOCUS (búsqueda exhaustiva con medieda de consistencia), Cluster-Based Feature Selection Approach (k-medidas de correlación en el espacio de características) y CFS (Correlation-Based Feature Selection) (búsqueda heurística con una medida basada en la correlación)

3.2.2. Métodos basados en envoltura (wrappers)

Cada subconjunto candidato se evalúa mediante la construcción de un regresor o clasificador (la medida de evaluación es su capacidad predictiva). En este caso, sí se tiene en cuenta la tarea que van a realizar las características.

Estos métodos producen mejores resultados, pero tienen un mayor riesgo de sobreajuste y son computacionalmente más costosos.

El wrapper más usado es la **eliminación recursiva de características**:

Algoritmo Eliminación recursiva de características

1: Construir un modelo con todas las características
2: Evaluar el modelo
3: Calcular la relevancia de las características
4: Crear una lista con las características ordenadas de mayor a menor relevancia.
5: **para** $size = n$ to 1 **hacer**
6: Crear un conjunto S_{size} con las $size$ características más relevantes
7: Construir un modelo utilizando las características S_{size}
8: Evaluar el modelo
9: [Opcional] Recalcular la relevancia de las características
10: **fin para**
11: Crear una lista con todos los S_i y el resultado de la evaluación
12: Determinar el subconjunto óptimo S_{opt}

Como podemos ver, este wrapper tiene dirección de búsqueda hacia atrás. También se puede ajustar para una **búsqueda hacia adelante**. El principal inconveniente de esto es el sobreajuste, que se puede evitar con un bucle externo que lleve a cabo un remuestreo:

Algoritmo Eliminación recursiva de características con remuestreo

1: **para** *Cada iteración de remuestreo* **hacer**
2: Crear los conjunto de entrenamiento E y prueba T
3: Construir un modelo sobre E con todas las características
4: Evaluar el modelo en T
5: Calcular la relevancia de las características
6: Crear una lista con las características ordenadas de mayor a menor relevancia.
7: **para** $size = n$ to 1 **hacer**
8: Crear un conjunto S_{size} con las *size* características más relevantes
9: Construir un modelo utilizando las características S_{size}
10: Evaluar el modelo
11: [Opcional] Recalcular la relevancia de las características
12: **fin para**
13: **fin para**
14: Crear una lista con todos los S_i y el resultado de la evaluación
15: Determinar el subconjunto óptimo S_{opt}
16: Crear un modelo con las variables S_{opt} y con el conjunto de entrenamiento original.

Algunos **algoritmos representativos de los wrappers** son. OBLIVION (búsqueda voraz y árboles de decisión), RFE+SVM (eliminación recursiva de características + SVM), FFE+NNets (búsqueda hacia adelante y redes neuronales) y GA+C4.5 (búsqueda aleatoria y G4.5)

3.2.3. Métodos empotrados (embedded)

El proceso de selección de características está integrado en la técnica de construcción del predictor (modelo); esto quiere decir que al mismo tiempo que se ajustan los parámetros internos, se seleccionan las características más relevantes.

Computacionalmente son menos costosos que los wrappers y tienen la ventaja sobre los filtros de que generan además el modelo. además, al igual que con los Wrappers, la selección de características está sesgada hacia una mejor eficiencia del proceso de regresión o clasificación.

Algunos ejemplos de estos métodos:

- **Técnicas basadas en árboles de decisión** (como Random Forest): permite analizar la interacción entre características.
- **Modelos con regularización** (como LASSO): normalmente se usan en modelos lineales penalizando los coeficientes de las características menos significativa.
- Estas técnicas permiten obtener un **ranking de características**: donde se tiene en cuenta la importancia de las variables en las técnicas basadas en árbol y los valores de los coeficientes de las técnicas basadas en regularización.