Generación Automática de Configuraciones Visuales

Victor Manuel Cardentey Fundora
Grupo C511

Karla Olivera Hernández
Grupo C511

Amanda González Borrell
Grupo C511

Tutor(es):
Lic. Daniel Alejandro Valdés Pérez
Lic. Ernesto Estevanell

1. Introducción

La generación automática de visualizaciones sobre un conjunto de datos se puede dividir en dos procesos: determinar una consulta de interés para el usuario y generar la configuración gráfica para visualizar los resultados de la consulta. En particular la selección de configuraciones gráficas es un problema que presenta dificultades para llegar a consenso entre expertos del dominio y los sistemas tradicionales que brindan solución a este problema utilizan enfoques basados en reglas. En años recientes se ha planteado la posibilidad de aplicar técnicas de *Machine Learning* ampliamente utilizadas en sistemas de recomendación tradicionales a la recomendación de configuraciones gráficas [**?**] [**?**]. La propuesta de este trabajo consiste en utilizar y comparar distintos modelos de *Machine Learning* en la tarea de selección de configuraciones gráficas e implementar un marco de trabajo para futuras investigaciones sobre el tema.

2. Desarrollo

2.1 Definición del problema

El proplema de elegir configuraciones gráficas se define como dado un *dataset* de k columnas elegir los valores de configuraciones gráficas para visualizar estos datos, ejemplos de configuraciones gráficas son el tipo de gráfico y el eje correspondiente a cada columna en dicho gráfico. Como convención para diferenciar el *dataset* correspondiente a un gráfico del *dataset* de *datasets* utilizado para entrenar los modelos propuestos se utilizará *Plotly Dataset* para referirse a este último durante el resto de este trabajo.

Este problema se modeló como un problema de clasificación multiclase:

Sea un conjunto $C = \{T, A\}$ de configuraciones gráficas donde:

1. $T = \mathbb{N}_k$ donde k es la cantidad de tipos de gráficos.
2. $A = \{a_1, a_2, ..., a_k\}$ donde k es la cantidad de tipos de gráficos y a_i es la cantidad de ejes válidos en un gráfico de tipo i.

Se consideraron como tipo de gráficos el gráfico de barras, gráfico de líneas y gráfico de puntos entonces $T = \mathbb{N}_3$, todos estos tipos de gráficos tienen dos ejes válidos por lo que $\forall i \in [1, k]\ A_i = \mathbb{N}_2$.

Se define el conjunto de posibles clases

$$L = \{(t_i, j) : t_i \in T\ \forall j \in [1, a_i]\ \forall i \in [1, k]\}$$

.

En este caso

$$L = \{(1,1), (1,2), (2,1), (2,2), (3,1), (3,2)\}$$

Estas categorías pueden interpretarse como:
$\{(\textbf{barra}, \textbf{eje\_x}), (\textbf{barra}, \textbf{eje\_y}), (\textbf{línea}, \textbf{eje\_x})$
$, (\textbf{línea}, \textbf{eje\_y}), (\textbf{punto}, \textbf{eje\_x}), (\textbf{punto}, \textbf{eje\_y})\}$

Entonces dado un conjunto de n columnas $V = \{v_1, v_2, ..., v_n\}$ pertenecientes al *Plotly Dataset* y un conjunto $L_V = \{l_1, l_2, ..., l_n\}$ de clases donde $\forall i \in [1, n]\, l_i \in L$ se define la función $H : V \to L_V$ como $H(v_i) = l_i\ \forall i \in [i, n]$. El objetivo de este trabajo es aproximar H mediante una función $H' \approx H$, utilizando un enfoque de aprendizaje supervisado esta función es representada mediante un modelo H' el cual posee un conjunto de parámetros $\theta_{H'}$ y puede ser entrenado en el conjunto de datos $D = (V, L_V)$, luego los parémetros del modelo son seleccionados mediante la minimización de una función objetivo f que captura el error en la predicción realizada por el modelo.

$$\theta^* = \arg\min_{\theta_{H'}} \sum_{v \in V} f(l_v, H'(D \mid \theta_{H'}, L))$$

En la sección **Modelos utilizados** se realizará una comparación de los resultados de diferentes modelos encontrados en la literatura de *Machine Learning* aplicados al presente problema.

2.2 Fabricación del *Plotly Dataset*

Plotly [**?**] es una compañía de *software* la cual desarrolla herramientas para la visualización y análisis de

datos. Esta compañía mantiene en funcionamiento la plataforma Plotly Community plotlyFeed [**?**] donde los usuarios pueden almacenar y compartir sus visualizaciones, esta plataforma provee una API REST [**?**] la cual permite la descarga de dichas visualizaciones. Estas visualizaciones son especificadas en un lenguaje declarativo lo cual permite que puedan ser parseadas para extraer sus configuraciones y datos. Este proceso fue realizado por [**?**] el cual creó un corpus con 10^6 visualizaciones, debido a limitaciones de recursos computacionales en este trabajo se seleccionó el corpus provisto por [**?**] el cual contiene 3×10^5 visualizaciones.

2.2.1 Representación de los datos

En la definición del problema los puntos de datos se corresponde a columnas de un *dataset* las cuales pueden ser de distinto tipo y longitud, esta representación no es procesable por los algoritmos de la literatura los cuales se basan en representar los datos de entrada como vectores en un espacio de dimensión fija.

Para ello se realiza el proceso de *feature extraction* el cual intenta representar los puntos de datos mediante características que permitan diferenciarlos. A continuación damos una descripción de los *features* utilizados en este trabajo.

Medidas de dimensión

1. **Longitud**: La cantidad de elementos de una columna, esta medida obtuvo un alto índice de relevancia lo cual parece respaldar ciertas heurísticas utilizadas de forma común por analistas como pueden ser "no tener demasiadas barras en los gráficos" o "no tener demasiadas porciones en un gráfico de pastel" debido a que dificultan la correcta observación de los datos.

Medidas de tipo

1. **Tipo general**: El tipo general se refiere a la clasificación de la variable estadística pudiendo ser categórica (C), cuantitativa (Q) o temporal (T), esta clasificación se apoya en heurísticas comunes como utilizar variables temporales y categóricas en el eje x.

2. **Tipo Específico**: Se refiere al tipo de dato utilizado para representar la variable pudiendo ser una cadena de texto (*string*), un valor booleano (*boolean*), un entero (*integer*), un decimal (*decimal*) o una fecha (*datetime*).

Medidas Generales

1. **Ordenación**: Esta medida representa que tan ordenados están los datos dado que una ordenación previa puedes ser un indicativo de que el usuario preparó la columna para ser considerada como variable independiente. Esta medida puede ser calculada mediante el número de inversiones necesario para ordenar los datos.

2. **Unicidad**: Determina el por ciento de valores únicos en la columna, esto puede ser util para descartar visualizaciones como el gráfico de barras.

Medidas de valores

Estas medidas se encargan de describir características de los datos de la columna y debido a que existen distintos tipos de variables estas medidas son dependientes del tipo.

1. Estadísticas [Q,T]:

   a) **Coeficiente de variación**: El coeficiente de variación expresa la razón entre la desviación típica y la media aritmética:

   $$CV = \frac{s}{\overline{x}}$$

   Tiene la ventaja de ser una medida de variabilidad relativa permitiendo poder comparar columnas con diferentes rangos de valores y unidades de medidas [**?**].

   b) **Coeficiente de dispersión cuartil**: Este se define como:

   $$\frac{Q_3 - Q_1}{Q_3 + Q_1}$$

   donde Q_1 y Q_3 son el primer y tercer cuartil respectivamente. Esta medida permite comparar los rangos de distintos conjuntos de datos aunque también es importante notar que es sensible a la presencia de valores extremos.

2. Distribución [Q]:

   a) **Gini**: El coeficiente de Gini es una medida de dispersión definida como la media de las diferencias absolutas entre todos los posibles pares de individuos de una población para una medida dada.

   $$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \overline{x}}$$

   Donde n es la cantidad de medidas y $\overline{x}$ es la media aritmética. El valor mínimo es 0 cuando todas las medidas son iguales, esto puede ser utilizado para medir la homogeneidad/heterogeneidad de los datos.

   b) **Skewness**: Esta medida describe la asimetría de la distribución de acuerdo a la media, indicando la dirección y la magnitud relativa de la desviación de la distribución tomando como referencia una distribución normal. Es el tercer momento estándar definido como:

   $$\tilde{\mu}_3 = E\left[\left(\frac{X - \mu}{\sigma}\right)^3\right]$$

   c) **Curtosis**: Esta medida permite describir el comportamiento de los datos en la cola de la distribución, esto permite obtener una medida de que tan susceptible es la distribución a

la aparición de valores extremales. Es el cuarto momento estándar definido como:

$$\tilde{\mu}_4 = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]$$

d) **Momentos de orden superior**: Estas medidas tienen en cuenta combinaciones no lineales de los datos y pueden ser utilizados para caracterizar las características de la curva.

$$\tilde{\mu}_k = E\left[\left(\frac{X-\mu}{\sigma}\right)^k\right],\ k > 4$$

3. Distribución [Q, C]

a) **Entropía**: Se refiere a la definición de *entropía* en el campo de teoría de la información. Esta medida fue utilizada en [**?**] para establecer un *ranking* entre histogramas de acuerdo a la uniformidad de la distribución utilizando la siguiente definición. Sea un histograma de k intervalos entonces la entropía del histograma h es

$$H(h) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

donde p_i es la probabilidad de que un elemento pertenezca al i-ésimo intervalo. Un alto valor de entropía se asocia a que los elementos pertenecen a una distribución uniforme y que el histograma tiende a ser plano.

b) **Normalidad**: Esta medida evalúa si los datos provienen de una distribución normal, es el resultado de aplicar el *test* propuesto por D'Agostino and Pearson [**?**]. Debido a que para probar la hipótesis se necesita del p valor este también es considerado como *feature*.

c) **Valores extremos**: Brinda información acerca de la presencia de valores extremos en los datos identificados mediante distintos criterios de acuerdo a el rango intercuartil (IQR), los percentiles y la desviación estándar.

Los *features* escogidos se escogieron de acuerdo a la relevancia mostrada por los mismos en los estudios previos realizados por [**?**] y [**?**], priorizando aquellos que estén relacionados con heurísticas utilizadas en el campo de la visualización de datos o que tengan un impacto en la representación visual de los datos.

2.2.2 Procesamiento de los datos

El *Plotly Dataset* obtenido en [**?**] recoge un superconjunto de estos *features* por lo que se utilizaron estos datos ya obtenidos. Se realizó un muestreo uniforme del conjunto para balancear la cantidad de puntos de datos por cada clase para evitar sesgos en el clasificador y se estandarizaron los datos para poder realizar una mejor comparación utilizando la fórmula:

$$x' = \frac{x - \overline{x}}{s}$$

Luego se crearon cinco *folds* sobre el conjunto de datos para entrenar y validar los modelos.

2.3 Modelos Utilizados

Existen diversos modelos utilizados en tareas de clasificación en la literatura los cuales se clasifican en tres tipos los cuales presentamos a continuación y listamos los algoritmos seleccionados por cada tipo.

1. Modelos Multiclase: Los modelos multiclase son aquellos que soportan de manera inherente la existencia de múltiples clases.
   a) Decission Tree
   b) K-Neirest Neighbors
   c) Naive Bayes
   d) Random Forest
   e) Redes Neuronales
2. One-Vs-One: Son modelos los cuales solo soportan clasificación binaria por tanto cuando se desea clasificar un objeto x teniendo n categorías posibles se construyen $\frac{n(n-1)}{2}$ clasificadores binarios los cuales analizan todos los pares de clases y por cada par escogen la clase más probable y votan por esta, retornándose la clase que más votos haya recibido.
   a) Gradient Boosting
   b) SVC
3. One-Vs-All: Al igual que los modelos One-vs-One solo soportan clasificación binaria pero difiere la estrategia para adaptarlos a clasificación multiclase, para ello se construye un clasificador binario por clase el cual se encarga de predecir si el elemento pertenece a dicha clase o no y se escoge la predicción del clasificador con mayor confiaza.
   a) Logistic Regression

   El modelo de *Deep Learning* creado en este trabajo es una red neuronal básica con una capa de entrada, una capa intermedia ReLu y una capa de salida cuya dimensión es igual a la cantidad de categorías con función de activación sigmoid.

   Para optimizar los hiperparámetros de los modelos se utilizó un algoritmo de búsqueda exhaustiva sobre una matriz de parámetros.

2.4 Resultados obtenidos

Para evaluar los resultados del modelo se utilizaron las medidas de *Accuracy* **ACC**, *ROC Area Under Curve* **ROC** y *Logarithmic Loss* **LL**, los experimentos fueron realizados en 5 *folds* sobre un corpus de 5×10^4 elementos.

La ACC de los modelos fue muy baja siendo el mejor resultado de 0.4552 por el modelo de Random Forest, sin embargo, se obtienen altos valores de ROC por lo que la distintinción de clases mediante los features resulta clara para los modelos utilizados, este hecho unido a que la medida LL es alta condujo a la hipótesis de que las probabilidades asignadas a las clases por los modelos eran bajas. Esto se comprobó con el modelo de mejores resultados el promedio de la probabilidad de la clase escogida en los conjuntos de validación fue de 0.45 por lo que que la probabilidad media de que el elemento no pertenezca a la clase seleccionada es de 0.55, lo cual nos lleva a argumentar que el modelo devuelve la opción menos mala entre las disponibles.

Se plantean diversos motivos por los cuales esto puede ocurrir:

1. El corpus utilizado fue de un tamaño reducido, casi un tercio del utilizado en [**?**], esta opción se considera la menos prometedora ya que la ACC del modelo propuesto en dicho trabajo para nuestro problema es de menos de 0.5
2. Los *features* utilizados no permiten particionar el conjunto de configuraciones gráficas de forma correcta, esto planteameanto se ve respaldado por la existencia de heurísticas en la visualización de datos que se apoyan en las relaciones entre columnas del *dataset* para determinar el tipo de visualización.
3. La estandarización llevada a cabo en los datos utiliza propiedades calculadas utilizando todo el corpus, intentar estandarizar las propiedades estadísticas de diversos dominios de datos pudiese resultar en propiedades de ciertos dominios siendo descartadas debido a que son muy bajas en comparación con otros.
4. El *Plotly Dataset* no es un corpus creado por expertos en visualización de datos por lo que la existencia de errores y malas prácticas en este corpus que puedan a llevar a errores de predicción no puede ser descartada.

Modelo	Accuracy	ROC	Log Loss
KNN	0.4159	0.7499	5.228
DT	0.4226	0.7728	1.611
NB	0.3235	0.6919	7.2495
RF	0.4552	0.8025	1.3466
NN	0.4099		
MLP	0.4321	0.786	1.4023
GB	0.4505	0.8002	1.3506
SVC	0.4342	0.7741	1.4325
LR	0.4132	0.7659	1.45

Figura 1: Resultados obetenidos por los algoritmos. (**KNN**) K-Neirest Neighbors, (**DT**) Decision Tree, (**NB**) Naive Bayes, (**RF**) Random Forest, (**NN**) Red neuronal implementada, (**MLP**) Multi-layer Perceptron, (**GB**) Gradient Boosting, (**SVC**) Support Vector Machine, (**LR**) Logistic Regression.

3. Conclusiones

En este trabajo se ha definido el problema de selección de configuraciones gráficas como un problema de clasificación bajo el paradigma de aprendizaje supervisado y se ha realizado una comparación de distintos modelos presentes en la literatura para la realización del mismo. Los resultados obtenidos fueron malos en todas las medidas utilizadas para su evaluación sin embargo, esto nos ha permitido identificar causas potenciales que pueden llevar al pobre rendimiento de un sistema de recomendación de visualizaciones.

4. Recomendaciones y trabajo futuro

Se propone incorporar *features* de pares de columnas y *features* relacionados con el *dataset* y probar representaciones alternativas para modelar este tipo de relaciones entre columnas como un grafo de conocimientos sobre el cual se puedan aplicar algoritmos de clasificación de nodos. Además se propone incorporar nuevos modelos de redes neuronales y optimización de hiperparámetros para estos modelos. Probar estos modelos en un corpus de dominio específico para determinar si la heterogeneidad de dominios es relevante en el resultado de los modelos.