Práctica 7

Luis Fernando Lomelín Ibarra
a01177015@itesm.mx
Tecnológico de Monterrey
Ingeniería en tecnologías computacionales
Monterrey, NL, México

David Alejandro Martínez Tristán
a01610267@itesm.mx
Tecnológico de Monterrey
Ingeniería en tecnologías computacionales
Monterrey, NL, México

ABSTRACT

En el presente documento se exponen los resultados de utilizar Máquinas de Soporte Vectorial (SVM) con diferentes *kernels* al igual que los resultados de los modelos K-NN, Regresión Logística y Bayes Ingenuo sobre un dataset de clasificación de dígitos, con el objetivo de entender mejor la aplicación de SVM y cómo se comparan con los demás modelos anteriormente vistos.

1 Introducción

Una Máquina de Soporte Vectorial (SVM) separa a los puntos de un dataset en el espacio de sus atributos en diferentes clases por medio de uno o un conjunto de separadores llamados hiperplanos, los cuales tienen como meta maximizar el margen geométrico desde ellos hacia los puntos.

2. Conceptos y Herramientas

Conceptos

- Regresión logística
- Clasificación multiclase
- *One-vs-rest* (OVR)
- Algoritmo de K-Nearest Neighbors (KNN)
- Clasificador Naïve Bayes
- Máquinas de Soporte Vectorial (SVM)
- Función de núcleo o *kernel*
- Función lineal
- Función polinomial
- Función RBF
- Función sigmoide
- Tasa de precisión
- Matriz de confusión

Herramientas

- Scikit Learn
- Árboles de Decisión
- Matplotlib
- Graphviz

3 Metodología

La práctica utilizó un dataset de clasificación de dígitos, el cual contiene 1797 muestras de imágenes de 8x8 pixeles representando dígitos del 0 al 9. El dataset se dividió de forma aleatoria en una proporción 80%-20% para generar el training set y el test set.

En primer lugar, se crearon y se entrenaron 4 modelos de SVM con diferentes *kernels:* lineal, polinomial, RBF y sigmoide.

Posteriormente, se creó y se entrenó un modelo de regresión logística con esquema *one-vs-rest* (OVR), el cual permite que este tipo de modelo pueda ser utilizado para una clasificación multiclase, asignando un problema de clasificación binaria a cada una de las clases.

A continuación, se creó y se entrenó un modelo Naïve Bayes.

Seguidamente, se crearon y se entrenaron 10 modelos KNN con el objetivo de encontrar el mejor valor de K de entre los siguientes para este problema: 1, 2, 3, 5, 10, 15, 20, 60, 75 y 100.

Finalmente, se obtuvo la tasa de precisión y la matriz de confusión de cada clasificador.

4 Resultados

4.1 SVM

4.1.1 Lineal

Tasa de precisión	0.9916666666666667

43	0	0	0	0	0	0	0	0	0
0	35	0	0	0	0	0	0	0	0
0	0	36	0	0	0	0	0	0	0
0	0	0	41	0	0	0	0	0	0
0	0	0	0	38	0	0	0	0	0
0	0	0	0	0	30	0	0	1	1
0	0	0	0	0	0	37	0	0	0
0	0	0	0	0	0	0	36	0	0
0	0	0	0	0	0	0	0	28	0
0	0	0	0	0	0	0	1	0	33

Figura: Matriz de confusión

La precisión que nos da el SVM Lineal es alta, lo cual nos indica que esta teniendo buenos resultados con el set de pruebas. Esto también es respaldado por la matriz de confusión la cual nos indica que en la mayoría de las clases, el modelo puede predecir bastante bien todos las instancias. Solo existen excepciones en las clases 6 y 9 las cuales solo se clasifican mal dos y una instancia respectivamente.

Se puede concluir que el SVM Lineal puede ser una buena opción para analizar datos similares a este dataset y se puede obtener resultados considerablemente buenos. También es importante mencionar que existe la probabilidad de que este haciendo overfitting ya que el valor de la tasa de precisión es casi perfecto.

4.1.2 Polinomial

Tasa de precisión	0.9888888888888889

42	0	0	0	0	0	0	0	0	0
0	35	0	0	0	0	0	0	0	0
0	0	36	0	0	0	0	0	0	0
0	0	0	41	0	0	0	0	0	0
1	0	0	0	38	0	0	0	0	0
0	0	0	0	0	30	0	0	0	1
0	0	0	0	0	0	37	0	0	0
0	0	0	0	0	0	0	36	0	0
0	0	0	0	0	0	0	0	28	0
0	0	0	0	0	0	0	1	1	33

Figura: Matriz de confusión

Se puede fácilmente notar que el SVM Polinomial se adapta muy bien al dataset. Esto se debe a que tiene una tasa de precisión de 0.9888888888888889. De la misma manera, la matriz de confusión nos demuestra que el modelo polinomial es casi perfecto en clasificar los elementos, ya que la mayoría de las instancias se ubican en la diagonal principal.

Esto nos indica que el modelo polinomial también funciona muy bien con este dataset, pero hay que tomar en consideración que este haciendo overfitting dado a que la precisión es casi perfecta.

4.1.3 RBF

Tasa de precisión	0.9916666666666667

43	0	0	0	0	0	0	0	0	0
0	35	0	0	0	0	0	0	0	0
0	0	36	0	0	0	0	0	0	0
0	0	0	41	0	0	0	0	0	0
0	0	0	0	38	0	0	0	0	0
0	0	0	0	0	30	0	0	0	1
0	0	0	0	0	0	37	0	0	0
0	0	0	0	0	0	0	37	0	0
0	0	0	0	0	0	0	0	28	0
0	0	0	0	0	0	0	1	1	33

Figura: Matriz de confusión

El modelo SVM RBF da resultados muy similares al modelo SVM Lineal. Al igual que el modelo lineal, presenta una tasa de precisión de 0.9916666666666667. También su matriz de confusión nos indica que no tiene problemas muy grandes al clasificar las instancias.

Similarmente todos los modelos SVM que hemos vistos, presentan una precisión y resultados muy perfectos lo cual nos indica la posibilidad de overfitting.

4.1.4 Sigmoide

Tasa de precisión	0.9111111111111111

41	0	0	0	1	0	0	0	0	1
0	29	1	2	2	1	0	0	0	1
0	0	34	0	0	0	0	0	0	0
0	0	0	37	0	0	0	0	0	0
2	1	0	0	34	0	0	0	0	0
0	0	0	0	0	29	0	1	1	0
0	0	0	0	1	0	37	0	0	0
0	3	0	0	0	0	0	35	0	2
0	1	1	2	0	0	0	0	22	0
0	1	0	0	0	0	0	1	6	30

Figura: Matriz de confusión

El modelo SVM sigmoide es el modelo de los SVM con la menor tasa de precisión. Este nos da una precisión de 0.9111111111. Esto también se ve reflejado en la matriz de confusión, en la que la mayoría de las clases se puede ver que al menos una instancia la clasifica erróneamente.

A pesar de todo esto los resultados que presenta son bastante buenos y probablemente es uno de los modelos que mejor se pueda adaptar a nuevos datos similares al dataset que se utilizo.

4.2 Regresión logística

Tasa de precisión	0.9666666666666667

42	0	0	0	0	0	0	0	0	0
0	34	1	0	0	0	0	0	0	0
0	0	34	0	0	0	0	0	0	0
0	0	1	39	0	1	0	0	0	0
1	0	0	0	38	0	0	0	0	0
0	0	0	0	0	29	0	0	1	1
0	0	0	0	0	0	37	0	0	0
0	0	0	0	0	0	0	36	0	0
0	1	0	1	0	0	0	0	27	1
0	0	0	1	0	0	0	1	1	32

Figura: Matriz de confusión

1.	0	0	0	0	0	0	0	0	0
0	0.97	0.03	0	0	0	0	0	0	0
0	0	1.	0	0	0	0	0	0	0
0	0	0.02	0.95	0	0.02	0	0	0	0
0.03	0	0	0	0.97	0	0	0	0	0
0	0	0	0	0	0.94	0	0	0.03	0.03
0	0	0	0	0	0	1.	0	0	0
0	0	0	0	0	0	0	1.	0	0
0	0.03	0	0.03	0	0	0	0	0.9	0.03
0	0	0	0.03	0	0	0	0.03	0.03	0.91

Figura: Matriz de confusión Normalizada

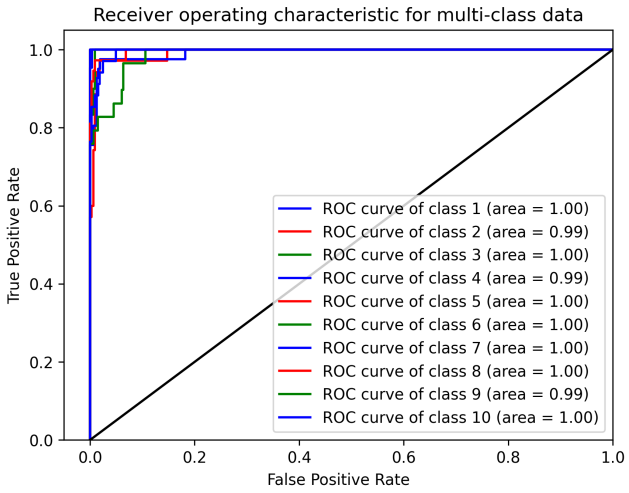

Figura: Curva ROC de todas las clases del Modelo Regresión Logística

Se puede observar que con este dataset el modelo de regresión logística funciona bastante bien. Esto está indicado por su tasa de precisión alta (0.966667) y por cómo resultó la matriz de confusión. En la matriz de confusión se puede admirar que la mayoría de las instancias se encuentran en la diagonal principal esto nos indica que el clasificador está correctamente prediciendo la clase de la instancia.

Además la curva ROC, la cual muestra el área debajo de la curva por clase, nos indica que está prediciendo bastante bien todas las clases. La mayoría indicando que el clasificador es perfecto para su clase.

Esto nos daría confianza para que el clasificador se pueda desempeñar bien con más datos similares al dataset. Pero cabe recalcar que como todos los casos que marcan que el clasificador es perfecto, se puede tener un caso de overfitting y que se pueda ver afectado significativamente con datos nuevos pero similares al dataset.

4.3 KNN

Mejor valor para K	3
Tasa de precisión	0.994444

43	0	0	0	0	0	0	0	0	0
0	35	0	0	0	0	0	0	0	0
0	0	36	0	0	0	0	0	0	0
0	0	0	41	0	0	0	0	0	0
0	0	0	0	38	0	0	0	0	0
0	0	0	0	0	29	0	0	0	0
0	0	0	0	0	0	37	0	0	0
0	0	0	0	0	0	0	36	0	0
0	0	0	0	0	0	0	0	29	0
0	0	0	0	0	1	0	1	0	34

Figura: Matriz de confusión

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0.03	0	0.03	0	0.94

Figura: Matriz de confusión Normalizada

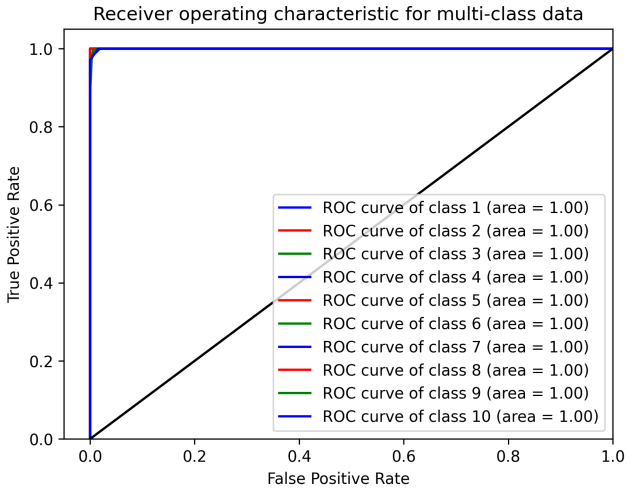

Figura: Curva ROC de todas las clases del Modelo K-NN

De todos los K-NN probados nuestro modelo nos regresó que el mejor de todos es cuando K es igual a 3. Este modelo cuenta con una tasa de precisión de 0.994444, lo cual es bastante bueno. Esto apunta a que con 3 vecinos puede muy seguramente clasificar muy bien a la mayoría de las instancias. Esto también se ve reflejado en la matriz de confusión la cual nos demuestra que este modelo predice casi perfectamente todas las clases, a excepción de la décima clase la cual solo dos instancias las clasificó mal.

Finalmente la curva ROC, la cual nos demuestra el rendimiento por clase, nos indica que el modelo generó clasificadores perfectos para todas las clases del dataset. Dado a que nos da resultados casi perfectos nosotros sospechamos que tal vez estemos contando con un caso de overfitting y que existe una buena probabilidad de que el modelo no se pueda adaptar bien con más datos.

4.4 Naïve Bayes

Tasa de precisión	0.8527777777777777

41	0	0	0	1	0	0	0	0	0
0	29	3	1	0	1	0	0	0	1
0	0	23	1	0	0	0	0	0	0
0	0	1	32	0	0	0	0	1	1
1	0	0	0	35	0	0	0	0	2
0	0	0	2	0	25	0	0	0	1
0	0	0	0	1	0	37	0	0	0
1	0	0	2	0	2	0	37	0	5
0	6	9	3	1	2	0	0	28	4
0	0	0	0	0	0	0	0	0	20

Figura: Matriz de confusión

0.98	0.	0.	0.	0.02	0.	0.	0.	0.	0.
0.	0.83	0.09	0.03	0.	0.03	0.	0.	0.	0.03
0.	0.	0.96	0.04	0.	0.	0.	0.	0.	0.
0.	0.	0.03	0.91	0.	0.	0.	0.	0.03	0.03
0.03	0.	0.	0.	0.92	0.	0.	0.	0.	0.05
0.	0.	0.	0.07	0.	0.89	0.	0.	0.	0.04
0.	0.	0,	0.	0.03	0.	0.97	0.	0.	0.
0.02	0.	0.	0.04	0.	0.04	0.	0.79	0.	0.11
0.	0.11	0.17	0.06	0.02	0.04	0.	0.	0.53	0.08

0.	0.	0.	0.	0.	0.	0.	0.	0.	1.

Figura: Matriz de confusión Normalizada

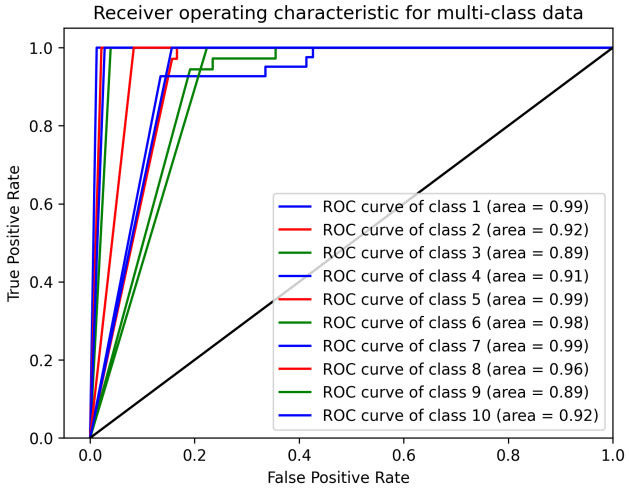

Figura: Curva ROC de todas las clases del Modelo Naïve Bayes

A diferencia de todos los modelos previamente vistos, el Bayes Ingenuo es el que menor tasa de precisión tiene de todos los modelos de la práctica. A pesar de esto, la matriz de confusión nos indica que el modelo de todos modos tiende a clasificar bien la mayoría de las instancias.

Analizando la curva ROC del modelo, podemos ver que para la mayoría de las clases el modelo es bastante bueno. Tomando todo esto en consideración es posible que este modelo pueda adaptarse mejor a más datos que los demás modelos. Pero esto es solo considerando los valores que arrojó el modelo. También hay que tener en cuenta que el modelo de Bayes Ingenuo puede hacer overfitting con pocos datos y puede que no escale tan bien con más datos. Esto solo se podría probar por medio de más experimentación con el modelo.

5. Conclusiones y Reflexiones

Luis:

Fue interesante ver en esta práctica como funcionan los SVM. Primero podemos ver como se adaptan mejor unos que otros con el mismo dataset. Esto, yo creo que puede variar mucho dependiendo del dataset y de como se acomoden los datos en el mismo. En este dataset vimos que se acomodo bastante bien a todos los SVM.

También fue un gran reto lidiar con cómo representar la curva ROC con un dataset multiclase. Al principio batallamos para encontrar como aplicarlo ya que

como habíamos investigado en prácticas anteriores, la curva ROC funciona para clasificadores binarios. Tras investigar me sorprendió que se puede aplicar para un clasificador multiclase, pero se tiene que hacer las comparaciones clase por clase. Es una solución bastante lógica pero no creo que hubiera podido llegar a ella a tiempo sin investigarlo.

Otra cosa que me llamó la atención es que el clasificador de Bayes Ingenuo no hizo overfitting muy fuerte. Como habíamos visto en clase, es probable que este modelo haga overfitting y que requiere de un dataset muy grande, pero en este caso funcionó bastante bien.

David:

La presente práctica me pareció muy interesante en particular, ya que integró varios modelos de clasificación vistos anteriormente y se comparó sus tasas de precisión utilizando un mismo dataset.

Así mismo, fue una gran experiencia de aprendizaje comparar cómo se comporta una SVM según el *kernel* que se le asigne y qué tan preciso es. También fue sorprendente aprender cómo se podía adaptar un modelo de regresión logística a una clasificación multiclase utilizando un esquema *one-vs-rest*.

La curva ROC permitió analizar con mayor detalle cada uno de los modelos complementarios y concluir que hay clasificadores que pueden ser más o menos precisos dependiendo de su aplicación y de la naturaleza del dataset.

Considero que es una de las mejores prácticas poner a prueba diferentes modelos con un dataset de tamaño controlado para determinar cuál es el más adecuado para la tarea antes de ponerlo a trabajar en un ambiente de producción donde se toman decisiones reales con los resultados obtenidos

REFERENCES

Scikit Learn. (2021). *Gaussian Naive Bayes (GaussianNB)*. Recuperado de https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

Yildirim, S. (2020). *ROC Curve and AUC — Explained*. Towards Data Science. Recuperado de https://towardsdatascience.com/roc-curve-and-auc-explained-8ff3438b3154

Bhandari, A. (2020). *AUC-ROC Curve in Machine Learning Clearly Explained*. Analytics Vidhya. Recuperado de https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Loukas, S. (2020). *ROC Curve explained using a COVID-19 hypothetical example: Binary & Multi-Class Classification tutorial*. Towards Data Science. Recuperado de https://towardsdatascience.com/roc-curve-explained-using-a-covid-19-hypothetical-example-binary-multi-class-classification-bab188ea869c

Garcia, F. (2018). *Clasificación de Fallas en Rodamientos de Máquinas Rotativas Utilizando Aprendizaje de Máquinas*. 10.13140/RG.2.2.33614.41284. Recuperado de https://www.researchgate.net/figure/Figura-40-a-Matriz-de-confusion-sin-normalizar-b-Matriz-de-confusion-normalizada_fig11_334821433

Scikit Learn. (2021). *Confusion matrix*. Recuperado de https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Philipp, P., Schreiter, L, Giehl, J., *et al*. (2016). Situation Detection for an Interactive Assistance in Surgical Interventions Based on Dynamic Bayesian Networks. Recuperado de https://www.researchgate.net/figure/Normalized-Confusion-Matrix-A-row-represents-an-instance-of-the-actual-class-ie-an_fig1_308074430

Scikit Learn. (2021). *Label binarize*. Recuperado de https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.label_binarize.html#sklearn.preprocessing.label_binarize

Scikit Learn. (2021). *One vs Rest Classifier*. Recuperado de https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

Scikit Learn. (2021). Receiver Operating Characteristic (ROC). Recuperado de https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html