

Práctica de Reconocimiento de Formas

Curso 2019/20

Enunciado final

Esta entrega final de la primera práctica de Reconocimiento de Formas (habrá otra en enero) consiste en la unión de todas las entregas parciales realizadas durante los meses de septiembre, octubre y primera quincena de noviembre. Se deben explicar los conceptos básicos aprendidos, las decisiones tomadas y el trabajo realizado, así como discutir los resultados obtenidos en cada una de las entregas parciales.

Evaluación

Para la calificación de la práctica se podrá citar a los miembros del equipo para revisar con el profesor los resultados. En este caso será imprescindible la asistencia de todos los componentes del equipo. La calificación de cada miembro puede ser diferente.

Entregables

Se debe entregar a través de Moodle el Python Notebook (fichero .ipynb) con todo el código que implemente los distintos apartados de la práctica, antes del miércoles 20 de noviembre a las 23:55. También se entregará un .pdf con la memoria de la práctica. Si has decidido realizar la memoria en el Notebook the Python, puedes generar automáticamente el documento .pdf con el comando `jupyter nbconvert --execute --to pdf notebook.ipynb`.

Incluye tanto en el .pdf como en el Python Notebook (fichero .ipynb) el nombre de los dos integrantes del grupo, una introducción, unas conclusiones y los apartados que se describen a continuación.

Apartados

Cada apartado está relacionado con una parte de la asignatura y con una o más entregas realizadas a lo largo de la evaluación continua. Por ello, consultar los esqueletos proporcionados y los mensajes enviados con las tareas de cada semana te será de ayuda para comprender lo que se pide en cada apartado.

La práctica se compone de las siguientes secciones, que deben estar reflejadas tanto en la memoria como en el Python Notebook (fichero .ipynb) entregado:

1. Introducción

Breve presentación de la práctica.

2. Clasificador de la distancia euclídea

- Describe muy brevemente en qué consiste el clasificador y cómo lo has implementado.
- Implementación del clasificador de la distancia euclídea según el estilo de programación Python-Numpy visto en clase.
- Tabla de resultados. Discusión.

3. Clasificador Estadístico Bayesiano

- Describe muy brevemente en qué consiste el clasificador y cómo lo has implementado.

- Implementación del clasificador estadístico bayesiano. Emplear el estilo de programación Python-Numpy visto en clase, si es posible.
- Rellena la tabla de resultados. Discute los resultados obtenidos en comparación con el clasificador de la distancia euclídea.

4. Regularización.

- Explica brevemente en qué consiste la regularización y justifica su necesidad.
- Implementa el clasificador Estadístico Bayesiano empleando dos parámetros de regularización: Un parámetro booleano que determine si se comparten o no las matrices de covarianzas y un parámetro “shrinkage” que regulariza la estimación de la matriz de covarianzas.

5. Evaluación del Rendimiento

Empleando el código de soporte que se ha proporcionado en los esqueletos de las prácticas parciales, entrena y evalúa el rendimiento de los clasificadores de la distancia euclídea y del clasificador estadístico bayesiano con regularización, empleando validación cruzada *k-fold*, en las bases de datos de Iris, Wine y Cancer.

Compara los resultados con los obtenidos en las secciones anteriores y discute las diferencias.

Empleando el código de soporte que se ha proporcionado en los esqueletos de las prácticas parciales, entrena, encontrando los parámetros de regularización que obtienen el mejor rendimiento para el clasificador bayesiano paramétrico, y evalúa el rendimiento de los clasificadores de la distancia euclídea y del clasificador bayesiano paramétrico, empleando el método de exclusión en las bases de datos MNIST e Isolet. Compara los resultados con los obtenidos en las secciones anteriores y discute las diferencias.

Para entrenar un clasificador y hacer una correcta evaluación de su rendimiento es necesario que el conjunto de datos que se utiliza para evaluación (conjunto de test) no se haya utilizado en ninguna etapa del entrenamiento. En caso contrario estaríamos sobre-ajustando a dicho conjunto de test. En este apartado, para simplificar el proceso se os ha propuesto entrenar y seleccionar los parámetros de regularización empleando un método wrapper que optimizase el rendimiento del conjunto de test. Razona por qué esta estimación del rendimiento está sesgada y qué habría que hacer para obtener una estimación más ajustada.

6. Aplicación en un caso real de reconocimiento de texto

Emplea los clasificadores construidos en el apartado anterior para crear un programa capaz de reconocer el texto presente en una hoja de texto escaneada como la que se puede encontrar en el fichero “EJEMPLO_PRACTICA6.JPG” de la práctica 6. Puedes emplear todo el código de soporte de la Práctica 6, si así lo deseas.

Describe el proceso de análisis que has seguido para mejorar tus resultados respecto a los resultados de partida. Motiva cada una de las decisiones tomadas. Describe cómo se han evaluado los resultados, si se han empleado otras imágenes tomadas por el estudiante y si es así cómo han cambiado los resultados con respecto a los resultados del fichero “EJEMPLO_PRACTICA6.JPG”. Proporciona el resultado obtenido sobre la imagen “EJEMPLO_PRACTICA6.JPG”.

6. Conclusiones

Discusión final sobre los resultados obtenidos.

También son bienvenidos los comentarios que quieras hacer sobre cualquier aspecto del curso.