Curso de aprendizaje automatizado

PCIC, UNAM

Tarea 5 (opcional): Modelos de variables latentes

Fecha límite: 30 de junio.

Formato: Archivo ZIP con código fuente y reporte.

Forma de entrega: Enviar correo a gibranfp@unam.mx, bere.mcic@gmail.com y richardt.pcic@gmail.com con asunto *Tarea de modelos de variables latentes* y archivo adjunto.

Ejercicio 1

Entrenar un modelo que pueda clasificar diferentes imágenes de escenas. Para resolver este problema tendrás que implementar la técnica conocida como *bag of features* [**?**], la cual está inspirada en técnicas de procesamiento del lenguaje natural y es popular para la clasificación de imágenes. El procedimiento básico de clasificación de imágenes basado en esta técnica es el siguiente:

1. Usando un algoritmo de visión por computadora, se extraen puntos sobresalientes de las imágenes y vectores que capturan información sobre sus pixeles vecinos.
2. Se agrupan los vectores extraídos (comúnmente con k-means) para encontrar cuáles son similares. A cada centro encontrado por el algoritmo se le denomina palabra visual (*visual word*) y a todo el conjunto se le conoce como vocabulario.
3. Se calcula un histograma que mide la frecuencia de palabras visuales que tiene cada imagen. Este histograma es la nueva representación de la imagen. Es importante normalizar los histogramas para que se conserve la distribución general de cada clase.
4. Opcionalmente es posible aplicar técnicas de reducción de dimensionalidad a los vectores calculados por el algoritmo de extracción de características y/o a los histogramas de frecuencias. De esta manera se puede evitar el uso de variables cuya varianza es mínima entre los datos.
5. Finalmente, se entrena el clasificador con los histogramas generados.

La tarea consiste en entrenar tres diferentes configuraciones del modelo[1]. En la primera configuración construye el vocabulario visual usando el algoritmo k-means con número de centros igual a 1000[2],

[1] Se recomienda guardar los modelos en disco para no tener que entrenarlos en cada ejecución de tu programa (véase `http://scikit-learn.org/stable/modules/model_persistence.html`).

[2] Debido al número de características SIFT se recomienda usar la función `MiniBatchKMeans` de scikit-learn para reducir el tiempo de ejecución del agrupamiento.

asimismo debes construir el histograma de frecuencias para cada imagen y entrenar un clasificador de tu elección[3] (puede ser Naive Bayes, Logistic Regression o SVM). Para la siguiente configuración debes usar PCA para reducir la dimensionalidad de los vectores SIFT y construir el vocabulario con los vectores reducidos. Finalmente, debes usar PCA no solamente para los vectores SIFT sino también para reducir la dimensionalidad del histograma de frecuencias. En el reporte debes mostrar las matrices de confusión de cada uno de los clasificadores y reportar su respectiva precisión. También debes indicar el número de componentes que usaste después de la reducción y justificar, de manera adecuada, esa elección. Adicionalmente contesta las siguientes preguntas:

- ¿Cuáles consideras que son algunas de las ventajas y desventajas de este método y por qué?
- Define, en el contexto de procesamiento del lenguaje natural, qué es una *stop word* y explica cómo determinarías *stop words* en este problema.

Base de datos

La base de datos consiste en imágenes que pertenecen a 15 categorías de lugares diferentes y se encuentran divididos en conjuntos de entrenamiento (100 imágenes por categoría) y de validación[4]. Se extrajeron características por cada imagen usando el algoritmo SIFT [**?**] y sus vectores de descripción con dimensionalidad igual a 128 fueron almacenados en archivos binarios de numpy[5] (`.npy`) dentro del directorio `features`. El algoritmo SIFT describe cada punto de interés usando información sobre la magnitud y orientación del gradiente de sus pixeles vecinos.

Ejercicio 2

Calcula la representación de bolsa de palabras con *tf-idf* de la base de datos de *20 newsgroups* y realiza lo siguiente:

1. Compara el rendimiento de la SVM con kernels coseno, lineal y RBF.
2. Prueba que el kernel coseno es Mercer

$$k(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^\top \mathbf{x}'}{\|\mathbf{x}\|_2 \cdot \|\mathbf{x}'\|_2}$$

Ejercicio 3

Investiga lo siguiente:

1. Considera la tarea de clasificar secuencias de proteína de longitud variable, ¿cómo entrenarías un SVM con este tipo de datos y cómo lo harías con regresión logística? Da un ejemplo de un kernel apropiado para esta tarea y describe las ventajas y desventajas de cada clasificador.

[3]Para evitar el desborde de memoria, recuerda liberar el espacio de variables que ya no se utilicen (por ejemplo, en Python puedes eliminar una variable con `del nombre_variable`).

[4]Disponible en `https://www.dropbox.com/s/fz9znz7ph4ahca1/imagedb.zip`

[5]Para cargar los archivos binarios de NumPy puedes usar la función `numpy.load("nombre.npy")`.

2. ¿Cuál es la distancia en términos de $\{\mathbf{w}, b\}$ entre los 2 hiperplanos de soporte en un SVM lineal y por qué el problema de optimización primario está dado por la siguiente ecuación?

$$\begin{aligned} &\text{minimiza } \frac{1}{2}\|\mathbf{w}\|_2^2 \\ &\text{sujeto a } y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + b) \geq 1, i \in [1, n] \end{aligned}$$

3. Describe el proceso para obtener la representación dual del problema de optimización de las máquinas de vectores de soporte.