Tareas Curso 2023-24

Todas las tareas propuestas requieren que se de una pequeña introducción teórica.

A) Multi-Label

Tarea A-1:Para entrenar una red neuronal para una tarea multi-label es suficiente con aplicar una función *softmax* a las salidas de la red y aplicar una función de coste *binary cross-entropy* en el entrenamiento.

- En la introducción teórica, justificar por qué esto es así.
- Implementar una red neuronal (*deep*) para resolver problemas multi-label. Se puede usar cualquier *backend*: Keras, Tensorflow, PyTorch, JAX.
- Probar las redes implementadas en un dataset multi-label generado a partir de juntar imágenes de MNIST y Fashion-MNIST (https://github.com/zalandoresearch/fashion-mnist).

B) Online learning

Tarea B-1: Implementar y probar los algoritmos kernelized perceptron y kernelized OGD del paper *Online Learning: A Comprehensive Survey*, Steven C.H. Hoi et al., pag. 26.

Algorithm 10: Kernelized Perceptron

INIT: $f_0 = 0$
for $t = 1, 2, \ldots, T$ **do**
  Given an incoming instance $\mathbf{x}_t$, predict $\hat{y}_t = \text{sgn}(f_t(\mathbf{x}_t))$;
  Receive the true class label $y_t \in \{+1, -1\}$;
  if $\hat{y}_t \neq y_t$ **then**
    $\mathcal{SV}_{t+1} = \mathcal{SV}_t \cup (\mathbf{x}_t, y_t)$, $f_{t+1} = f_t + y_t \kappa(\mathbf{x}_t, \cdot)$;
  end if
end for

Algorithm 11: Kernelized OGD

INIT: $f_0 = 0$
for $t = 1, 2, \ldots, T$ **do**
  Given an incoming instance $\mathbf{x}_t$, predict $\hat{y}_t = \text{sgn}(f_t(\mathbf{x}_t))$;
  Receive the true class label $y_t \in \{+1, -1\}$;
  if $\ell_t(f_t) > 0$ **then**
    $\mathcal{SV}_{t+1} = \mathcal{SV}_t \cup (\mathbf{x}_t, y_t)$, $f_{t+1} = f_t - \eta_t \nabla \ell_t(f_t(\mathbf{x}_t)) = f_t - \eta_t \ell'_t \kappa(\mathbf{x}_t, \cdot)$;
  end if
end for

- Probar con problemas juguete como two-moons, two-circles.
- Comparar resultados online con offline: accuracy, number of SVs, etc.
- Probar con problemas de clasificación reales: MNIST, etc.

C) Semi-Supervised Learning

Tarea C-1: Extender a problemas de dos dimensiones el método mezcla de Gaussianas semi-supervisado que utiliza el algoritmo *Expectation-Maximization* (EM).

- Generar un problema consistente en dos distribuciones Gaussianas de 2 dimensiones con distintas medias y desviaciones estándar en los ejes X/Y. Empezar con dos distribuciones separadas (que no se superpongan, o que lo hagan mínimamente) y pocas muestras etiquetadas (caso extremo, solo una muestra en cada distribución).
- Programar el algoritmo EM y comprobar que resuelve correctamente el problema.
- Una vez comprado que funciona, ponerlo a prueba complicando el escenario haciendo que las distribuciones solapen cada vez más.

Tarea C-2: En clase vimos cómo aplicar *Manifold Regularization* con métodos kernel modificando el kernel mediante el grafo Laplaciano en un problema de ejemplo de clasificación. En esta tarea se pide utilizar esta técnica en problemas de regresión (la modificación del kernel es la misma). La idea es ensayar varios ratios de muestras etiquetadas / no etiquetadas, estudiar el grado de modificación del kernel necesaria (cuanto "deformamos" el kernel a través del parámetro "gamma"), cómo influye la construcción del grafo Laplaciano (ej. número de vecinos si se usa kNN, etc.), grafo no normalizado versus normalizado.

D) Active Learning

Tarea D-1: Los métodos de AL se pueden combinar con métodos de diversidad para intentar mejorar la selección de muestras. En esta tarea se pide:

- Implementar el método AL de muestreo por incertidumbre basado en seleccionar las muestras con mayor entropía (pág. 14 de los apuntes de clase).
- A partir de esta selección probar algunos de los criterios de diversidad:
    - MAO: *most ambiguous and orthogonal*. Se trata de ir seleccionando, de las muestras seleccionadas por el algoritmo AL, las más distantes entre ellas. La medida de distancia dependerá del algoritmo de predicción usado: para un algoritmo lineal se usaría la distancia Euclídea. Para métodos Kernel habría que utilizar el mismo kernel que use el algoritmo.
    - Clustering: separar las muestras en grupos mediante un algoritmo de clustering (probar varios) e ir seleccionando muestras de cada grupo.

Tarea D-2: Un grupo de algoritmos de AL son aquellos basados en comités de expertos. En esta tarea se pide:

- Implementar estrategias basadas en medir la diferencia entre expertos usando la divergencia Kullback-Leibler (pág. 24 de los apuntes):
    - Probar con expertos basados en bootstrap: mismo algoritmo base (árboles de decisión u otros) entrenados con conjuntos generados mediante bootstrap.
    - Probar con expertos basados en distintos algoritmos: árboles (o random forests), SVMs, Neural Networks, etc.

E) Anomalías

Tarea E-1: Entre los métodos de detección de anomalías en imágenes vistos en clase está Padim o Patchcore pero hay muchos más. Uno muy interesante es FastFlow.

- Describe en qué consiste el algoritmo FastFlow para la detección de anomalías en imágenes.
- Aplícalo a un conjunto de imágenes, por ejemplo *bottle* del MVTec, tanto para las tareas de segmentación (de la anomalía) como para la tarea de clasificación. Usa la librería anomalib.

Tarea E-2: La detección de anomalías en series temporales es compleja debido a la relación temporal que existe entre las muestras. Una de las aproximaciones es descomponer esa relación con transformadas (ej. Fourier).

- Describe en qué consiste el algoritmo “Time series outlier detection with Spectral Residuals” https://arxiv.org/abs/1906.03821
- La librería ALIBI Detect tiene la función SpectralResidual que tiene implementado el algoritmo. Aplícalo sobre un conjunto sintético y otro real y evalúa los resultados.

Tarea E-3: El algoritmo DOMINANT de detección de anomalías en grafos está basado en las ideas de redes convolucionales (GNN) y autoencoders (GAE) en grafos.

- Introduce teóricamente las redes convolucionales en grafos (GNN).
- Aplica las GNN en ejemplos sencillos en los que se muestre los conceptos anteriormente introducidos.

Tarea E-4: El algoritmo DOMINANT de detección de anomalías en grafos está basado en las ideas de redes convolucionales (GNN) y autoencoder (GAE) de los grafos.

- Introduce teóricamente los autoencoders en grafos (GAE).
- Realiza ejemplos sencillos de Autoencoders en grafos que muestren los conceptos anteriormente introducidos.

F) XAI

Tarea F-1: En esta tarea se pide implementar una versión básica de *Integrated Gradients*. Este método se basa en calcular la integral de los gradientes desde una imagen de entrada a la salida. La imagen se integra multiplicada por una variable (α) en el intervalo de 0 a 1.

$$\mathsf{IntegratedGrads}_i(x) ::= (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha$$

Aquí *x’* es una imagen de referencia, normalmente *x’ = 0*.
Se puede hacer una aproximación realizando esta integral como un sumatorio para valores discretos de α desde 0 hasta 1 sumando los gradientes obtenidos mediante *Vanilla Gradient / Saliency*.

- Implementar esta versión básica discretizando alpha desde 0 hasta 1 para distintos valores incrementales ($\Delta\alpha$) y comparar con el resultado de aplicar directamente *IntegratedGradients* de la librería Captum.

Tarea F-2: Relacionado con el ámbito de la explicabilidad de modelos están los conceptos de transparencia, ética y justicia en IA.

- Introduce el concepto de grupo vulnerable y posibles técnicas para mitigar sesgos. https://fairmlbook.org/
- La librería DALEX (introducida en clase) contiene funciones para este propósito *model_fairness.* Muestra varios ejemplos de usos de estas técnicas.

Tarea F-3: Los Shapley-Values son ampliamente utilizados para la explicación de la predicción de un modelo. No es la única posibilidad, los valores *break_down* (y su versión con interacciones) son una alternativa.

- Introduce el concepto de los valores *break_down*. Ventajas e inconvenientes frente a los valores Shapley.
- La librería DALEX tiene funciones para calcularlos. Muestra varios ejemplos de aplicación y su comparación con los valores Shapley.
- Estrategias para reducir el tiempo para el cálculo de los valores Shapley.

Tarea F-4: Los valores contrafactuales son interesantes para aprovechar el conocimiento recogido en un modelo de aprendizaje máquina.

- Introduce el concepto de valores contrafactuales y posibles métodos para obtenerlos.
- La librería *Alibi Explain* tiene implementados varios métodos. Muestra varios algoritmos para su cálculo y ejemplos de aplicación.