

Richard Staszekiewicz
Miłosz Łopatto

ZUM projekt - dokumentacja wstępna

Temat projektu

10. Aktywne uczenie się modeli klasyfikacji na podstawie małych zbiorów trenujących przez zgłaszanie zapytania o prawdziwe wartości atrybutu docelowego dla ograniczonej liczby przykładów z dostarczonego dużego zbioru danych nieetykietowanych wybranych według określonych kryteriów (np. przykłady bliskie granicy decyzyjnej dotychczasowego modelu lub takie, dla których jego predykcje są obarczone największą niepewnością) i iteracyjne doskonalenie modelu na podstawie powiększanego w ten sposób zbioru trenującego. Implementacja w formie opakowania umożliwiającego użycie dowolnego algorytmu klasyfikacji dostępnego w środowisku R lub Python stosującego standardowy interfejs wywołania. Badanie wpływu użycia aktywnego uczenia się na jakość modeli klasyfikacji tworzonych na podstawie małych zbiorów trenujących za pomocą wybranych algorytmów dostępnych w środowisku R lub Python.

Interpretacja tematu

Temat zinterpretowano jako polecenie zbudowania biblioteki w języku Python poddające obiekty o interfejsie klasyfikatora zaczerpniętym ze znanej biblioteki scikit-learn uczeniu aktywnemu na małej ilości otagowanych danych.

Opis części implementacyjnej

W ramach implementacji zostaną zrealizowane osobno moduły do algorytmu uczenia aktywnego, interfejsu oraz integracji.

Lista wykorzystanych w eksperymentach algorytmów

Z eksperymentami będą kompatybilne następujące algorytmy z biblioteki **SciKit Learn** w wariancie klasyfikatorów:

- Nearest Neighbors (sklearn.neighbours.KNeighborsClassifier)
- Linear SVM (sklearn.SVM.SVC)
- RBF SVM (sklearn.SVM.SVC)
- Gaussian Process (sklearn.gaussian_process.GaussianProcessClassifier)

- Decision Tree (sklearn.tree.DecisionTreeClassifier)
- Random Forest (sklearn.ensemble.RandomForestClassifier)
- AdaBoost (sklearn.ensemble.AdaBoostClassifier)
- Naive Bayes (sklearn.naive_bayes.GaussianNB)
- QDA (sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis)

Wszelkie zbierane dane numeryczne będące rezultatami działania algorytmu będą analizowane zbiorowo za pomocą funkcji biblioteki **Pandas**, głównie metod klasy `pandas.DataFrame`.

Interfejs użytkownika podczas aktywnego uczenia zostanie zrealizowany prawdopodobnie jako aplikacja webowa przy pomocy biblioteki **Streamlit**.

Plan badań

W ramach projektu planowane jest zrealizowanie dwóch badań na dwóch różnych etapach. Badania Integracyjne na etapie Implementacji, służące do weryfikacji jakości biblioteki oraz jej niezawodności oraz Badania Wpływu Strategii Zapytań na etapie Walidacji, która ukaże różnice pomiędzy stosowaniem różnych rodzajów strategii próbkowania punktów do otagowania.

Cel poszczególnych badań

- **Badania Integracyjne**
Są to badania zaprojektowane by naśladowały testy integracyjne systemu z symulowanym użytkownikiem. Nie będą się one wiązały z wynikami odrębnymi od informacji systemowych i będą prowadzone wyłącznie na Etapie Implementacji w celu weryfikacji poprawnego działania systemu.
- **Badania Wpływu Strategii Zapytań**
Są to badania wpływu różnych strategii wyznaczania punktów do otagowania przez algorytm aktywnego uczenia. Zamierzone jest zbadanie co najmniej 3 strategii: ATS, Uncertainty Sampling oraz Expected Error Reduction.

Charakterystyka zbiorów danych

Planowanym jest używanie dwóch rodzajów zbiorów:

- Zbiór [Titanic](#)
Będziemy dokonywać na nim klasyfikacji binarnej, dokładniej przewidywania wartości klasy `Survived`. Ze względu na prostą możliwość zbudowania automatycznej "wyrocni" niezbędnej w algorytmie uczenia aktywnego, na tym

zbiorze przeprowadzane będą eksperymenty numeryczne wpływu strategii próbkowania na budowę modelu.

- Zbiór [CIFAR10](#)

W ramach biblioteki torchvision, zbiór ten jest reprezentowany poprzez 60000 kolorowych obrazków o wymiarach 32x32 podzielonych na 10 równolicznych klas. Ze względu na jego prostą interpretowalność przez człowieka będzie on wykorzystywany jako zbiór testowy w interaktywnym interfejsie graficznym w Streamlit.

Badane parametry algorytmów

Algorytmy klasyfikujące nie są przedmiotem badania, które skupia się na active learningu i jego parametrze strategii. W związku z powyższym, oprócz tego jednego hiperparametru, reszta pozostanie stała i udnnotowana w dokumentacji projektu.

Miary jakości i procedury oceny modeli

- Badania Integracyjne

Ocenie będzie poddawany przyrost informacyjny dot. stanu kodu oraz czas odpowiedzi systemu. Ocena nastąpi poprzez skonfrontowanie rezultatów ze spodziewanymi.

- Badanie Wpływu Strategii Zapytań

Strategie zapytań będą mierzone na każdym jednostkowym przyroście informacyjnym poprzez ocenę każdego z modeli miarami Accuracy, Negative Log Loss oraz ROC Area Under Curve. Wspólnie tendencje tych trzech metryk będą poddawane analizie porównawczej na późniejszych etapach oceny wpływu parametru selekcji.

Otwarte kwestie wymagające późniejszego rozwiązania (wraz z wyjaśnieniem powodów, dla których ich rozwiązanie jest odłożone na później)

- Ze względu na brak działających modułów, nie stwierdzono jeszcze zasadności i stacku istnienia testów automatycznych.