

ההסתברות נוטה לכיוון אחת מן המשימות – מיצוי (exploitation). בין שני ערכי הקיצון אלו קיים ערך של הפרמטר המביא לפילוג ביניים שמאזן בין מקרי הקיצון, ערך זה הינו הערך האופטימלי. האלגוריתם שאנו מציעים, מבוסס על אלגוריתם הפרספטרון המוכר עבור סיווג בינארי, בשילוב הפילוג אותו בנינו. את אלגוריתם זה אנו מכנים בשם SHAMPO (קיצור של Shared Annotator for Multiple Problems).

בנוסף, אנו מציעים מספר ווריאציות נוספות של אלגוריתם SHAMPO. וריאציה של אלגוריתם אגרסיבי, בו עדכון המודל מתבצע לא רק בעת שגיאה, אלא גם כאשר אין שגיאת סיווג, אך השוליים קטנים מתחת לסף מסויים. וריאציה נוספת, בה נעשה שימוש במידע פריורי על מידת הקושי של משימות הסיווג, בבניית הפילוג על פני המשימות. ישנה וריאציה נוספת של האלגוריתם בה הפרמטר b הינו אדפטיבי, ומשתנה בהתאם למספר העדכונים שבוצעו עבור כל אחת ממשימות הסיווג. לסיום, מוצגת וריאציה המשלבת את אלגוריתם הפרספטרון מסדר שני. עבור כל אחת מווריאציות אלו של האלגוריתם אנו מבצעים אנליזה של עבור חסם הסכום המצטבר של השגיאות בכל צעד.

לאחר מכן אנו מציגים כיצד ניתן להשתמש בגישה זו של פתרון מספר משימות בינאריות במקביל עם פרשן משותף, על מנת לפתור בעיות סיווג מרובה מחלקות באמצעות מספר בעיות המוכרות בשם Contextual Bandits בשילוב אלגוריתם SHAMPO ובצירוף חסמים ספציפיים עבור בעיות אלו. לסיום, אנו מודדים את האלגוריתמים שהצענו על מספר מאגרי מידע שונים: שני מאגרי מידע של זיהוי ספרות בתמונות (OCR) בשם MNIST ו-USPS. מאגר מידע נוסף של זיהוי תנועות קוליות (Vowels Prediction) בשם VJ (Vocal Joystick). ומאגר מידע נוסף המשולב מכמה תתי מאגרים של סיווג מסמכים וניתוח הלך רוח (Sentiment Analysis). אנו מראים כי שימוש באלגוריתם SHAMPO ובחירה מתאימה של הפרמטר b , משפר את הביצועים על ממוצע השגיאות על פני כל המשימות ואף עבור שילוב של משימות הסיווג הבינאריות למשימת סיווג אחת מרובת מחלקות. אנו מראים גם כי אכן למרות שהמשימות הקלות יותר דורשות פחות משוב, השגיאה עליהן אינה פוחתת (לפחות לא באופן משמעותי). אנו מסיימים את חיבור זה בדיון על עבודות וחיבורים המתקשרים לחיבור זה ובמספר דרכים נוספות להרחבת עבודה זו.

שקיבל, אל הפרשן. הפרשן מוגבל ואינו מסוגל לסווג מספר דוגמאות במקביל, אלא דוגמה יחידה בכל צעד. כלומר, המשימות חולקות פרשן משותף וישנו צורך בכל אחד מן הצעדים, להחליט מהי המשימה שאת הדוגמה שהתקבלה עבורה, יתייג הפרשן. לאחר שמתקבל הסיווג הנכון מן הפרשן, האלגוריתם מעדכן את המודל של במשימה המתאימה, על ידי שימוש במשוב שהתקבל מן הפרשן.

משימת האלגוריתם היא להחליט בכל שלב, מהי המשימה (או מהי הדוגמה) עבורה יבקש משוב מן הפרשן. באופן אינטואיטיבי, נעדיף בכל שלב לבקש מן הפרשן לסווג את הדוגמה עבורה הסיווג שנתן האלגוריתם, בעל הוודאות הנמוכה ביותר. באופן זה נוכל להיות בטוחים שלא יבזבז זמן פרשן על דוגמאות ומשימות, עבורן האלגוריתם די בטוח בסיווג שלו והצורך שלו במשוב עבור משימות ודוגמאות אלו הינו פחות. לשם כך, אנו מציעים להשתמש במדד המקובל של הערך המוחלט של השוליים המוגדר כמרחק הדוגמה מן המסווג הלינארי באותו השלב. ניתן לראות בגודל זה כמדד כלשהו עבור מידת הביטחון בסיווג האלגוריתם את הדוגמה. כאשר השוליים גדולים, הביטחון בסיווג של האלגוריתם גדול יחסית, מכיוון שהדוגמה מרוחקת מהמסווג, ואילו כאשר השוליים קטנים יחסית, הביטחון בסיווג קטן יותר. בטענה דומה השתמשו במאמרים נוספים העוסקים בלמידה אקטיבית עבור קבוצת דוגמאות ובמאמרים בנושא דגימה סלקטיבית, על מנת להחליט אלו מן הדוגמאות יש לתייג.

אם כן, על פי מדד השוליים, הבחירה האינטואיטיבית תהיה מן הסתם בכל צעד, הדוגמה והמשימה עבורה השוליים הם הקטנים ביותר מהשוליים של שאר הדוגמאות המשוויכות למשימות השונות באותו הצעד. למעשה, כאשר המודל אינו מדויק מספיק, הסיווג יהיה בקירוב גס מאוד, מה שעלול להוביל לשגיאות בסיווג ובהשלכת גודל השוליים על הביטחון בסיווג. תופעה זו קורה בעיקר בצעדים הראשוניים של האלגוריתם, כאשר המודל עדיין לא הושפע ממספיק דוגמאות ואינו מכויל מספיק. לשם כך, בנינו פילוג על פני כל משימות הסיווג, כאשר בכל צעד מוגרלת המשימה עבורה יש לבקש מהפרשן סיווג. פילוג זה, אכן נותן משקל רב יותר למשימות ככל שהדוגמאות שלהן בעלות שוליים קטנים יותר באותו הצעד. הפילוג המוצע הינו הכללה של מקרי הקצה (exploration-exploitation) באמצעות שימוש בפרמטר חיובי קבוע – b . כאשר פרמטר זה גדול מאוד (שואף לאינסוף), הפילוג הופך לאחיד, וכך מתבצעת דגימה אחידה בכל צעד על פני כל המשימות - גישוש (exploration). כאשר פרמטר זה קטן במיוחד (שואף לאפס), במקרה זה, הפילוג הופך לפילוג דלתא, כאשר כל משקל

תקציר

בלמידה מונחית, צוואר הבקבוק העיקרי הוא הצורך בקבלת הסיווג הנכון עבור המידע. שיטה נפוצה בקטגוריה זו עובדת באופן הבא: ראשית יש צורך לאסוף מידע רב באופן אוטומטי (שלב זה לכשעצמו הוא קל יחסית ובעל מחיר זול), ולאחר מכן להעביר את המידע למשתמש או למומחה על מנת שיתן את הסיווג הנכון עבור כל אחת מן הדוגמאות שאספנו. שלב זה של פירוש וסיווג המידע המתקבל יכול להיעשות באמצעות מיקור חוץ לקהל רחב (כדוגמת פרוייקט ה-recaptcha של גוגל) או על ידי מומחה (כמו ב-Linguistic data Consortium). לאחר מכן, ניתן להשתמש במידע שסווג על מנת לבנות מודלים עבור משימת סיווג יחידה, או מספר משימות. החיסרון בגישה זו הוא שאינה מנצלת באופן אופטימלי את המשאב העיקרי (והיקר בדרך כלל), הלא הוא הפרשן. פעמים רבות, משימות סיווג שונות אינן בעלות דרגת קושי זהה, ישנן משימות קלות יותר וישנן קשות יותר. לכן, יתכן ומתן כל כמות המידע עבור כל המשימות יחד, לסיווג ע"י פרשן אינה תמיד רעיון טוב.

היבט נוסף של בעיה זו הינו הצורך להעביר את המידע למשתמש לצורך סיווג, כך שהמערכת תבקש מהמשתמש את הסיווג הנכון עבור כל דוגמה שמתקבלת. עם זאת, כאשר מספר מערכות מעבירות למשתמש מידע לסיווג במקביל ובאופן בלתי תלוי אחת בשניה, המשתמש יהיה מוצף במידע ולא יהיה מסוגל לתפקד באופן הרצוי.

לדוגמה, לעיתים ישנו הצורך לסווג מידע חדשותי ממספר סוכנויות. אדם יחיד אינו מסוגל להתמודד עם כמות גדולה של מידע במקביל, אלא הוא מוגבל לסיווג מספר יחידות מידע חדשותי בזמן מוגבל. אם כך, כיצד יוחלט איזו מהן יסווג?

הגדרת הבעיה בה ענו עוסקים בחיבור זה, תוכננה להתמודד בדיוק עם בעיות מסוג זה, ובפרט, כיצד למצות באופן האופטימלי את זמן הפרשן המסווג המוגבל.

בחיבור זה, אנו מציעים גישה חדשה של למידת מספר משימות סיווג בינאריות במקביל באופן מקוון, החולקות יחדיו פרשן יחיד. לימוד המשימות מתבצע בצעדים, כאשר בכל צעד, מתקבלות לסיווג מספר דוגמאות, דוגמה אחת מכל משימת סיווג. האלגוריתם נותן עבור כל אחת מהדוגמאות המתקבלות, את הסיווג המתאים ביותר על פי המודל שלמד עד צעד זה. לאחר מכן, נדרש האלגוריתם לעדכן את המודל, על מנת לספק סיווג מדויק יותר בעתיד. לשם כך המודל מעביר את הדוגמאות