המאמר היומי של מייק: 02.05.25  
ON SPEEDING UP LANGUAGE MODEL EVALUATION

המאמר שמנסה לטפל באחת הבעיות הכי מעשיות ופחות מדוברות בעבודה עם LLMs: איך מבצעים הערכת ביצועים יעילה של עשרות או מאות פרומפטים או מודלים על סטים גדולים של שאלות, מבלי לבזבז כמויות לא סבירות של זמן חישוב. כל הערכה כזו דורשת להריץ מודל כבד שעשוי להיות בעל עשרות או מאות מיליארדי פרמטרים על כל דוגמה, עבור כל פרומפט. כשיש מאות פרומפטים ואלפי דוגמאות, אנחנו מדברים על מאות אלפי הרצות, שזה די יקר. זה שלא מדובר פה באימון אלא רק בהערכה וזה מה שהופך את הבעיה לעוד יותר מעצבנת: אנחנו רוצים רק לדעת מי הכי טוב, בלי לשלם את המחיר של להריץ את כולם על הכל.

המאמר מציע שני אלגוריתמים חדשים שמנסים לפתור בדיוק את זה, בצורה חכמה ואדפטיבית. הראשון נקרא (המבוסס על UCB שזה Upper Confidence Bound המפורסם)UCB-E, והוא בעצם מבוסס על רעיונות מהעולם של Multi-Armed Bandits (או MBA בקצרה). כלומר, במקום לבדוק את כל השיטות על כל הדוגמאות, האלגוריתם בונה לכל שיטה תחזית של כמה היא טובה לפי מה שכבר נבדק, ומוסיף לה “בונוס אי-ודאות” (בדומה ל-MCTS) שמעודד לבדוק שיטות(מודל + פרומפט למשל) שעדיין לא נבחנו מספיק. ככה הוא לא רק בוחר את השיטה שנראית הכי מבטיחה, אלא גם לא מזניח שיטות שיכולות להפתיע. עם הזמן, הוא משקיע את מאמצי ההערכה רק בשיטות שבאמת שווה לדעת עליהן משהו.

אבל האתגר האמיתי — והחידוש הגדול של המאמר — מגיע בשיטה השנייה, שנקראת UCB-E-LRF. כאן הכותבים מבינים משהו הרבה יותר עמוק: טבלת הביצועים (שיטות × דוגמאות) אולי נראית כמו מטריצה ענקית שאין ברירה אלא למלא, אבל בפועל יש בה הרבה מבנה. יש דוגמאות שהן די דומות זו לזו, ויש שיטות שמתנהגות בצורה מאוד דומה. כלומר, קיימת קורלציה פנימית, שמאפשרת לחשוב על הטבלה כמטריצה בעלת דרגה(ראנק) נמוכה, כלומר כזו שאפשר לשחזר אותה היטב מתוך חלק קטן יחסי מהערכים. האלגוריתם מנצל את זה בדיוק.

הוא מתחיל ממדגם קטן של תוצאות אמיתיות (למשל רק 5% מהטבלה), ואז מאמן מודל של מטריצת דירוג נמוך, כזה שמקצה לכל שיטה ולכל דוגמה וקטור, כך שהמכפלה שלהם חוזה את התוצאה הצפויה. באופן הזה, האלגוריתם מסוגל לשערך את כל שאר התוצאות שלא נבדקו בפועל (בדומה למערכות המלצה עם low-rank factorization של פעם). מעבר לזה, הוא גם יודע להעריך את חוסר הוודאות של כל אחת מהתחזיות האלה. עם כל סיבוב הוא בוחר איפה הכי משתלם לבדוק שוב: איפה שהתחזית הכי לא ודאית, או איפה שיש פוטנציאל למצוא את השיטה הכי טובה. כך, הוא לומד בהדרגה את המבנה האמיתי של הבעיה, ומפנה את חישובי ההערכה בדיוק למקומות שיכולים להשפיע על ההחלטה.

הגישה עושה שימוש מושכל בתבניות שקיימות בדאטה, ויודעת להכליל מעבר למה שנמדד. היא גם אדפטיבית לגמרי, כלומר משתפרת תוך כדי תנועה, בלי להניח מראש מי תהיה השיטה הטובה. ובעיקר היא מאפשרת לחסוך בין 85% ל־95% מההרצות שהיינו צריכים לעשות בגישה נאיבית. במונחים של עבודה עם LLMs, זה ההבדל בין מערכת שאפשר להריץ על GPU ביתי לבין אחת שדורשת תקציב של אלפי דולרים.

התרשמתי מהשילוב בין כלים מתורת ההחלטות (כמו UCB) לבין שיטות מטריציות מודרניות (כמו factorization), וכמה רחוק אפשר להגיע אם מחברים בין עולמות - מאמר מומלץ!

https://arxiv.org/abs/2407.06172