# Bayesian Hyper Parameter Optimizationאופטימיזציה בייסיאנית של היפר פרמטרים

*תקציר: במהלך ההכשרה, אסכם מספר נושאים "רוחביים" שאינם מתקשרים לפרק ספציפי. הנושא הראשון הוא אופטימיזציה בייסיאנית של היפר-פרמטרים. במסמך זה אסביר מה הבעיה שאופטימיזציה בייסיאנית מנסה לפתור, כיצד זה פותר ואציג ניסוי מצומצם שמראה את היתרון של אופטימיזציה בייסיאנית לעומת שיטות אחרות.*

## רקע

מציאת היפר-פרמטרים הינה בעיה מוכרת בעולם ה DS. רוב האלגוריתמים מתאפיינים במספר רב של קונפיגורציות הרצה, להן השפעה רבה על התוצאות של האלגוריתם. מציאת היפר פרמטרים אופטימליים הינה בעיה קשה שדורשת מספר רב של הרצות. גישות מוכרות לחיפוש הינן grid search ו random search אשר מנסות למצוא את הפתרון האופטימלי על ידי הרצות ערכים שלא מסתמכים על הרצות קודמות. ב Bayesian optimization מנסים לנצל תוצאות של הרצות קודמות על מנת לבחור ערכים חדשים לפרמטרים, מה שמקל על חיפוש במרחבי חיפוש גדולים (scalability) .

## רעיון

ממדלים את פונקציית המטרה ובכל איטרציה בוחרים את הערכים שנראה (לפי המודל) שישפרו את פונקציית המטרה בצורה המשמעותית ביותר. בצורה פורמלית: נרצה למצוא סט ערכים    כאשר היא פונקציית המטרה (למשל, accuracy במהלך ה cross validation), הם מרחב האופציות לערכים של היפר פרמטרים וערכים מסוימים להיפר פרמטרים בהתאמה.

איך ממדלים את פונקציית המטרה? **בעזרת ניסיונות קודמים**. ביותר מילים, המודל משתמש בתוצאות של ערכים קודמים וכדי לבנות מודל הסתברותי שמנסה לשערך (מודל זה נקרא surrogate). השלבים של האופטימיזציה היא:

1. נבנה מודל surrogate
2. עבור תנאי עצירה מסוים:
   1. נמצא ערכים להיפר-פרמטרים שנותנים תוצאה מקסימלית ל surrogate
   2. ננסה את הפרמטרים עם האלגוריתם עצמו ונקבל ערך מפונקציית המטרה האמתית
   3. נעדכן את מודל ה surrogate

ה tradeoff בשיטה הוא שבחירה של הערכים הבאים בכל איטרציה לוקחת יותר זמן (לעומת grid או random) אבל נריץ פחות איטרציות. כלומר, במקרה שהרצה של פונקצית המטרה היא יקרה (למשל, כוללת אימון של מודל על הרבה מידע) נרצה להשתמש בשיטה הבייסיאנית. לעומת זאת, אם המודל הוא פשוט (בלי הרבה היפר-פרמטרים) ואין הרבה מידע אולי נעדיף gridsearch.

## יותר פרטים

בפועל, צריך לשים לב לפונקציה שמגדירה את ה surrogate ולפונקציה שבוחרת את הערכים הבאים לבדיקה. אופציות מוכרות לפונקציית surrogate הן Gaussian Process, Random Forest Repressor, Tree Structured Tree Parzen Estimator. אופציות לפונקציית בחירה של ערכים הם Expected Improvement, Probability Improvement, GP Upper Confidence Bound.

### Gaussian Process

תחילה, לכל משתנה במרחב משייכים פונקציה כאשר אותן פונקציות מתפלגות באופן גאוסייאני. כלומר, מתחילים מ עבור אם נקודות כלשהן נחשבות דומות על ידי פונקציית הקרנל K אז הפונקציות f שלהן ייחשבו דומות אף הן. לאחר קבלת מספר תצפיות על ידי: כאשר ישנו חישוב על מטריצות שלא ממש זורם לי להעתיק לפה שקובע את ערכי הפונקציות הממוצעות והקרנל החדשות.

בגדול, מה שיוצא מזה זו פונקציה שבנקודות שהיא מכירה כבר יש וודאות מאד גבוהה (כי כבר יש לנו את הערך האמיתי של הפונקציה בנקודה) וככל שמתרחקים מהנקודות האלה הדיוק קטן כי מניחים התפלגות נורמלית (גאוסייאנית). נראה כמו בFigure 1.

|  |
| --- |
| The Intuitions Behind Bayesian Optimization with Gaussian Processes  Figure 1: דוגמה ל GP |

### Tree Parzen Estimator (TPE)

משתמשים bayes rule כדי לשערך את הפונקציה: . נבטא כל חלק: כלומר נגדיר פונקציית התפלגות אחת עבור ערכים ששימוש בהם בפונקציית המטרה נתן תוצאה טובה מערך סף מסוים ואחרת עבור האחרות (שנתנו תוצאה לא מספיק טובה ביחס לערך הסף). עם הגיון/מתמטיקה, אפשר להראות שנרצה לקחת דוגמאות רק מההתפלגות של הערכים "הטובים" (במקרה שלנו, ). בכל איטרציה לוקחים ערך מההתפלגות של המוצלחים ומקווים שגם היא תהיה מוצלחת. בכל מקרה, מעדכנים את ההתפלגויות בהתאם. נראה כמו בFigure 2.

|  |
| --- |
| Towards automating machine learning: benchmarking tools for hyperpara…  Figure 2: דוגמה ל TPE |

### Expected Improvement

נוסחת העל: . השיפור הצפוי מסט ערכים כלשהו יהיה ההפרש בין הערך של פונקציית המטרה עבור ערכים אלו לעומת הערך שניתן עבור סט הערכים הכי טוב שמצאנו עד כה. במידה וערך זה שלילי נסמן שהשיפור הצפוי הוא 0. לכל שיטה לבניית surrogate יש ערך שונה ל EI ולא אפרט עליהם פה.

## מדגים

לשם ההדגמה, עבדתי עם MNIST dataset מ sklearn ועשיתי שימוש ב 250 הפיצ'רים הראשונים בטבלה (על מנת לקר זמני ריצה). השוויתי בין Bayesian, Grid, Random. המודל הינו random forest regressor מ sklearn. בדיקת טיב האלגוריתם התבצעה בעזרת MSE ו 2FCV. מרחב החיפוש שהשתמשתי בו מתואר בטבלה 1. את random, Bayesian הגבלתי ל 25 איטרציות.

|  |  |  |
| --- | --- | --- |
| משתנה | מינימום | מקסימום |
| n\_estimators | 50 | 500 |
| max\_depth | 2 | 50 |
| min\_samples\_split | 2 | 9 |

טבלה 1: מרחב החיפוש של הערכים היפר-פרמטרים

מבחינת זמנים, את Grid עצרתי כי לקח לו יותר מידי זמן. Random הצליח לסיים לרוץ לאחר 11:51 דקות ו Bayesian סיים לרוץ לאחר 10:19 דקות. הדבר לא ממשמסתדר עם העובדה ש Bayesian משקיע יותר זמן בלבחור את האיטרציה הבאה שלו אבל אניח שאם הייתי מריץ את הניסוי הזה מספר פעמים אז הזמן הממוצע של random היה גדול מזה של Bayesian. מבחינת תוצאות: שתי השיטות הגיעו לתוצאה דומה (דיוק של 0.74) כאשר Bayes הגיע לתוצאה מעט טובה יותר. אם נסתכל על התוצאות לאורך האיטרציות (Figure 3) ניתן לראות מעין התכנסות עבור bayes אך לא עבור random.

|  |
| --- |
| Figure 3: דיוק כתלות באיטרציה |