

Átfutási idő előrejelzése gyártórendszerekben gépi tanuló algoritmusok segítségével

Lead time prediction in manufacturing systems with machine learning algorithms

SZALLER Ádám

Tudományos segédmunkatárs, Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutatóintézete (MTA SZTAKI), Mérnöki és Üzleti intelligencia Kutatólaboratórium (H-1111 Budapest, Kende u. 13-17., Tel.: +3612796262, Fax: +3614667503, e-mail: szaller.adam@sztaki.mta.hu, www.sztaki.hu)
PhD hallgató, Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar, Gyártástudomány- és technológia Tanszék (H-1111 Budapest, Műegyetem rkp. 3. T épület 4. em., Tel.: +3614632515, Fax: +3614633176, e-mail: szaller@manuf.bme.hu, manuf.bme.hu)

A gyártási átfutási idő az egyik legfontosabb teljesítménymutató a termelő vállalatok számára. A termék gyártásának kezdete és annak elkészülte közötti időtartam minél pontosabb becslésére különböző matematikai módszerek állnak rendelkezésre. A dolgozatban egy minta gyártórendszernek megfelelő működésű szimulációs modellen kerül tesztelésre többféle gépi tanuló algoritmus (lineáris regresszió, döntési fa alapú megoldások, neurális hálók) ezen probléma megoldására.

Manufacturing lead time is one of the most important key performance indicators for producing companies. For estimating the time interval between the starting and ending time of a product's manufacturing process, several mathematical methods are available. In the paper, different machine learning algorithms – linear regression, decision tree-based approaches, neural networks – are tested on a simulation model, which corresponds to the behaviour of a sample production system.

átfutási idő, gépi tanuló algoritmus, regresszió, szimuláció, gyártórendszer

1. BEVEZETÉS

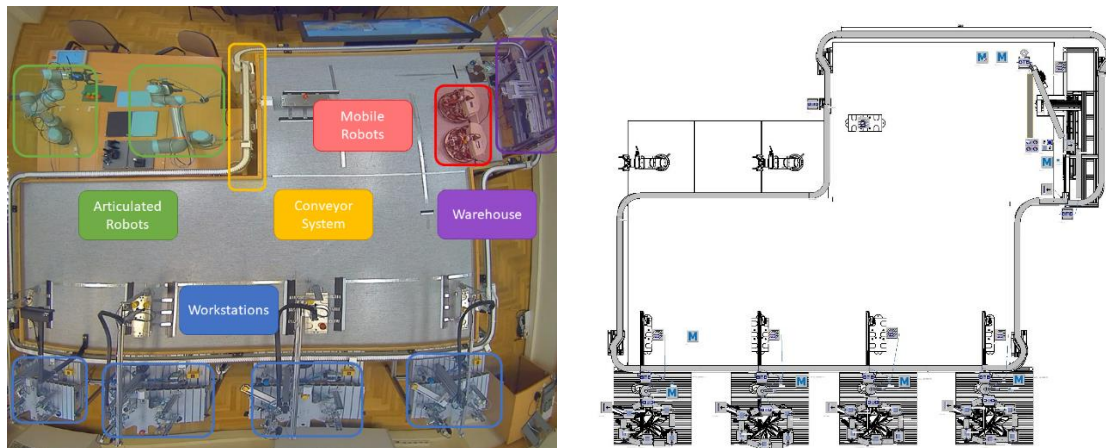
A gyártási átfutási idő az egyik legfontosabb teljesítménymutató azon vállalatok számára, akik a vevői igényeket (többek között) a megrendelés és annak elkészülte közötti időtartam minimalizálásával igyekeznek minél jobban kiszolgálni. A szállítási határidők szempontjából lényeges, hogy ezt az időt minél kisebb hibával képesek legyenek megbecsülni, előrejelezni. A dolgozatban a Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutatóintézetének SmartFactory mintagyártórendszerén különböző gépi tanuló algoritmusok kerülnek tesztelésre a rendszerben megmunkált munkadarabok átfutási idejének becslése céljából.

Az algoritmusok tanításához szükséges nagy mennyiségű adat könnyebb előállítására érdekében a gyártórendszer Tecnomatix Plant Simulation szimulációs szoftverrel elkészített modelljének létrehozása után az algoritmusok bemenetül szolgáló adatok nem a valós fizikai rendszerből, hanem az annak megfelelő szimulációs modelltől generálódnak. A termékek a gyártórendszer különböző pontjain logolt időadatai alapján több különböző gépi tanuló algoritmus kerül bemutatásra, tesztelésre és összehasonlításra a becslés pontossága szempontjából.

2. MINTA GYÁRTÓRENDSZER – SMARTFACTORY

Az MTA-SZTAKI Mérnöki és Üzleti Intelligencia Kutatólaboratóriuma felismerte a szükségességét egy, a mérnöki oktatásban is használható bemutatórendszer létrehozásának. Ezen elgondolás eredménye a SmartFactory minta gyártórendszer. A rendszer több fajta kísérletre használható: átlátható méretben vizsgálhatóak rajta tipikus gyártási és logisztikai folyamatok,

számítógépes reprezentációval összekötött vagy kiegészített szimulációk futtathatóak rajta, valamint komplex ütemezési és logisztikai feladatok is megvalósíthatóak a segítségével [3]. A gyártórendszer a következő részekből épül fel (feliratozva az 1. ábra bal oldalán): a gyártórendszer bemenete, és egyben a kész termékek kimenete a magasraktár (warehouse), innen a négy munkaállomás (workstations) valamelyikére robotok (mobile robots) vagy a szállítószalag (conveyor system) segítségével jutnak el a munkadarabok. A munkaállomások előtt átmeneti tárolók helyezkednek el, a mobil robotok ide helyezik le a munkadarabokat, amiket innen, vagy a szállítószalagról az állomásokhoz rendelt pneumatikus portális megfogók továbbítanak a munkaállomás öt fészkes forgóasztala felé. A munkaállomások összesen ötféle megmunkálási műveletet végezhetnek, ezek ciklusidejei minden esetben 1 perces várható értékű, 10 másodperces szórású normális eloszlást követnek.



1. ábra

SmartFactory minta gyártórendszer [2] és szimulációs modellje

Mivel az átfutási idő becsléséhez megerősítéses tanuláson alapuló algoritmusokat alkalmazunk, ezért szükséges a megfelelő számú (tízezres nagyságrendű) minta használata ezek tanításához és teszteléséhez. Ekkora mennyiségű adatot egy valós, fizikai rendszerrel nagyon időigényes lenne előállítani – a rendszernek megfelelő működésű szimulációs modell segítségével azonban másodpercek alatt elvégezhető ez a feladat. A modell (1. ábra jobb oldala) a Tecnomatix Plant Simulation szoftver segítségével készült, közel 20 másodperc alatt képes kb. 72 ezer termék gyártásának szimulációjára, és közben a releváns adatok elmentésére is (ez a gyártási folyamat a valóságban több, mint 118 napig tartana). Egy termék átfutási ideje kb. 6 perc és 3 óra között változik (természetesen akkor nagyobb ez az érték, ha sok termék van a rendszerben feltorlódva).

3. ÁTFUTÁSI IDŐ ELŐREJELZÉSE

Az átfutási idő előrejelzésénél a következőkben pár mondatban bemutatott, Python programozási nyelven előre megírt gépi tanuló algoritmusok kerülnek összehasonlításra [1]:

Lineáris regresszió (linear regression): a kimeneti változó – jelen esetben az átfutási idő – a jellemzők súlyozott összege, az algoritmus futtatásakor a cél a súlyok értékének meghatározása. Az algoritmus a tanítási folyamat során a valós értéktől való átlagos négyzetes eltérést minimalizálja.

Döntési fák módszere (decision trees): döntési fák esetén megkülönböztethető diszkrét és folytonos eset, a bemeneti jellemzőket és a kimeneti értéket tekintve is. Diszkrét kimeneti érték esetén klasszifikációt (osztályozás) végzünk, folytonos esetben – mint amilyen az átfutási idők becslése is – pedig regressziót. A döntési fák gyakorlatilag az ún. fa gráfstruktúrára épülnek, ahol a csúcsok kérdéseknek, az élek pedig a kérdésekre adott válaszoknak feleltethetők meg. A cél, hogy minél kevesebb kérdés feltételével és megválaszolásával eljussunk az eredményhez. A fa tanításán ezen kérdéseknek olyan sorrendbe állítását értjük, amellyel a leggyorsabban jutunk a válaszhoz. Folytonos változók esetén a kérdések tipikusan egyenlőtlenességekre vonatkoznak, pl. egy adott jellemző értéke kisebb-e egy bizonyos másik értéknél.

Véletlen erdő regresszió (random forest regression): A döntési fák módszerének továbbfejlesztése. A döntési fák egyik legnagyobb problémája, hogy túltanulják (overfitting) a tanulóhalmazt – vagyis az algoritmus túlságosan jól igazodik azokhoz az adatokhoz, amiken tanítva lett, a tesztalmazon viszont

nem produkál ilyen jó eredményeket. A véletlen erdők módszerénél véletlen alapon több olyan különböző döntési fa kerül előállításra, amely ugyan túltanulja a tanulóhalmazt, de ezek eredményeinek átlagolásával a módszer olyan megoldást hoz létre, amely már nem illeszkedik ennyire szorosan a tanulóhalmazhoz, emiatt a tesztalmazon sokkal jobb eredményeket produkál.

Gradiens fokozó fák módszere (gradient boosting trees): Hasonlóan működik a véletlen erdőhöz, de azzal ellentétben a többféle döntési fa nem véletlen alapon kerül előállításra, hanem minden előállított fa az előző eredményét igyekszik javítani, így elkerülve a túltanulást.

Neurális háló (neural network): A neurális hálóval való becslést úgy is felfoghatjuk, mint a lineáris regresszió általánosítását. Az utóbbi módszernél korábban említett súlyozott összeget gyakorlatilag sokszor egymás után kiszámítjuk (rejtett rétegek). A folyamatot itt egy irányított gráffal lehet a legjobban szemléltetni: ennek csomópontjaiban a jellemzők, élein a súlyok szerepelnek, és az élek a kimeneti érték felé mutatnak. Tulajdonképpen az előbb ismertetett gráfot sokszor egymás után kapcsolva épül fel az algoritmus, annyi kiegészítéssel, hogy minden réteg után egy nemlineáris függvény kerül alkalmazásra az eredményeken, a becslés pontosságának javítása érdekében.

Az említett algoritmusok a mintát tanító és tesztalmazra bontják: az algoritmus a minta értékeinek 75%-án tanul, a fennmaradó 25%-on pedig teszteli a megtanult rendszert. Talán a legfontosabb kérdés, hogy a becslést milyen jellemzők alapján kívánjuk elvégezni – ezek kiválasztásához szükséges ismerni a gyártórendszer működését. Mivel a munkaállomásokhoz a termékek kétféle úton is eljuthatnak (robot, szállítószalag), és négy darab munkaállomás van, ezért összesen nyolcféle útvonalat definiálhatunk a SmartFactory-ban. Ennek megfelelően a következő három jellemzőt vesszük figyelembe:

- az adott útvonalon már a magasraktárba visszatért előző termék átfutási ideje,
- a termék rendszerbe való bekerülésekor az adott útvonalon lévő termékek száma,
- a termék rendszerbe való bekerülésekor a teljes rendszerben lévő termékek száma.

A jellemzők kiválasztása kísérletezéssel történt: például kevésbé pontos eredményt kapnánk, ha nem csak az adott útvonalon végighaladt előző egy termék átfutási idejét vennénk figyelembe, hanem több, korábbi termék idejének átlagát. Mivel nyolcféle útvonalat különböztetünk meg, ezért nyolc különböző esetben alkalmazzuk a fentebb említett algoritmusokat. Az eredményeket az 1. táblázat foglalja össze: az első oszlopban lévő címkék mutatják, hogy melyik állomáson kerültek megmunkálásra a termékek, és hogy milyen módon kerültek oda. Például, a *WS1_szalag* azokra a termékekre utal, amelyek az 1. munkaállomáson kerülnek megmunkálásra, és oda a szállítószalagon jutottak el. A táblázat tartalmazza az adott szállítási módhoz tartozó átfutási idők átlagát és szórását (szalag/robot átlag és szórás), mind a nyolc útvonalhoz külön az ötféle algoritmus által szolgáltatott értékek ugyanezen mutatószámait (útvonal átlag és szórás), illetve az algoritmusoknál egyesével is kiszámításra kerültek a nyolcféle útvonalra összesítve vonatkoztatott értékek (algoritmus átlag és szórás). A táblázat egyes celláinak színekjelölése a következőt jelenti: minél erősebb pirossal van jelölve egy cella, annál alacsonyabb értéket tartalmaz, illetve minél erősebb zölddel, annál magasabb érték szerepel benne.

A gépi tanuló algoritmusok összehasonlítása

1. táblázat

	lineáris regresszió	döntési fák	véletlen erdő	gradiens fokozó fák	neurális háló	szalag /robot átlag	szalag /robot szórás	útvonal átlag	útvonal szórás
<i>WS1_szalag</i>	72%	78%	75%	80%	75%	79,5%	5,4%	76,0%	3,1%
<i>WS2_szalag</i>	72%	78%	77%	81%	76%			76,8%	3,3%
<i>WS3_szalag</i>	74%	80%	82%	85%	74%			79,0%	4,9%
<i>WS4_szalag</i>	80%	86%	89%	90%	86%			86,2%	3,9%
<i>WS1_robot</i>	70%	75%	75%	79%	72%	73,8%	5,7%	74,2%	3,4%
<i>WS2_robot</i>	69%	71%	72%	77%	70%			71,8%	3,1%
<i>WS3_robot</i>	67%	70%	69%	72%	67%			69,0%	2,1%
<i>WS4_robot</i>	73%	85%	83%	87%	73%			80,2%	6,7%
<i>algoritmus átlag</i>	72,1%	77,9%	77,8%	81,4%	74,1%				
<i>algoritmus szórás</i>	3,9%	5,8%	6,5%	5,8%	5,6%				

Az 1. táblázat alapján a következő megfigyeléseket tehetjük:

- Az algoritmusok összességében 67% és 90% közötti pontossággal becsülik meg egy termék átfutási idejét, ami a megmunkálási idők 10 másodperces – vagyis az 1 perchez képest 17%-os – szórását figyelembe véve elfogadható eredmény.
- A lineáris regresszió szolgáltatja a legnagyobb hibájú, a gradiens fokozó fák módszere pedig a legpontosabb eredményeket átlagban (kb. 9%-os a különbség a kettő között). Érdemes megfigyelni, hogy a *WS3_szalag* és a *WS4_robot* útvonalak esetében ezen két algoritmus pontossága közötti különbség 11%, illetve 14% – ezekben az esetekben a két becslési módszer eredménye közötti eltérés okozza a többenél magasabb, útvonalra vonatkozó szórásértéket is.
- Észrevehető, hogy a 4. munkaállomáson megmunkálásra kerülő termékek átfutási idejét sokkal pontosabban meg tudja becsülni az algoritmus – akár a szállítószalagon, akár a robotokkal érkező termékekről van szó.
- A legpontatlanabb becslést az összes algoritmus *WS3_robot* útvonalra, vagyis a 3. munkaállomásra robotok segítségével továbbított termékekkel kapcsolatban tudja tenni – egyes algoritmusoknál a más útvonalakra vonatkozó becslésekhez képest akár 18-19%-os eltérés is megfigyelhető.
- Különbség van a szállítási módok becsült átfutási időre való hatásában is: átlagosan 6%-kal pontosabban meg lehet becsülni ezt az értéket a szállítószalaggal továbbított termékeknél, mint a robottal mozgatottaknál.

Látható, hogy az algoritmusok megfelelően működnek az átfutási idő becslése szempontjából, a gradiens fokozó fák módszere ezen probléma megoldására kiemelkedik a többi közül – azonban az is megállapítható, hogy a gyártórendszer sajátosságaiból adódóan nem minden útvonalra lehet azonos pontossággal megbecsülni ezt az értéket.

4. ÖSSZEFOGLALÁS, FEJLESZTÉSI LEHETŐSÉGEK

A dolgozatban különböző gépi tanuló algoritmusok segítségével az MTA SZTAKI SmartFactory minta gyártórendszerében megmunkált termékek átfutási ideje került becslésre, előrejelzésre. Az algoritmusok tanításához és teszteléséhez szükséges adatmennyiség a gyártórendszernek megfelelően működő, erre a célra Tecnomatix Plant Simulation szoftverrel készített szimulációs modellből generálható, mivel a valóságban ekkora minta létrehozása nagyon időigényes lenne. A gépi tanuló algoritmusok Python programozási nyelv alatt futnak, és a gyártórendszerben logikailag elkülöníthető (de közös részekkel is rendelkező) nyolc útvonalon mozgó termékek átfutási idejét külön becsülik meg.

Összességében megállapítható, hogy az algoritmusok (különösen a gradiens fokozó fák) a megmunkálási időket terhelő szórás nagysága ellenére elfogadhatóan becsülik a termékek átfutási idejét, segítségével a megfelelő jellemzők értékének ismeretében előre jelezhető egy termék várható elkészülésének időpontja. Természetesen (ahol erre lehetőség van), a használt módszerek finomhangolásával valószínűleg tovább javíthatók lennének az eredmények, például a döntési fa alapú megoldásoknál a fák mélységének szabályozásával, vagy a neurális hálók rejtett rétegszámának állításával. További vizsgálatokat lehet tenni a megmunkálási idők kisebb szórása esetén (ekkor várhatóan pontosabbak lesznek a becslések), de a váratlan gépleállások figyelembevétele a modellben is egy fejlesztési lehetőség. Más gépi tanuló eljárások, például a tartóvektor-gépek (support vector machines) kipróbálása, és a jelenlegi módszerekkel való összehasonlítása is kutatási lehetőségeket rejt magában.

5. IRODALOMJEGYZÉK

- [1] Andreas C. Müller, Sarah Guido: Introduction to Machine Learning with Python, O'Reilly, USA, 2016.
- [2] Beregi Richárd, Szaller Ádám, Kádár Botond: Synergy of multi-modelling for process control. INCOM 16th IFAC Symposium on Information Control Problems in Manufacturing, Bergamo, Olaszország, 2018 (bírálat alatt)
- [3] Gépgyártástechnológia laborgyakorlati segédanyag, Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar, Gyártástudomány és –technológia Tanszék