Denis Belomestny. Foto: Vladimir Unkovic

Entscheidungen begleiten uns im Alltag. Wenn Sie zum Beispiel eine Ware im Laden kaufen, entscheiden Sie sich für eine Marke. Wenn Sie das Haus verlassen, entscheiden Sie darüber, was Sie anziehen sollen. Oft treffen die Leute Entscheidungen anhand der unvollständigen oder unsicheren Information (zum Beispiel anhand der Information auf Etiketten oder der Wettervorhersage). Was ist eine optimale Entscheidung unter diesen Umständen? Wie soll man handeln um diese optimale Entscheidung zu treffen? All diese Fragen beantwortet die mathematische Entscheidungstheorie.

Entscheidungen mathematisch treffen

Approximative Dynamische Programmierung mit Tiefen Neuronalen Netzwerken

Von Denis Belomestny

Eine wichtige Klasse von Entscheidungsproblemen sind solche, die von der Zeit abhängig (dynamisch) und irreversibel sind. Solche Problemen sind ganz natürlich, da jede reale Entscheidung die zeitbedingten Änderungen berücksichtigen soll, an die sich der Entscheidungsträger anpasst. Diese Art von Entscheidungsproblemen können extrem komplex sein: Sie können Veränderungen in den Präferenzen, Technologien und Ressourcen beinhalten.

Um ein dynamisches Entscheidungsproblem zu lösen, müssen wir Annahmen treffen, wie der Entscheidungsträger alternative Strategien bewertet. Die Standardannahme ist, dass er den erwarteten Nutzen maximiert. Um eine beste Entscheidung für heute zu berechnen, muss man die optimale Entscheidung für die Zukunft, das heißt für alle mögliche zukünftige Szenarien, kennen. Daher soll die Suche nach der optimalen Entscheidung nicht chronologisch, sondern in der umgekehrten Reihenfolge, das heißt rückwärts in der Zeit erfolgen, weil das gegenwärtige Optimum vom zukünftigen Optimum abhängig ist. Diese fundamentale Überlegung bildet eine Grundlage für das Prinzip der dynamischen Programmierung, das von Richard Bellman im Jahr 1950 entwickelt wurde. Richard Bellman gilt als Pionier der dynamischen Programmierung. Er erkannte die gemeinsame Struktur von dynamischen Entscheidungsproblemen und zeigte, wie die Rückwärtsinduktion angewendet werden kann, um eine große Klasse von dynamischen Entscheidungsproblemen unter Unsicherheit zu lösen.

Es gibt zwei Schlüsselvariablen in jedem dynamischen Programmierungsproblem: eine Zustandsvariable und eine Kontrollvariable. Zum Beispiel bei dem Problem der Verwaltung von Wasserreservoirs ist das Wasserniveau im Reservoir eine Zustandsvariable und die abgelassene Wassermenge eine Kontrollvariable. Bei Investitionsproblemen wird der Zustand durch das Depot von Aktien bestimmt, und die Entscheidungen über Kauf/Verkauf von Aktien werden durch entsprechende Kontrollvariable (Zeit des Verkaufs, Anzahl von Aktien) quantifiziert. Eine zusätzliche Komplexität kann durch Unsicherheit oder fehlende Informationen entstehen. Die Komplexität der dynamischen Programmierung ergibt sich in der ersten Linie aus dem exponentiellen Wachstum der Anzahl möglicher Szenarien, wenn die Anzahl möglicher Werte für die Zustandsvariablen, Entscheidungsvariablen und oder die Anzahl von Zeitperioden ansteigt. Dieser exponentielle Wachstum wird zu einer großen Herausforderung in Anwendungen der dynamischen Programmierung. In diesem Beitrag werde ich verschiedene Strategien zum Umgang mit diesem Problem beschreiben. Zunächst möchte ich zwei wichtige Klassen der dynamischen Entscheidungsprobleme vorstellen.

Optimale Stoppprobleme

Eine wichtige Klasse von dynamischen Entscheidungsproblemen sind die sogenannten Optimalen Stoppprobleme.

Historisch gesehen entstammen die Optimalen Stoppprobleme aus dem Wunsch, die zufälligen Sequenzen von Ergebnissen in Glücksspielen zu kontrollieren. Will man den optimalen Zeitpunkt zum Abbrechen eines Spiels wissen, muss man ein Optimales Stoppproblem lösen. Eine allgemeine mathematische Theorie, die einen Einblick in solche Probleme geben könnte, wurde erst im letzten Jahrzehnt ernsthaft untersucht. Als Beispiel beschreiben wir zunächst das sogenannte Sekretärinnenproblem. Dieses Problem hat eine lange Geschichte und ist zuerst als Thema für die Diskussion in einem „Scientific American"-Artikel von 1960 erschienen. Eine Lösung dieses Problems wurde dort intuitiv beschrieben. Dynkin hat im Jahr 1963 bewiesen, dass diese Lösung optimal ist. Das Problem betrifft einen Arbeitgeber, der eine Sekretärin aus einer Gruppe von jungen Frauen anstellen muss. Bei jedem Interview kann er nur erkennen, wie sich die befragte junge Frau mit denen vergleicht, die er zuvor gesehen hat. Direkt nach dem Interview muss der Arbeitgeber sich dafür entscheiden, die junge Frau einzustellen oder abzulehnen, ohne dass es eine Rücknahmemöglichkeit für diese Entscheidung gibt. Sein Ziel ist es, durch eine kluge Auswahlpolitik die Wahrscheinlichkeit zu maximieren, dass er das Beste aus der Menge von Mädchen wählt. Bei Stoppproblemen entscheidet man sich immer zwischen „Stoppen" und „Weitermachen". Die Antwort auf die Frage, wann man am besten stoppen soll, hängt unter anderem auch von der erwarteten Nutzenfunktion des Entscheidungsträgers ab.

Wo kann man Stoppproblemen heute noch begegnen? Einer der aktivsten Orte, an denen Optimale Stoppprobleme heutzutage zur Anwendung kommen, ist die Wall Street. Zur Illustration betrachten wir das Problem der Bestimmung, wann eine Amerikanische Option beansprucht werden soll. Diese Option gibt dem Käufer das Recht, die Aktie für einen bestimmten Preis K bis zu einem festen Zeitpunkt T zu kaufen. Die genaue Zeit des Kaufs kann von dem Käufer bestimmt werden. Zwei Fragen ergeben sich: Wann sollte die Option beansprucht werden? Was ist ihr Wert? Um diese Fragen zu beantworten, ist ein Aktienpreismodell erforderlich. Dann kann man mit Hilfe der dynamischen Programmierung den Wert der Option sowie eine optimale Ausübungsregel berechnen. Im Grunde genommen gibt es zwei Alternativen an jedem Handelstag. Man kann die „Option beanspruchen", das heißt die Aktie zum Ausübungspreis K kaufen und sofort wieder verkaufen zum gegenwärtigen Preis S, was einen Gewinn S-K ergibt. Alternativ kann man die „Option behalten" und bei dem maximalen erwarteten Gewinn für die Restzeit bleiben. Gemäß dem Prinzip der dynamischen Programmierung soll man diese zwei Alternativen vergleichen und eine wählen, die zu höheren erwarteten Gewinn führt.

Stochastische Kontroll- oder Steuerungsprobleme

Eine weitere große Klasse von dynamischen Entscheidungsproblemen in den Wirtschafts- und Naturwissenschaften lässt sich mathematisch als stochastisches Kontroll- oder Steuerungsproblem darstellen. Hier sind die Steuerung und Optimierung etwa von Portfolios oder derivativen Produkten unter Transaktionskosten ein typisches praktisches Problem. Innerhalb eines stochastischen Systems kann die Entwicklung eines Prozesses durch eigene Einwirkung mit bestimmt werden, und eine Handlungsstrategie soll so gewählt werden, dass über einen vorher festgelegten Zeitraum ein optimales Nutzen-Kosten-Verhältnis erreicht wird. In einem Kontrollproblem stellt sich eine intertemporale Nutzen-Kosten-Funktion als Zielfunktional dar, das es zu maximieren gilt, und Kontrollvariablen sind Entscheidungen. Die Kontrolle oder Steuerung wird selbst zu einer Funktion über Zeit und Zustand, die für jeden möglichen Zustand des Systems eine Handlung vorgibt. Als Beispiel könnte man sich das Problem der Energiespeicherung anschauen. Im Gegensatz zu rein finanziellen Verpflichtungen wie Aktien und Anleihen kann Energie (Strom, Erdgas, Öl etc.) physisch gespeichert werden. Während solche Energieressourcen wie Gas und Öl direkt gespeichert werden können, kann die Speicherung von elektrischer Energie durch Umwandlung in eine andere Energieform zur Speicherung erfolgen. Eine Batterie verwendet beispielsweise einen reversiblen elektrochemischen Prozess. Die Speicherung ermöglicht eine zeitliche Übertragung der Energie und ermöglicht die Nutzung der fluktuierenden Markt-Energiepreise. Das Grundprinzip lautet: „niedrig kaufen" und „hoch verkaufen", so dass der realisierte Gewinn die Zwischenspeicher- und Betriebskosten deckt. Die Gewinnchancen hängen von der Dynamik des Preisprozesses und dessen Prognostizierbarkeit ab. Die Preisentwicklung kann entweder systemisch oder spekulativ sein. So weist beispielsweise der Erdgasmarkt eine starke Saisonalität auf, da die Hauptverbrauchergruppe Haushalte sind, die Gas für die Heizung im Winter verwenden. Üblicherweise gehören Lagereinrichtungen zu den Hauptakteuren der jeweiligen Branchen, die über das enorme Kapital verfügen, das typischerweise für den Bau und die Wartung dieser Anlagen benötigt wird. Mit den liberalisierten Märkten haben jedoch alle Beteiligten heute die Möglichkeit, einen Speicher zu mieten, um auf Preise zu spekulieren. Daher wird es notwendig, den finanziellen extrinsischen Wert des Speichers zu berechnen. Man kann nämlich fragen, wie viele man davon gewinnen kann, um die Kontrolle über ein Lager für eine bestimmte Anzahl von Jahren zu erlangen. Eine weitere verwandte Frage ist, wie man die optimalen Strategien für den Kauf, die Lagerung und den Verkauf von Energie in der Zeit beschreibt. Es gibt keine einfache Antwort auf diese Fragen, da der Lagerhalter viele betriebliche und technische Einschränkungen wie Lagerkapazitätsgrenzen oder Lieferkosten hat, die wiederum von der gewählten Lagerungsstrategie abhängen. Um die Abhängigkeit zwischen der zeitlichen Wahlmöglichkeit bei der Auswahl der Kauf- und Verkaufszeiten und den Lagerbestandsbeschränkungen richtig zu berücksichtigen, muss man das komplexe stochastische Kontrollproblem betrachten und das Prinzip der dynamischen Programmierung anwenden, um dieses Problem zu lösen.

Approximative dynamische Programmierung

Die vorangegangenen Abschnitte zeigen, dass die dynamische Programmierung ein mächtiges Werkzeug ist, das es uns ermöglicht hat, eine Vielzahl von ökonomischen und physikalischen Modellen zu formulieren und zu lösen, die sequenzielle Entscheidungsfindung unter Unsicherheit beinhalten – zumindest „in der Theorie". Leider sind die Fälle, in denen die dynamische Programmierung zu analytischen Lösungen in geschlossener Form führt, selten und oft eher fragil in dem Sinne, dass kleine Änderungen in der Formulierung eines Problems die Fähigkeit zerstören können, eine analytische Lösung zu erhalten. Obwohl die meisten Probleme keine analytischen Lösungen haben, garantieren die mathematischen Sätze die Existenz von Lösungen, und diese Lösungen können mit numerischen Methoden berechnet (oder approximiert) werden. In den letzten Jahrzehnten haben schnellere Computer und bessere numerische Methoden die dynamische Programmierung zu einem Werkzeug von erheblichem praktischen Wert gemacht, indem sie die Palette der zu lösenden Probleme erheblich erweitert haben. Es gibt jedoch noch immer viele schwierige Herausforderungen, die uns daran hindern, Modelle zu entwickeln und zu lösen, die so detailliert und realistisch sind, wie wir es gerne hätten, ein Problem, das in empirischen Anwendungen besonders akut ist. Die Hauptherausforderung ist, was Bellman und Dreyfus (1962) den Fluch der Dimensionalität genannt haben. Dabei handelt es sich um eine exponentielle Abhängigkeit der Komplexität von dem Zeithorizont und der Anzahl von Zustandsvariablen. Leider muss man feststellen, dass es eindeutig nicht möglich ist, eine Rückwärtsinduktion (Dynamische Programmierung) für jede mögliche Geschichte (Szenarios) von Zustandsvariablen auszuführen, da es unendlich viele (tatsächlich ein Kontinuum) von solchen Szenarien gibt. In diesen Fällen ist es notwendig, die Wertfunktion zu interpolieren, deren Werte nur explizit an einer endlichen Anzahl von Punkten (Gitter) im Zustandsraum berechnet werden. Gitter können zufällige Gitter sein, die durch zufällige Generierung der Zustandsvariable aus einer Wahrscheinlichkeitsverteilung entstehen. Der Grund, warum man ein Gitter mit zufälligen Punkten vorziehen kann, ist, den Fluch der Dimensionalität zu vermeiden. Sobald ein bestimmtes Gitter ausgewählt ist, wird die Rückwärtsinduktion in der normalen Weise ausgeführt, wie es in einem Problem mit endlicher Anzahl von Zuständen durchgeführt werden würde. Allerdings, selbst wenn die tatsächliche Rückwärtsinduktion nur auf dem gewählten Gitter ausgeführt werden kann, werden wir immer noch numerische Integration zur Berechnung des bedingten Erwartungswertes benötigen. Diese Integration kann die Werte der Wertfunktion auch außerhalb des Gitters gebrauchen. Aus diesem Grund ist normalerweise eine Form der Interpolation (oder in einigen Fällen eine Extrapolation) erforderlich. Viele Methoden der Interpolation können als gewichtete Summen von Werten der Wertfunktion auf dem Gitter dargestellt werden. Ein alternativer Ansatz kann als Kurvenanpassung beschrieben werden. Anstatt zu versuchen, die berechneten Werte der Wertfunktion an den Gitterpunkten zu interpolieren, behandelt dieser Ansatz diese Werte als einen statistischen Datensatz und versucht Parameter in einer flexiblen parametrischen Approximation für die Wertfunktion mit Hilfe der nichtlinearen Regression zu schätzen. Besonders geeignet dafür sind die sogenannten Neuronalen Netze. Ursprünglich wurden sie entwickelt, um die neurobiologischen Prozesse innerhalb des Nervensystems bei Tieren und Menschen besser zu beschreiben. Seit Ende der 1980er Jahre entwickelte sich neben dieser neurobiologisch orientierten Forschungsrichtung eine eigene nur auf statistische Problemstellungen bezogene anwendungsorientierte Richtung. Das Multi-Layer-Perceptron (MLP) ist das bekannteste struktur-abbildende künstliche Neuronale Netzwerk. Es ist eine Weiterentwicklung der Perceptrons, einem zweischichtigen, trainierbaren neuronalen Netz, dass von dem Psychologen Rosenblatt 1958 entwickelt wurde. Als überwachtes Lernverfahren bietet MLP die Möglichkeit, funktionale Zusammenhänge zwischen einer oder mehreren abhängigen und unabhängigen Variablen zu modellieren. Eine der bemerkenswertesten Fakten über neuronale Netze ist, dass sie überhaupt jede Funktion berechnen können. Nehmen Sie an, jemand gibt Ihnen eine komplizierte Funktion f(x). Dann, unabhängig von der Funktion f, gibt es garantiert ein neuronales Netzwerk das für jeden möglichen Argument x, f(x) als Ausgangswert liefert. Dieses Ergebnis gilt auch dann, wenn die Funktion viele Argumente hat und vektorwertig ist. Dieses Ergebnis sagt uns, dass neuronale Netze eine Art der Universalität besitzen. Egal, welche Funktion wir berechnen wollen, wir wissen, dass es ein neuronales Netzwerk gibt, das diese Aufgabe erfüllen kann. Dieser Universalitätssatz gilt auch dann, wenn das Netzwerk nur eine einzige Schicht zwischen den Eingangs- und Ausgangsneuronen hat – eine sogenannte einzelne versteckte Schicht. So können selbst sehr einfache Netzwerkarchitekturen extrem leistungsfähig sein. Angesichts dieser Tatsache könnten Sie sich fragen, warum wir uns jemals für tiefe Netzwerke interessieren würden, das heißt für Netzwerke mit vielen versteckten Schichten. Können wir diese Netzwerke nicht einfach durch flache, einzelne Netze mit versteckter Schicht ersetzen?

Während dies prinzipiell möglich ist, gibt es gute praktische Gründe, tiefe Netzwerke für die Approximation der Wertfunktion in der dynamischen Programmierung zu nutzen. Ein wichtiger Grund dafür ist eine geschachtelte Struktur der Rückwärtsinduktion, die einer verschachtelten Zusammensetzung von Funktionen in tiefen neuronalen Netzwerken entspricht. Man kann sich an dieser Stelle fragen, ob tiefe neuronale Netzwerke überhaupt gelernt werden können. In der Tat, wurde seit 2006 eine Reihe von effizienten Techniken entwickelt, die das Lernen in tiefen neuronalen Netzen ermöglichen. Diese Deep-Learning-Techniken basieren typischerweise auf stochastischen Gradientenmethoden. Lassen Sie uns jetzt zu der approximativen dynamischen Programmierung zurückkehren. Wenn eine Methode zur Interpolation/Extrapolation der Wertfunktion bestimmt wurde, muss eine zweite Wahl über die geeignete Methode für die numerische Integration getroffen werden, um den bedingten Erwartungswert der Wertfunktion anzunähern. Hier gibt es im Grunde genommen zwei Möglichkeiten:

1) deterministische Quadraturregeln oder
2) Monte-Carlo-Methoden.

Die deterministischen Quadraturmethoden sind sehr genau, jedoch werden sie bei multivariaten Integrationsproblemen unhandlich, wenn Tensorprodukte eindimensionaler Quadratur verwendet sind. Für jede Art von deterministischen Quadraturmethoden kann gezeigt werden, dass sie dem Fluch der Dimensionalität in Bezug auf die Berechnungskomplexität im ungünstigsten Fall unterliegen. In der Tat, mit der Theorie der Rechenkomplexität kann man beweisen, dass jede deterministische Integrationsprozedur dem Fluch der Dimensionalität unterliegt, zumindest im Sinne eines Worst-Case-Maßes der Komplexität. Da die multivariate Integration ein „Teilproblem" ist, das gelöst werden muss, um dynamische Programmierung durchzuführen, sollte es auch nicht überraschend sein, dass auch die dynamische Programmierung dem gleichen Fluch unterliegt. Es gibt jedoch Beispiele von zufälligen Algorithmen, die den Fluch der Dimensionalität umgehen können. Monte-Carlo Integration ist ein klassisches Beispiel. In der Tat implizieren das Gesetz der großen Zahlen und der zentrale Grenzwertsatz, dass das Monte-Carlo-Integral mit der Rate eins durch Wurzel von N gegen den bedingten Erwartungswert konvergiert, wobei N die Anzahl von Monte-Carlo Ziehungen ist.

Der Monte-Carlo-Integration, gelingt es somit, den Fluch der Dimensionalität in der multivariaten Integration zu brechen. Die naive Anwendung der Monte-Carlo-Integration wird jedoch nicht notwendigerweise den Fluch der Dimensionalität des Problems der dynamischen Programmierung brechen. Der Grund ist, dass eine Form der gleichmäßigen (im Gegensatz zur punktweisen) Konvergenz der bedingten Erwartungswerten vorliegen muss um zu garantieren, dass die gesamte Rückwärtsinduktion zur wahren Lösung konvergiert, wenn die Anzahl der Monte-Carlo-Ziehungen, N, groß wird. Mit Hilfe der Theorie von empirischen Prozessen kann man zeigen, dass randomisierte Versionen der Rückwärtsinduktion den Fluch der Dimensionalität des dynamischen Entscheidungsproblems zu durchbrechen vermag.

Summary

Decision problems are part of our everyday life. For example, if we want to buy a product in the store, we choose a brand. Before leaving the house, we decide what to wear. Often, people make decisions based on incomplete or uncertain information (for example etiquette or the weather forecast). What is an optimal decision under these circumstances? How should we act in order to make this optimal decision? These questions are answered using mathematical decision theory. An important class of decision problems are those that are time-dependent (dynamic) and irreversible. Such problems are quite natural, as every real decision must take into account the time-related changes the decision-maker has to adjust to. These types of decision problems can be extremely complex, involving changes in preferences, technologies, and resources. To solve a dynamic decision problem, we must make assumptions about how the decision-maker evaluates alternative strategies. The standard assumption is that the decision-maker maximizes the expected benefit. Furthermore, in order to calculate the best decision for today, the decision-maker must make the optimal decision for the future too, i.e. for any future scenarios. The search for the optimal decision should therefore not be made chronologically, but in reverse order, i.e. backwards in time, because the present optimum depends on the future optimum. This fundamental consideration forms the basis of the principle of dynamic programming developed by R. Bellman. The complexity of dynamic programming results primarily from the exponential growth in the number of possible scenarios with the increase in the number of possible values for the state variables, decision variables and/or the number of time periods. This exponential growth becomes a major challenge in dynamic programming applications. In this article I describe different strategies for dealing with this problem.

Der Autor

Denis Belomestny studierte an der Moskauer Lomonosov-Universität Mathematik mit dem Schwerpunkt Statistik und Stochastische Prozesse. Nach seinem PhD-Abschluss im Jahr 2002 arbeitete er als Postdoc am Institut für Angewandte Mathematik der Universität Bonn. Anschließend war er unter anderem am Berliner Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS) tätig. Ab 2007 übernahm er mehrere Lehraufträge an der Wirtschaftswissenschaftlichen Fakultät der Berliner Humboldt-Universität. Am WIAS arbeitete Belomestny zudem als Berater für die Landesbank Berlin, Nordbank und WGZ Bank. Seine Arbeitsgebiete umfassen die Statistik von Stochastischen Prozessen, die einen starken Bezug zu Fragestellungen aus der Finanzmathematik und der Ökonometrie aufweisen. Seit einigen Jahren beschäftigt er sich auch verstärkt mit simulationsbasierten (Monte-Carlo) Algorithmen für Probleme des optimalen Stoppens und der optimalen Steuerung. Solche Algorithmen finden breite Anwendung in der Finanzmathematik, zum Beispiel bei der Bewertung von Amerikanischen Optionen. Im Februar 2011 hat Belomestny die Professur für Angewandte Stochastik an der Universität Duisburg-Essen übernommen.

DOI: 10.17185/duepublico/70317
URN: urn:nbn:de:hbz:464-20190731-105059-1

Erschienen in: UNIKATE 53 (2019), S. 73-78