Introduction to the Virtual Issue: Machine Learning in Political Science

Skyler J. Cranmer
Carter Phillips and Sue Henry Associate Professor of Political Science
The Ohio State University

What is Machine Learning?

When I was growing up, computers able to drive cars, identify cancer more accurately than radiologists, master chess and go, and interact with humans in such a way that it is not possible to tell man from machine was the stuff of science fiction. Today, these are all realities and do not even require particularly powerful computers. Machines are, at present, able to learn many things better and faster than humans, and the set of areas in which artificial intelligence dominates organic intelligence is increasing daily. In fact, it is difficult to use a computer or phone today without interacting with some form of artificial intelligence, whether it be the biometric lock on the device or the content/ad/product recommendations offered on many websites. While some are concerned about this course of events, I for one am excited and feel privileged that we may well see the birth of genuine, sentient, artificial intelligence our lifetimes. But how can a machine learn and what are the implications of machine learning for political science research?

Machine learning (ML) is a subfield of artificial intelligence in which the machine is trained to improve its performance on some task based on algorithms and/or statistical models, but without being specifically programmed to perform that task. To make this definition more intuitive, consider the example of facial recognition software. My phone unlocks when I look at it, but how is that possible? A machine learning algorithm has "studied" many pictures, including pictures of me, and learned to recognize me apart from anyone else based on my features even though no one programmed anything about my features into the phone. This is fundamentally different than, say, a calculator in which the laws of mathematics have been explicitly programmed, or an app that is programmed to open upon being clicked.

Implicit in the definition above is that training is essential to learning. Just as you or I require practice with a given skill, so does ML. The set of data on which the algorithm "learns" is called the *training set* and the set of data on which the algorithm applies what it has learned is called the *test set*. Naturally, these datasets do not overlap (e.g. you cannot test your prediction about who will win a given Formula 1 race when you already know the race's outcome). Also just as with you or I, practice makes perfect: ML algorithms perform better the more training data they receive. An algorithm trained on 1,000,000 observations (called *instances* in ML parlance) and 10,000 independent variables (called *features* in ML) is expected to perform much better than an algorithm trained on 100 instances and 10 features.

Political scientists are not generally in the business of developing self-driving cars, so for what do we use machine learning? For the most part (and there are some very exciting exceptions to this claim), we use ML for prediction. ML algorithms can usually predict outcomes with greater accuracy than the standard regression-type models we use in political science (often *much* greater accuracy), though this comparison is not really fair because we typically use regression and other statistical models to perform deductive hypothesis tests rather than inductive learning. In fact, inductive learning has been taboo in the discipline for decades and is only recently beginning to gain some traction. When I was a graduate student learning statistical methods for the first time, my first year professors routinely cautioned the students that they *must always* have a hypothesis prior to analyzing data and than "data mining" was a fruitless and misguided enterprise in the social sciences.

While it seems this attitude is common in introductory graduate political science education, many years later I have now come to see it as almost entirely wrong.

Naturally then, this Virtual Issue should consider why inductive learning is valuable. The first article in the issue is dedicated entirely to that topic and I will try not to echo it too much, but I wish to highlight here my belief that inductive learning, as a first step to eventual deductive hypothesis testing is actually essential. To illustrate this claim, I will consider my work in my primary substantive field of international relations and conflict processes specifically. In the social sciences, we are typically taught that the first step of the scientific method is to develop a causal theory that produces falsifiable hypotheses. It is telling that in other fields, like biology and physics, the first step of the scientific method is "observe nature." Indeed, developing a sound causal theory of war is completely impossible if we have *absolutely no* knowledge of war or peace. So, from where do the foundations of a theory of war emerge? Almost universally (and I am certainly guilty of this as well), we base our theories of war on our knowledge of history and our consideration of more current events. Yet reflecting on my knowledge of history, American and European wars are dramatically over-represented and among them wars in the last 100 years are even further over represented. In other words, the data upon which any theory I develop is "trained" suffers from massive selection bias and it would in fact be rather surprising if my theory were not biased as a result. While I have pointed to selection problems in my own understanding of history, I would claim with high confidence that I am not alone in having such limitations. To make the point more clearly, let us consider the other side of this coin. Were I to claim that my knowledge of history is perfect – that it suffers from no omissions, no biases in terms of geography, time, or process type, and that my mind is so powerful that I am able to synthesize all these data and recognize relevant patterns in them upon which to ask "why" and develop my theory – that claim would be laughably absurd, yet that is the claim being made implicitly when nature is not explicitly observed as part of the theory generation process. Powerfully though, that claim is entirely credible when an ML algorithm is provided a random sample of training data. I feel that the biased and inherently limited observation of nature prior to theory formation is problematic and that has fueled much of my interest in machine learning.

By way of defending inductive learning, I would also be remiss not to point out the rather obvious fact that deductive hypothesis testing need not be the end goal and that accurately predicting outcomes in politics is enormously useful: strong and timely predictions can save money, lives, and alter the course of history. If the world is to be a better place for having political scientists in it, we would do well to get much better at prediction.

A Comically Brief History of Machine Learning

Like all computing, machine learning traces its roots to Leibniz's (1703) development of binary, Babbage's theoretical proposal for general purpose punch-card programming (Babbage 1864, Babbage 1889), and Lovelace's development of algorithmic computation (Menabrea & Lovelace 1843) under Babbage's advising.[1] Unlike other areas of computing however, ML's pre-history owes much to mathematical and statistical concepts. In particular, the development of Bayes' Theorem (Bayes 1763, Laplace 1812) and its ability to update probabilities foreshadow the revolution in machine learning that would take nearly 200 years to come to fruition. The advent of least squares (Legendre 1805) and Markov chains (Markov 1906) also lay important foundations.

Machine learning, as a field, began in earnest in the early 1950s. Alan Turing, very much the father of artificial intelligence (and computer science for that matter; all that after he was a key playing in breaking the Nazi Enigma encryption rendering great aid to the Allied effort in WWII), had been developing foundational concepts in modern computing for some time, but in 1950 he published "Computing Machinery and Intelligence" in which he articulated a process that would later be developed into what we call genetic algorithms today and

[1] Interesting historical fact: the developer of the first computer program, author of the first published computer algorithm, and thus first computer programmer – Ada Lovelace – was an English woman who accomplished these feats at the age of 27. Unfortunately, she died of cancer at the age of 36. It is also worth noting that some historians claim Babbage was primarily responsible for these developments.

proposed that this was a path to machines that could learn (Turing 1950). It is worth a short digression here to note that Turing's visionary research was brought to an abrupt and repugnant halt shortly thereafter. In 1952, Turing called the police to report that his home had been burglarized. In the course of their investigation, the police discovered that Turing was homosexual and arrested him for "gross indecency." Stripped of his security clearance and forced to accept chemical castration in order to avoid prison, Turing took his own life in 1954. In 2009, 55 years later, Prime Minister Gordon Brown formally apologized and in 2013 The Queen issued Turing a posthumous pardon. Turing's 1950 paper was followed rapidly by the 1951 building of the "Stochastic neural analog reinforcement calculator" by Minsky and Edmonds, a first attempt at artificial intelligence designed to mimic a 40 synapse neural network. In 1952 IBM created the first program to play checkers and improve its game based on experience. 1958 and 1967 saw important algorithmic advances with the perceptron (Rosenblatt 1958) – an early classification algorithm – and the nearest neighbor algorithm (Cover & Hart 1967) to plot map routes respectively. All told however, the mid-1960's to early 1980's see a lull in innovation many call the "winter of machine learning" that is not broken until a burst of advances lead to the development of Artificial Neural Networks (Fukushima 1980), Recurrent Neural Networks (Hopfield 1982), the re-discovery of backpropogation (which was actually developed in 1970 (Linnainmaa 1970) but largely unused), Q-learning (Watkins 1989), random forests (Ho 1995), and support vector machines (Cortes & Vapnik 1995). Though artificial intelligence was alive and well in the public imagination during the "winter" – producing some excellent science fiction such as 2001: A Space Odyssey (1968), Blade Runner (1982), and The Terminator (1984) – AI and machine learning were thrust publicly into science *fact* when IBM's Deep Blue defeated chess grandmaster Gary Kasparov in 1997.

What we might call the modern period of machine learning began in the early 2000's with the rapid advance of recurrent artificial neural networks, which have come to be called called *Deep Learning* algorithms. This modern period is characterized not just by rapid advances in nearly every academic subfield of machine learning, but with the widespread application of machine learning to medicine, government, and business. While any systematic cataloging of this explosion in research and application is far beyond the scope of this discussion, highlights include a dramatic advance in natural language processing that has seen IBM's Watson computer beat humans at *Jeopardy!* and become a virtual sales associate for the North Face's online store, Facebook's DeepFace algorithm that can recognize human faces with better accuracy than humans (about 98%), Google's AlphaGo defeating the world's best human Go players, and yes, self-driving cars.

The Landscape of Machine Learning

When considering machine learning algorithms, it is helpful to be aware of two dimensions upon which they can be classified: how they learn and what they do. Let us consider each in turn.

How they Learn: Supervised, Unsupervised, and Reinforced

The way an ML algorithm learns can be broken down into three broad learning paradigms: supervised learning, unsupervised learning, and reinforced learning.

Supervised learning occurs when the training set upon which the algorithm is to learn is *labeled*. Labeling means that, in the training set, the desired output of the algorithm has been coded (usually by humans) so that the algorithm can learn by comparing its outputs with the "taught" outputs (e.g. the labels) and modify its model to optimize accuracy. A common example is the Capcha system, where websites will often ask visitors to verify that they are human by selecting, for example, photos with cars in them from among a set of photos that include cars and photos that do not. With many photos labeled by many humans, the AI trained on these data should be able to distinguish photos of cars from other photos. In political science, supervised learning is the most common from of ML and occurs when we use a training set to train an algorithm to predict civil war (Ward, Greenhill & Bakke 2010), transnational terrorism (Desmarais & Cranmer 2013), or elections (Klarner 2018). Common supervised learning algorithms include decision trees and random forests, linear and logistic regression, naive Bayes, nearest neighbor algorithms, and artificial neural networks.

Unsupervised learning occurs when all data (including the training data) are unlabeled and the algorithm is left to identify commonalities on its own. Unsupervised learning algorithms are able to sift vast amounts of data and identify patterns that might not be perceptible to humans. Unsupervised learning is primarily used for clustering – which means to find patterns and groupings in unlabelled data and is similar to classification except that in classification the classes are defined by the (human generated) labels in the training set. Marketing is a major application area for unsupervised learning – such algorithms are used to group customers by purchasing behavior and/or identify co-purchasing patters in order to generate targeted and personal advertisements. Common algorithms for unsupervised learning include k-means (used more in industry than academia because of tight assumptions) and neural networks.

Reinforcement learning trains algorithms to optimize some sort of score. Though reinforcement learning can sometimes sounds similar to supervised learning, it differs in that the algorithm is never explicitly corrected but instead uses Markov decision processes to take actions in an environment that affects its state. For example, if a Go AI were to be trained with reinforcement learning (as Google did with AlphaGo), over many many games (virtual, against another AI so that millions of games may be played in a short amount of time) the algorithm explores many strategies and learns how to exploit the most effective ones. This works even in complex environments like chess where a different move at any one state will result in every subsequent state being different. For this reason, reinforcement learning is often used to train AIs to play games, but it is also useful in areas such as robot control.

What they Do: Classification and Regression

The two primary classes of problems for which machine learning algorithms are useful are classification and regression.

Classification is the problem of predicting which discrete type an observation will fall into. Put another way, classification seeks to predict qualitative outcomes. For example, given an MRI image of a tumor, is it cancerous or not? In political science we might, for example, use classification algorithms to predict whether a certain seat goes to a Democrat or Republican. I should also note that there is a subtle but important difference between classification and *clustering*. Classification is a form of supervised learning that requires user-defined classes, meaning that the analyst must determine what classes are to be predicted, what they mean substantively, and provide a labeled training set. Clustering is a form of unsupervised learning in which the algorithm identifies observations of the same/similar class without being told which classes to look for (e.g. no user defined classes and no labeled training set). In some instances, unsupervised clustering might find the same classes that a supervised classification algorithm would, but that is not guaranteed. On the upside, unsupervised clustering might reveal meaningful patters of which the analyst was unaware. On the downside, unsupervised clustering might reveal clusters that are difficult to interpret substantively. Probably the most common supervised classification algorithm is the decision tree and the most common unsupervised clustering algorithm is k-means.

Regression is interpreted much more broadly in machine learning than it typically is in statistics, econometrics, or political methodology. When the typical social scientist refers to regression, they mean some member or extension of the generalized linear class of models. When someone with a machine learning background says "regression" they mean the problem of predicting a continuous numerical outcome; nothing about least squares or maximum likelihood is implied (though such models can be used in an ML context). In other words, regression problems are problems in which one wants to predict "how much" or "how many." For example, predicting trade flows or stock prices are regression problems. Popular machine learning algorithms for regression – such as boosting, lasso regression, or elastic net regression – often involve modification of linear or generalized linear models, though other approaches such as deep/neural networks bear little resemblance to the linear additive models we in political science typically associate with regression.[2]

[2] It is worth noting however that the activation function within a single node of a deep/neural network does bear some resemblance to this structure.

Articles in this Virtual Issue

Selecting articles for inclusion in this virtual issue, I have attempted to strike a balance between articles that contribute to machine learning methodology and articles that apply machine learning to important political science problems. This job was made difficult by the fact that, likely because of political science's longstanding aversion to inductive research discussed above, much of the great and cutting-edge research at the intersection of machine learning and political science is published outside of political science journals. Further, much of the research in *Political Analysis* that uses machine learning does so in application to natural language processing (i.e. text-as-data). That is natural because natural language processing is a burgeoning area in machine learning across fields, but *Political Analysis* recently released an excellent Virtual Issue on innovations in Text Analysis. To avoid overlap of content with that Virtual Issue, I deliberately excluded articles using machine learning for language processing. While I find that somewhat unfortunate as it excludes many important articles, including an excellent recent work by Denny & Spirling (2018) that was published too recently to be included in the Text Analysis issue though I believe it will prove to be a foundational work in that area, these restrictions forced me to focus on articles that address the quantitative and discrete approaches to machine learning. I have sought to include a range of dates to illustrate the history of ML in our discipline, though the recency of ML's entry into political science naturally implies that the selection of articles will be skewed towards more recent work.

At the risk of shameless self-promotion, the first article in this issue is one I wrote with Bruce Desmarais. **Cranmer & Desmarais (2017)** provides a detailed discussion of the merits and limitations of prediction as an enterprise in political science. As one of the primary impediments to machine learning research in political science is the discipline's traditional aversion to inductive learning and a tendency to shy from prediction as a tool only for "applied" sciences (e.g. policy studies) in favor of a focus on deductive hypothesis testing, I feel it appropriate to open this Virtual Issue with a discussion of what we can learn from predictive modeling, how we can use prediction to aid deductive testing, and why prediction itself is useful. The article also covers some basic concepts such as out-of-sample prediction and choosing an appropriate accuracy metric for the type of outcome being predicted.

The next article in the issue is a classic. **Schrodt's (1990)** article is a foundational work on machine learning in political science and, judging by the fact that most ML articles in political science have been published after 2011, an impressive 20 years ahead of its time. Using a form of decision tree algorithm, Schrodt predicts the outcomes of interstate conflicts, finding that the algorithm is 50-60% accurate in out-of-sample prediction and that only a small selection of the available predictors are necessary to accomplish high predictive accuracy.

Ahlquist & Breunig (2012) provide a powerful illustration of the point that researchers should be careful not to rely too much on their personal knowledge of a subject, as human perceptions are frequently biased. The authors apply an unsupervised clustering technique to typology construction. They find that, when applied the the "varieties of capitalism" problem, existing theory provides a typology that is weak at best. While the practical advice they derive from this study is that including typology-driven indicators in regressions is a bad idea, the illustration of the limits of the human ability to process and categorize data into meaningful typologies struck me powerfully when I first read this article and has colored much of my thinking about the importance of inductive learning in the years since.

Montgomery, Hollenbach & Ward (2012) address how predictive models can be further improved by post-processing. The authors show that, when aiming for optimum predictive accuracy, forecasting models need not be pitted against one another and a "winner" selected. Rather they propose using ensemble Bayesian model averaging (EBMA) to create even more accurate predictions from several component models by pooling information across them in a process somewhat similar to a weighted average. After demonstrating the technique's efficacy across three applications – predicting insurgencies, electoral vote shares, and Supreme Court votes – the article introduces easy-to-use software to implement the method.

Peterson & Spirling (2018) show that the predictive prowess of machine learning algorithms can offer new opportunities for measurement. Because accurate estimation of ideal points based on roll-call votes is notoriously hard in parliamentary systems, they propose to use the accuracy of a supervised ML classifier to measure the polarization of parliament. Specifically, they train the algorithm to predict the party of the member using a set of features on the member's participation (including what they said) during a series of debates. When the predictive accuracy of the algorithm is high, so is polarization in the legislative body; when it is low, so is polarization. An elegant way to solve a difficult measurement problem.

Cantú & Saiegh (2011) use machine learning to tackle a problem of tremendous societal importance: detecting election fraud. They introduce to political science a technique for using synthetic data in the training set to train a naive Bayes classifier to classify elections as either fraudulent or legitimate. They evaluate their trained classifier on a test set of district-level elections in Buenos Aires between 1931 and 1941, finding that the classifier's decisions match historical accounts very closely. This paper is noteworthy not only because it uses ML to address a societally relevant problem, but also in its novel (in political science) use of synthetic data to train the classifier.

Addressing the same problem and building on Cantú & Saiegh's (2011) work, **Montgomery, Olivella, Potter & Crisp (2015)** develop an approach to detecting election fraud that is able to combine an election forensics approach (i.e. searching for anomalous numerical patterns) with contextual features (i.e. measures that correlate to a country's risk of fraud). The authors train a Bayesian additive regression tree (BART) model to combine these classes of features in the search for fraud. They find that the BART model substantially improves out-of-sample predictive performance. Moreover, they illustrate the extent to which ML can inform the substance of political science because the BART model can judge the relative importance of different features (and thus different classes of features).

A Proposed Agenda for Machine Learning in Political Science

Perhaps I will re-read this essay at the end of my career and be aghast by the inaccuracy of my predictions, leaving me to wonder why I put them in print, but I believe machine learning will play an important role in the coming decades of political science. Specifically, I see three areas where machine learning has at least the *potential* to transform the way the discipline works.

First, machine learning has a large and integral role to play in the automated and large scale harvesting of political science data that is likely to color political research in decades to come. Already, machine learning algorithms are at the forefront of natural language processing; it would not be possible to turn large bodies of text into data without machine learning. Though I have deliberately not included work on natural language processing in this virtual issue because there is already a virtual issue on that topic, ML makes it possible to categorize speech (e.g. sentiment analysis) and even extract event data from text. Our capabilities in this area of research will only expand, thus dramatically expanding the quality and scope of data available to researchers. It is also worth noting that while current efforts are rightly focused on text processing, ML algorithms are capable of doing the same/similar things with voice, image, and video data. As such, I see ML as the most powerful data gathering/processing tool currently in our arsenal.

Second, I believe ML will be instrumental in allowing political science to have a greater policy and social impact than it has so far. This, I posit, will be largely due to advances in our ability to accurately predict important phenomena. Prediction has not been a traditional focus for political scientists outside of election forecasting, but I believe our field's near-exclusive focus on explanation is misguided. I wholeheartedly agree with Guo, Gleditsch & Wilson's (2018) recent call to re-tool machine learning to predict conflict and would expand the claim to other areas of political science as well. That is certainly not to say that explanation is not important; it is very important. But so is prediction. Ideally, prediction and explanation work hand-in-hand with each other and are not necessarily competitors. That said, predictive work is more easily translated into policy or social action with the ability to significantly improve the world in which we live. Whether it be work on predicting political violence, democratization, treaty ratification, public transit utilization, or really any area of political science, strong predictions allow for strong actions. A great example of ML being used to these ends is Erin Lin's (2018) work in which she uses ML to analyze high-resolution satellite imagery of rural Cambodia, identifying bomb craters and predicting the location of unexploded ordnance from bombing during the Vietnam war. This research improves detection by over 160% and guides de-mining teams in an endeavor that literally saves lives. As I said above, I will say again: if the world is to be a better place for having political scientists in it, we would do well to get much better at prediction. Machine learning is the major innovation that we can use to do so.

Finally, I believe that a good deal of the important advances in machine learning for political science will come not from within the academy, but from industry. Companies such as Facebook and Google are hiring political scientists with expertise in machine learning, computer scientists, statisticians, and others to work on important tasks, such as protecting the integrity of US elections from foreign interference. Obvious "political" examples such as this aside, many companies can benefit from accurately predicting the sort of phenomena that fall within the domain of political science and have powerful financial incentives to do so. Because the predictive ability of machine learning lends itself so well to product-oriented social/political research, I suspect this will be an important force in years to come.

Final Remarks

Machine learning has thus far been used fairly sparingly in political science, though the frequency of its use has increased dramatically in recent years due to advances in computing power as well as the algorithms themselves. In my opinion, this is mostly a good thing. Improving our ability to predict improves both our base-knowledge of the phenomena we study thus allowing for better explanatory theories, and also improves the relevance of our research outside of academia. A word of caution is however in order. Because machine learning algorithms do literally learn from their training data, they are able to learn things we do not want them to. For instance, it is possible for ML algorithms to re-inscribe social biases, such as racism or sexism. This is possible even if features like race and sex are removed because many other observable features are correlated with things like race or sex. Depending upon which algorithm is being used, it may even be difficult to identify that this is happening; artificial neural networks for example do not provide intelligible feedback as to how they make their classification decisions. To give a more concrete example, there has been considerable consternation over courts using an ML-based product called Correctional Offender Management Profiling for Alternative Sanctions or COMPAS (produced by the Equivalent corporation) to help determine criminal sentencing based on risk scores computed for each convicted offender. The COMPAS software takes race and sex into account, as well as many other factors. In 2017, the US Supreme Court declined to hear a case alleging that use of COMPAS violates due process, allowing a Wisconsin Supreme Court ruling in favor of the State[3] to stand.

While we must not be naive about the dark side of the recent revolution in machine learning, we must not be overly paranoid either. There is much our field can gain from this relatively new technology. It can provide us with new knowledge and empower us to make greater societal impact with our efforts. The body of ML work in political science – especially work on elements of ML other than natural language processing – is relatively small, but growing in volume and importance. I have every confidence that *Political Analysis* will be a leading outlet for innovations in and with machine learning in the years to come.

Funding

The author gratefully acknowledges support from the National Science Foundation (SES-1514750 & SES-1461493), the National Institutes of Health (R-34, DA043079-01A1), and the Mershon Center of the Ohio State University.

[3] *State v. Loomis*, 881 N.W.2d 749 (Wis. 2016)

About the Author

Skyler Cranmer is the Carter Phillips and Sue Henry Associate Professor of Political Science at the Ohio State University. His areas of interest include network science and machine learning with applications ranging from conflict processes to neuroscience.

References

Ahlquist, J. S. & Breunig, C. (2012), 'Model-based clustering and typologies in the social sciences', *Political Analysis* **20**(1), 92–112.

Babbage, C. (1864), Chapter viii - of the analytical engine, *in* C. Babbage, ed., 'Passages from the Life of a Philosopher', Longman, Green, Longman, Roberts, & Green, London, p. 112–141.

Babbage, C. (1889), Babbage's calculating engines - being a collection of papers relating to them; their history, and construction, Cambridge University Press, New York.

Bayes, T. (1763), 'An essay towards solving a problem in the doctrine of chances', *Phil. Trans. of the Royal Soc. of London* **53**, 370–418.

Cantú, F. & Saiegh, S. M. (2011), 'Fraudulent democracy? an analysis of argentina's infamous decade using supervised machine learning', *Political Analysis* **19**(4), 409–433.

Cortes, C. & Vapnik, V. (1995), 'Support-vector networks', *Machine Learning* **20**(3), 273–297.
URL: *https://doi.org/10.1023/A:1022627411411*

Cover, T. & Hart, P. (1967), 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory* **13**(1), 21–27.

Cranmer, S. J. & Desmarais, B. A. (2017), 'What can we learn from predictive modeling?', *Political Analysis* **25**(2), 145–166.

Denny, M. J. & Spirling, A. (2018), 'Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it', *Political Analysis* **26**(2), 168–189.

Desmarais, B. A. & Cranmer, S. J. (2013), 'Forecasting the locational dynamics of transnational terrorism: a network analytic approach', *Security Informatics* **2**(1), 8.
URL: *https://doi.org/10.1186/2190-8532-2-8*

Fukushima, K. (1980), 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position', *Biological Cybernetics* **36**(4), 193–202.
URL: *https://doi.org/10.1007/BF00344251*

Guo, W., Gleditsch, K. & Wilson, A. (2018), *Nature* **562**(7727), 331–333.

Ho, T. K. (1995), Random decision forests, *in* 'Proceedings of 3rd International Conference on Document Analysis and Recognition', Vol. 1, pp. 278–282 vol.1.

Hopfield, J. J. (1982), 'Neural networks and physical systems with emergent collective computational abilities', *Proceedings of the National Academy of Sciences* **79**(8), 2554–2558.
URL: *https://www.pnas.org/content/79/8/2554*

Klarner, C. E. (2018), '2018 state legislative election forecasts', *PS: Political Science & Politics* **51**(S1), 21–27.

Laplace, P. (1812), *Théorie Analytique de Probabilités*, Courcier, Paris.

Legendre, A. M. (1805), *Nouvelles methodes pour la determination des orbites des cometes [microform] / par A.M. Legendre*, F. Didot Paris.

Leibniz, G. (1703), 'Explication de l'arithmétique binaire', *Memoires de l'Académie Royale des Sciences* **3**, 85–93.

Lin, E. (2018), Detecting bomb craters from very high resolution remote sensing images for potential estimation of uxo in rural cambodia. Working paper.

Linnainmaa, S. (1970), The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors, Master's thesis, University of Helsinki, Helsinki.

Markov, A. (1906), 'Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga', *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete* **15**(2), 135–156.

Menabrea, L. F. & Lovelace, A. (1843), 'Sketch of the analytical engine invented by charles babbage, with notes upon the memoir by the translator', *Scientific Memoirs* **3**.

Montgomery, J. M., Hollenbach, F. M. & Ward, M. D. (2012), 'Improving predictions using ensemble bayesian model averaging', *Political Analysis* **20**(3), 271–291.

Montgomery, J. M., Olivella, S., Potter, J. D. & Crisp, B. F. (2015), 'An informed forensics approach to detecting vote irregularities', *Political Analysis* **23**(4), 488–505.

Peterson, A. & Spirling, A. (2018), 'Classification accuracy as a substantive quantity of interest: Measuring polarization in westminster systems', *Political Analysis* **26**(1), 120–128.

Rosenblatt, F. (1958), 'The perceptron: A probabilistic model for information storage and organization in the brain', *Psychological Review* **65**, 386–408.

Schrodt, P. A. (1990), 'Predicting interstate conflict outcomes using a bootstrapped id3 algorithm', *Political Analysis* **2**, 31–56.

Turing, A. (1950), 'Computing machinery and intelligence', *Mind* **59**(236), 433–460.

Ward, M. D., Greenhill, B. D. & Bakke, K. M. (2010), 'The perils of policy by p-value: Predicting civil conflicts', *Journal of Peace Research* **47**(4), 363–375.
URL: *https://doi.org/10.1177/0022343309356491*

Watkins, C. J. (1989), Learning from delayed rewards, PhD thesis, University of Cambridge, Cambridge.