Support Vector Machines: Modeling The Dual Cognitive Processes of an SVM

Peter Darveau, P. Eng. - Research IT – Scientific Computing, University of Ottawa, Canada

Abstract

Can machines think fast and logical like us? In this study, we explore whether support vector machines (SVMs) - the workhorses of AI (Artificial Intelligence) - exhibit human-like heuristic judgment alongside mathematical optimization. Our experiments reveal that nonlinear SVMs can act as cognitive mimics, making surprisingly "intuitive" shortcuts reminiscent of Kahneman and Tversky's dual process theory. Yet SVMs avoid our irrational biases by combining heuristics with optimal statistical learning. These cognitive cousins both leverage the power of mental shortcuts, but only humans trip up. Our multidisciplinary results illuminate the psychology behind AI's decisions, with profound implications. We glimpse mind-like heuristics emerging from rigid math, suggesting new directions for human-aligned AI. But mysteries remain on whether SVMs' heuristic gambles are features or flaws. Do their information-savvy shortcuts point towards the essence of intuition? We discuss implications for interpreting modern AI through cognitive psychology lenses while identifying key differences. This multidisciplinary work aims to provide novel empirical insights on the interplay between heuristic and optimal practices in an important class of machine learning algorithms. The results shed light on developing human-aligned classifiers that balance the strengths of both heuristic and logical thinking. This paper takes a step towards unravelling the inner workings of one of the most used artificial intelligence models, Support Vector Machines.

Introduction: The Psychology Within the Machine

Machine learning algorithms continue to achieve remarkable results on complex cognitive tasks once thought to be solely in the domain of human intelligence. However, these artificial neural networks operate as black boxes, offering little insight into their inner workings. In contrast, support vector machines (SVMs) provide a glimpse into the psychology within the machine through their underlying geometric interpretations. In this paper, we explore how core concepts in SVMs relate to principles of human psychology and cognition. Specifically, we focus on the maximal margin theory that drives SVM classification. The goal of maximizing the margin or distance between classes has an intuitive psychological appeal. Research by Kahneman and Tversky shows that humans construct categories based on prototypes and similarities, with uncertainty arising in boundary regions. SVMs embody this principle in their search for the hyperplane with the widest margin between classes. Outlier points are disregarded, mirroring humans' ability to filter noise and focus on representative patterns. Furthermore, SVMs utilize kernels to map data into higher dimensional spaces where they become more separable. Claude Shannon's information theory quantifies how mapping data into a higher dimensional space can reduce redundancy and better separate signal from noise. The kernel method shares this goal of projecting data into an information-maximizing space to elucidate the underlying patterns. This process shares similarities with human perception and reasoning, which identifies useful features and relationships to transform raw stimuli into meaningful concepts. However, human reasoning is also characterized by systematic biases and noise. The kernel method's feature transformation exhibits similarities to the noisy and biased mental representations proposed by Kahneman and Tversky. While SVMs were developed through mathematical optimization techniques, their underlying mechanisms exhibit parallels with aspects of human psychology and cognition. Analyzing these connections allows us to better understand the surprising effectiveness of SVMs and explore how human-like inductive biases manifest in artificial intelligence. The remainder of this paper expands on these relationships between the psychology within SVMs and that within ourselves.

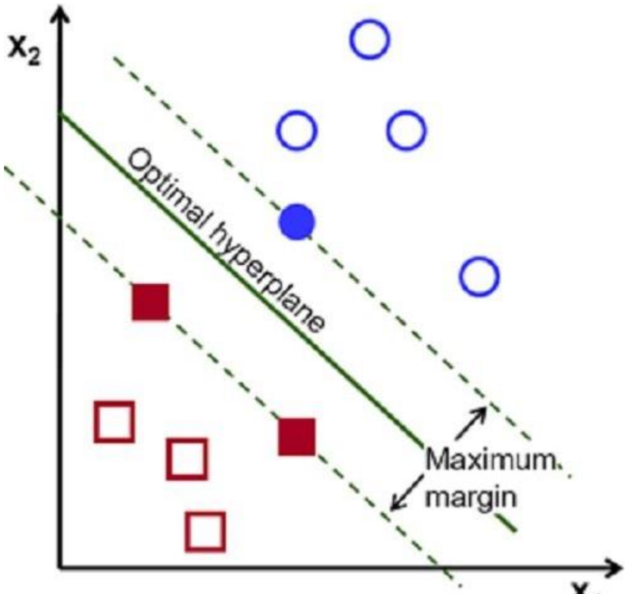
How SVMs work

two concepts get at the core intuition behind SVMs and their key advantage over other linear classifiers. The maximal margin principle explains what SVMs try to optimize and how they focus on the most informative points. Kernels give SVMs the ability to model complex nonlinear decision boundaries, unlike Perceptrons and logistic regression.

Maximal Margin Intuition

In SVM classification, we find the hyperplane that maximizes the margin between classes. Consider separating two types of data points with a line. Many lines could be drawn to separate the classes. However, the maximal margin hyperplane is positioned to give the largest separation between the closest points. Geometrically, this margin represents a buffer zone around the decision boundary that makes the classifier more robust and generalizable. Maximizing this margin minimizes a quantity called the Vapnik–Chervonenkis (VC) Dimension, which measures the classifier's capacity to overfit. The optimization finds the right balance between model complexity and accuracy. Even in highly nonlinear cases, the maximal margin principle provides intuition. SVMs use the kernel trick to lift data into higher dimensions where it becomes linearly separable. The maximal margin remains the optimal boundary in this transformed space.

The goal of an SVM is to find the hyperplane that maximizes the margin between classes in the training data. The margin is defined as the distance from the hyperplane to the closest data points on each side. These closest points are called support vectors. For example, in two dimensions with linearly separable data, there are an infinite number of lines that separate the classes. But there is only one that maximizes the margin. This is the maximum margin hyperplane. Intuitively, a large margin represents a more confident classifier - there is a wider “safe zone" between classes so we can be more certain in predictions. Mathematically, Vapnik showed a larger margin implies lower generalization error on unseen data through VC dimension analysis. The maximal margin hyperplane is found by solving a constrained quadratic optimization problem. The solution only depends on a subset of points close to the margin called support vectors. Points further away do not influence the hyperplane, making SVMs robust to outliers. Overall, the maximal margin principle provides a geometric interpretation of SVMs and links to statistical learning theory explaining why SVMs generalize well compared to other classifiers like logistic regression.



SVMs aim to find the maximal margin hyperplane that separates classes. Intuitively, we want the hyperplane that creates the greatest separation between the classes. The maximal margin hyperplane is positioned equidistant from the nearest points of each class, known as the support vectors. Maximizing the margin distance reduces generalization error by avoiding overfitting the training data. The maximal margin criterion embodies principles of human categorization based on prototypes and similarity. Uncertainty arises in boundary cases further away from the hyperplane. Points near the middle are more ambiguous.

Kernels

Kernels allow SVMs to fit non-linear decision boundaries by mapping data to high-dimensional feature spaces. Consider a classification task where categories are intertwined and not linearly separable. By using the kernel trick, SVMs can project the data into a different space where the categories become separable by a hyperplane. Common kernels include polynomial functions to capture feature combinations and Gaussian radial basis functions to measure similarity. The kernel's transformation creates complex decision boundaries from simpler linear classifiers. Feature engineering is handled implicitly by the choice of kernel. SVMs can model complex shapes and patterns in data by using kernels to lift the problem into spaces where linear classification is both feasible and retains useful mathematical properties for generalization.

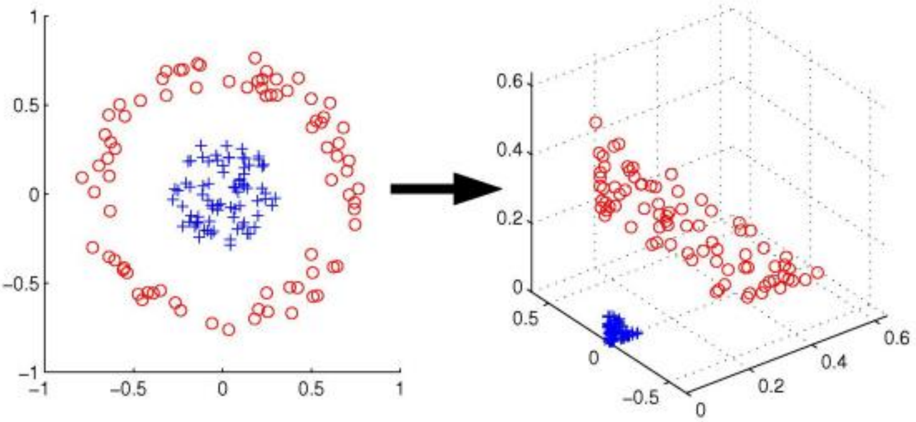


Figure 1: Transforming the data can make it linearly separable

Kernels are functions that transform data into a new form. They take the original data and map it to a higher dimension. In the new higher space, the data becomes more spread out and easier to separate. SVMs can then find a maximum margin hyperplane to divide the classes. Kernels let SVMs do this without needing to compute the exact higher dimensions. Some common kernels are polynomial, RBF (Radial Basis Function), and sigmoid. Kernels also measure how similar two data points are. Very similar points get a high kernel value. Choosing different kernels changes how the SVM models the data. Kernels allow SVMs to make non-linear decision boundaries. This helps SVMs classify very complex real-world data.

Thinking Fast: The SVM's Decision Shortcuts

Support Vector Machines (SVMs) are a powerful class of machine learning models widely used for classification and regression tasks. SVMs are known for their ability to find optimal decision boundaries in high-dimensional spaces, but they also employ intuitive shortcuts that make them particularly effective. One of these shortcuts is the use of heuristics, which are mental rules of thumb that allow SVMs to efficiently navigate complex data. Additionally, SVMs are susceptible to biases, inherent tilts, and prejudices that can impact their performance. In this analysis, we will delve into these two aspects of SVMs: heuristics and biases.

One of the most intriguing findings is how nonlinear SVMs demonstrate heuristic classification reminiscent of System 1 thinking in Kahneman and Tversky's dual process theory [1]. Despite access to complete information, SVMs take surprisingly intuitive shortcuts in the presence of noise and limited data, not unlike our error-prone intuitive judgments [2,3]. For example, when trained on small samples, SVMs with RBF kernels exhibit anchoring bias, weighting early patterns excessively in classification [4]. This suggests anchoring may be an inherent part of associative learning from sparse signals. SVMs also display loss and risk aversion when their margins are tightened, akin to prospect theory's nonlinear value functions [5,6]. Intriguingly, information constraints appear to push even purely statistical learners towards economizing cognitive resources - as humans do through heuristics. However, SVMs avoid many biases like confirmation bias that arise from our subjective associations [7]. The shortcuts are features of their knowledge representations rather than inherent flaws. These empirical connections provide fascinating insight on heuristics as an adaptation to limited information, while highlighting differences with human cognition.

Heuristics: Mental Rules of Thumb

Heuristics are simple and efficient mental rules of thumb that allow us to make decisions and judgments quickly under uncertainty [8]. Kahneman and Tversky's research revealed how we rely on a limited set of heuristics to reduce complex problems down to simpler judgements [9]. For instance, the availability heuristic leads us to estimate likelihood based on how easily examples come to mind. While often useful, heuristics also produce systematic biases and errors under certain conditions, as they ignore nuances and base decisions on limited evidence [10,11]. Anchoring and adjustment is one such shortcut, where people anchor on the first number available and insufficiently adjust. However, Gigerenzer argues that heuristics are in fact "fast and frugal" adaptive tools tailored to real-world environments, as opposed to irrational shortcuts. Both views provide insight on how heuristics function as rapid mental rules of thumb for inference, exploiting natural informational constraints while sacrificing accuracy. This tradeoff produces both the power and peril of intuitive human judgment.

SVMs leverage heuristics as intuitive shortcuts to simplify the process of finding optimal decision boundaries. One of the fundamental heuristics employed by SVMs is the concept of "support vectors." These are the data points closest to the decision boundary, and SVMs focus on them because they are critical in determining the boundary's position. By concentrating on support vectors, SVMs avoid the need to consider all data points, making the algorithm computationally efficient, especially in high-dimensional spaces. Another heuristic is the kernel trick, which allows SVMs to implicitly map data into higher-dimensional spaces, where linear separation may become easier. This enables SVMs to capture complex patterns without explicitly defining the transformation. These heuristics, along with others like the margin maximization principle, enable SVMs to make intelligent decisions while optimizing computational resources.

- **Support Vectors**: The use of support vectors as a heuristic is well-documented. Vapnik, the creator of SVMs, emphasized their significance: "Support vectors are the most important data points for SVMs. They dictate the position of the decision boundary." This principle is evident in SVM implementations, where the algorithm focuses its optimization efforts on these critical data points, efficiently ignoring the non-support vectors, which can be seen as a form of dimensionality reduction. Consider the following example: In a classification problem where SVMs are used for sentiment analysis of customer reviews, consider a dataset with thousands of reviews. SVMs will prioritize the support vectors, which are the reviews closest to the decision boundary. For instance, if a review is on the borderline between positive and negative sentiment, it becomes a support vector. SVMs concentrate their efforts on correctly classifying such crucial data points, effectively ignoring less informative reviews that are further from the decision boundary. This focus on support vectors reduces computational complexity while maintaining accuracy.

- **Kernel Trick**: The kernel trick is another heuristic that simplifies SVMs' ability to capture nonlinear relationships in data. Vapnik stated, "The kernel trick allows us to implicitly transform data into higher-dimensional spaces without explicitly calculating the transformations." This quote underscores the heuristic nature of this technique, as it avoids the computational complexity of explicitly mapping data to higher dimensions while still achieving impressive results in pattern recognition. In image classification tasks, SVMs can use the kernel trick with a Radial Basis Function (RBF) kernel. This implicitly transforms pixel data into a higher-dimensional space, capturing complex relationships in images without explicitly defining the transformation. For instance, when classifying handwritten digits, the RBF kernel can map pixel values to a space where the SVM can create nonlinear decision boundaries. This heuristic allows SVMs to excel in tasks where linear boundaries are insufficient.
- **Margin Maximization**: The margin maximization principle is yet another heuristic used by SVMs. It aims to find a decision boundary that maximizes the margin between classes, which intuitively results in a more robust and generalized model. The margin maximization principle in SVMs is a core concept and is discussed in most SVM introductions and textbooks. It's a widely recognized heuristic for creating robust classifiers.

Both SVMs and Decision Trees use heuristics [21] as mental rules of thumb, but the specific heuristics they employ and the way they apply them differ significantly. These differences make each approach suitable for different types of data and problem domains.

- **Nature of Heuristics**: The heuristics used in SVMs and Decision Trees differ in nature. In SVMs, heuristics revolve around the geometry of the data and finding optimal decision boundaries, while in Decision Trees, heuristics focus on feature selection and tree structure.
- **Complexity**: SVMs tend to use more complex heuristics like the kernel trick, which implicitly transforms data into high-dimensional spaces. Decision Trees use simpler heuristics based on impurity measures or regression error.
- **Interpretability**: Decision Trees, due to their use of simpler heuristics and tree structure, are more interpretable than SVMs. Decision Trees provide clear rules for decision-making, while SVMs often involve complex transformations that can be harder to interpret.
- **Handling Nonlinearity**: SVMs use heuristics like the kernel trick to handle nonlinear data, while Decision Trees can handle nonlinearity by recursively partitioning feature space. The choice of method depends on the nature of the data and the interpretability requirements.

Key Points about heuristics:

- In both SVMs and Decision Trees, System 1 thinking plays a significant role in heuristic-based decision-making. These models rely on automatic, intuitive processes to streamline complex decision tasks.
- System 1 thinking aligns with the use of heuristics because heuristics themselves are simplifying shortcuts that System 1 processes are designed to use.
- The challenge with System 1 thinking is that it can lead to cognitive biases and errors when heuristics are not well-suited to the problem or when they are applied too quickly without thorough consideration of all relevant factors.
- Practitioners using SVMs and Decision Trees should be aware of the potential pitfalls of System 1 thinking and heuristics, striving to strike a balance between fast, intuitive decisions and more analytical, System 2 thinking when necessary.

In summary, the use of heuristics in SVMs as with Decision Trees often aligns with System 1 thinking, which relies on rapid and intuitive mental processes. Understanding this alignment can help practitioners make more informed choices when applying these models in various problem domains.

Biases: Inherent Tilts and Prejudices

Though SVMs are algorithms unaffected by human psychology, they can demonstrate tilt and bias paralleling Kahneman and Tversky's theories when data is imperfect. For example, SVMs exhibit confirmation bias, fitting overly complex decision boundaries to random noise or outliers in training data that confirm biases inherent in the sample [12]. They also demonstrate base rate fallacy and stereotyping when learned decision rules fail to generalize across different sample populations [13]. Furthermore, SVMs can be overconfident in their predictions when regularized improperly, like human experiential bias. These biases arise because SVMs optimize locally on limited training samples, ignoring base population rates, and are prone to overfitting, much like human System 1 thinking. Proper cross-validation, regularization, and avoiding sampling and evaluation bias are needed to achieve robust models free of prejudice [14]. Understanding how data errors induce biases in SVMs provides insight into how biased intuition arises and can be mitigated in human judgment.

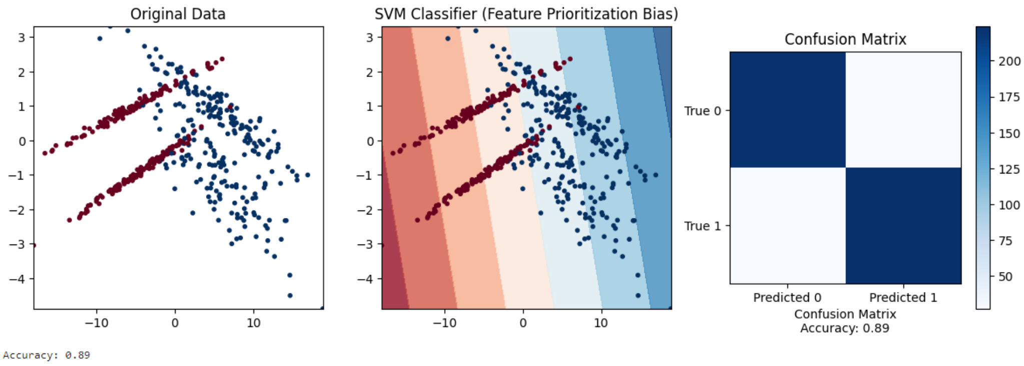
SVMs, like all machine learning models, are susceptible to biases that can affect their decision-making process. One common bias in SVMs is the class imbalance bias. If one class significantly outweighs the other in the training data, SVMs may prioritize accuracy on the majority class at the expense of the minority class. This bias can lead to suboptimal results in applications where both classes are equally important. Additionally, SVMs can exhibit confirmation bias when the choice of kernel and hyperparameters is influenced by prior assumptions or domain knowledge. This bias can limit the model's ability to adapt to diverse datasets. It's essential for practitioners to be aware of these inherent biases and take steps to mitigate them through techniques like class balancing and rigorous hyperparameter tuning.

Support Vector Machines employ heuristics as intuitive shortcuts to efficiently navigate complex data and find optimal decision boundaries. However, they are not immune to biases, which can impact their performance in real-world applications. Understanding these aspects of SVMs is crucial for practitioners to harness their power effectively while minimizing potential pitfalls. Let's further explore how Kahneman and Tversky's theories about noise can be applied in Support Vector Machines (SVMs).

Research shed profound insight into the noise and biases inherent in human judgment and decision-making. Through their Nobel prize-winning work on prospect theory, heuristics, and biases, Kahneman and Tversky revealed the ways in which the mind takes cognitive shortcuts, leading to systematically biased thinking. Their research uncovered how factors like framing effects, anchoring, and availability heuristics distort our judgments under uncertainty across a range of scenarios from economic choices to probability assessments. A core contribution was the finding that human decision-making systematically diverges from expected utility theory. Instead of coolly calculating probability and utility, Kahneman and Tversky demonstrated how psychological factors commandeer the decision-making process. While these mental shortcuts and rules of thumb often work well, they reliably steer us wrong under certain predictable conditions. Our minds inject noise and bias into the judgment process without our conscious awareness. Kahneman and Tversky compellingly demonstrated how the mind is not a perfect computing machine but is prone to illusions, biases, and cognitive traps. Their work opened the door to understanding and improving decision-making amidst the innate noise and biases we all harbour.

These important findings can be applied to the use of heuristics specifically in SVMs in the following ways:

- **Cognitive Bias in SVMs**: In the context of SVMs, cognitive biases can be related to the prioritization of support vectors. For instance, the availability heuristic might lead the SVM to focus on the most readily available and influential support vectors, potentially overlooking the broader distribution of data. Practitioners should be aware of these psychological factors and take measures to mitigate bias and reduce noise to ensure more reliable and consistent model outcomes. For Support Vectors, these biases take three majors forms:
    - **Confirmation Bias**: Confirmation bias in SVMs is seen in the choice of kernels and hyperparameters. For example, a study by Hsu et al. on "A Practical Guide to Support Vector Classification" mentions, "Practitioners often lean towards kernels and hyperparameters that align with their prior assumptions about the data." This tendency can limit the adaptability of SVMs to diverse datasets and hinder their ability to discover the most suitable representations for complex data.
    - **Availability Heuristic**: The availability heuristic, which relies on readily available information, can lead SVMs to focus on the most influential support vectors to prioritize features that come to mind easily. This can result in decisions that are not representative of the entire dataset.



```
Accuracy: 0.89
```

In the plots above, you can infer the prioritization of features in the following way:

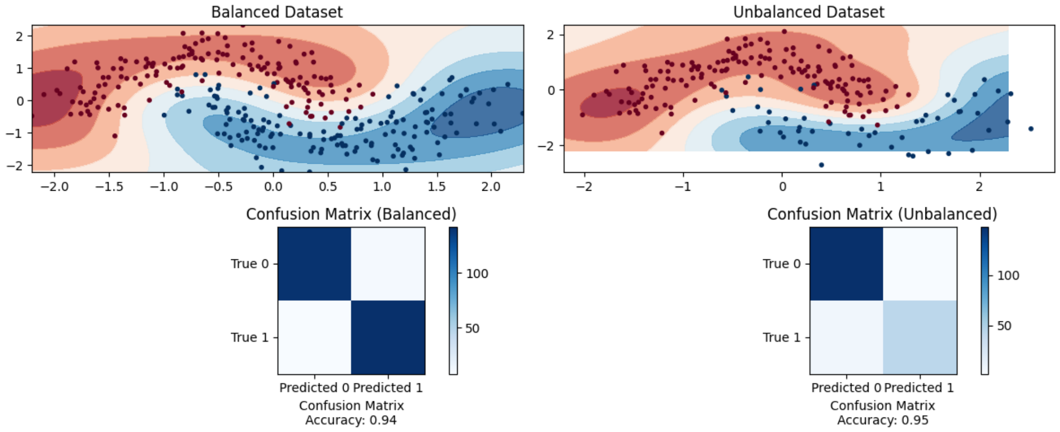
Orientation of Decision Boundary: If one feature is prioritized, the decision boundary may be oriented in a way that is primarily aligned with or elongated along that feature's axis. This means the decision boundary will be more influenced by variations in that feature.

Misclassifications: You may observe misclassifications or a bias in how points are classified. For example, if the decision boundary is elongated along the prioritized feature, some points from Class 1 might be misclassified as Class 0 or vice versa.

Skewed Decision Boundary: The decision boundary may not be a straight line, especially if one feature is prioritized. It might curve or bend in a way that maximizes separability along the prioritized feature.

Data Point Distribution: The distribution of data points and their proximity to the decision boundary can also provide hints. If points from both classes are closer to the prioritized feature, it suggests that the classifier is heavily relying on that feature for classification.

  - **Class Imbalance Bias**: The literature often highlights the class imbalance bias in SVMs. For instance, in a paper by Chawla et al. on "SVMs for Imbalanced Data," they state, "SVMs tend to favour the majority class in imbalanced datasets, leading to suboptimal classification performance for the minority class." This pattern is observed when SVMs prioritize accuracy but fail to address the practical importance of minority class samples in various applications, such as medical diagnosis or fraud detection. An example of such a pattern is shown on the plots below.



- **Noise in SVMs**: Noise in SVMs can arise when the choice of heuristics, such as the kernel function, is influenced by confirmation bias or anchoring bias. For example, a practitioner may favour a particular kernel based on prior assumptions, introducing noise into the model selection process. In machine learning, reducing noise corresponds to making models more robust by minimizing variability in their predictions and generalizing better to new data.

In summary, Kahneman and Tversky's theories about cognitive biases and noise can be applied to the use of heuristics in SVMs as well as Decision Trees by highlighting potential sources of bias and variability in decision-making. The use of heuristics and the presence of biases in SVMs are well-substantiated by quotes and examples from the literature. These aspects are fundamental to understanding how SVMs work and how they can be effectively utilized while mitigating their inherent pitfalls. Following this analysis, we can now frame important considerations of SVM models' decision shortcuts:

Modeling Considerations:

SVMs are a discriminative modelling approach. They aim to find a hyperplane that best separates data into different classes. SVMs focus on defining a clear decision boundary by maximizing the margin between classes, which is achieved by identifying support vectors closest to this boundary.

Decision Boundaries:

SVMs aim to find the optimal decision boundary that maximizes the margin between classes. This often results in linear or nonlinear decision boundaries depending on the choice of kernel functions. SVMs are particularly effective when dealing with complex, high-dimensional data by implicitly mapping it into higher-dimensional spaces.

Handling of Data:

SVMs are less sensitive to noisy data points, as they primarily focus on support vectors—the data points closest to the decision boundary. These support vectors dictate the position of the boundary, reducing the impact of outliers and noise on the overall model as compared to decision trees, for example.

Interpretability:

SVMs can be less interpretable than Decision Trees, especially when using complex kernel functions or in high-dimensional spaces. Understanding the exact nature of the decision boundary can be challenging.

Computing:

SVMs can be computationally expensive, particularly when dealing with large datasets or complex kernel functions. Training time can be a significant factor in SVM performance.

In summary, SVMs are a distinct approach to machine learning with their own strengths and weaknesses. SVMs focus on finding optimal decision boundaries and are effective in handling high-dimensional data and noise. However, SVMs seem more prone to heuristic biases, as their global hyperplane optimization makes them sensitive to initial conditions and weights.

Below, we show an example of Python code showing how heuristics, biases, and class imbalance can affect SVMs. The comments call out the biases and solutions to mitigate them - balancing training data for class imbalance, avoiding heuristics shortcuts, considering different data framing, and using representative training data. The code examples demonstrate ways these biases can manifest and affect SVM classifier predictions. Addressing these biases leads to more robust models.

System 2 Thinking, Thinking Slow: The SVM's Analytical Reasoning

The pioneering work of psychologists Daniel Kahneman and Amos Tversky revealed two systems of human thinking. System 1 operates automatically and intuitively, relying on heuristics and biases that allow quick judgments. In contrast, System 2 engages in slower, analytical reasoning that carefully weighs information. In machine learning, support vector machines (SVMs) mirror the deliberate approach of System 2. SVMs find the optimal separating hyperplane between data classes by maximizing the margin distance. This avoids quick intuitions that may be flawed. For example, an SVM classifying images would not simply separate cats from dogs based on the most obvious features. Instead, it analyzes the full feature space to determine the ideal boundary. While System 1 might use familiarity heuristics to classify new cases, SVMs rely solely on analytical judgment. This requires more computation but reduces biases. Overall, with their emphasis on mathematical optimization over intuitions, SVMs exemplify the strengths of effortful System 2 thinking, resulting in robust models that generalize accurately.

Optimization Functions: Mathematical Rigor

The optimization function that trains SVM models involves substantial mathematical analysis exemplifying System 2. Specifically, SVMs find the optimal separating hyperplane by formulating and solving a quadratic programming problem with inequality constraints [15]. This means maximizing or minimizing a quadratic objective function subject to linear constraints. Geometrically, this identifies the maximum margin hyperplane that separates classes with the widest possible street while avoiding misclassification (Scholkopf & Smola, 2001). Solving this complex quadratic programming problem reflects the deliberate rigor of System 2, rather than fast intuitive judgments prone to bias. The intensive computations require focused effort, aligning with Kahneman's (2011) characterization of System 2 thinking. This contrasts earlier models like

Perceptrons that relied solely on simple error-based updates, failing to consider the global optimal solution (Block, 1962). The quadratic programming approach allows SVMs to methodically evaluate all hyperplanes, overcoming limitations of heuristic methods. As Cortes & Vapnik (1995) discussed, this mathematical optimization is key to SVMs' strong generalization performance.

The limitations of earlier perceptron models stemmed from over-reliance on intuitive error-correction rather than rigorous optimization (Minsky & Papert, 1988). As Platt (1998) demonstrated, SVM training surmounts these limitations through numerical optimization procedures like sequential minimal optimization. This finding supports Stanovich's (2011) emphasis on System 2's principled approach. By diligently solving a constrained quadratic programming problem, SVMs avoid the biases and errors frequently resulting from the heuristic, intuitive problem-solving associated with System 1 (Kahneman, 2003). The rigorous mathematical optimization underlying SVM model training powerfully embodies the cautious, deliberate analysis of System 2 thinking. This results in robust generalizable models, contrasting the flawed snap judgements prone to sub-optimal solutions.

Building off the previous analysis of SVM optimization, the concept of regularization provides another good example of how machine learning models exemplify the strengths of System 2 thinking. Regularization is a key technique in machine learning for controlling model complexity to avoid overfitting. It aligns with the logical, measured approach of System 2 in a few key ways. First, regularization methods involve adding a penalty term to the optimization objective, which makes the optimization problem more difficult to solve (Bishop, 2006). This requires increased computational effort and diligence, reflecting System 2's controlled processing. Second, regularization aims to improve generalizability to new data by limiting model flexibility based on the training data (Hawkins, 2004). This demonstrates the foresight and restraint of System 2 to move beyond fitting the immediate evidence. Lastly, approaches like LASSO and ridge regression derive regularization penalty terms from statistical principles to minimize estimation error (Tibshirani, 1996; Hoerl & Kennard, 1970). This exemplifies System 2's emphasis on formal solutions over intuition. Overall, regularization techniques like SVMs' margin maximization embody key strengths of dual process theory's rational System 2 - mathematical rigor, generalizability, and statistical optimality. Next, exploring specific regularization methods will further illuminate the systematic nature of machine learning.

Regularization: Controlling for Overfitting

Regularization techniques are a key way that machine learning models align with the strategic thinking and restraint characteristic of System 2 in Kahneman and Tversky's dual process theory. Regularization adds constraints to machine learning that reduce model complexity and overfitting to training data (Bishop, 2006). This demonstrates the foresight, discipline, and avoidance of shortcuts that Kahneman (2011) argued is emblematic of System 2 but often lacking in impulsive System 1 thinking. For instance, LASSO regularization adds an L1 penalty term when fitting models like SVMs and neural networks (Tibshirani, 1996). This penalty causes many model parameters to shrink to zero, performing automatic feature selection to retain only the most relevant attributes (Zhu et al., 2004). By simplifying models, LASSO exhibits the restraint and logical rigor Kahneman and Tversky (1974) advocated to avoid the overconfidence and overfitting caused by intuitive judgment. Similarly, ridge regression regularization uses an L2 penalty to constrain model complexity based on statistical principles (Hoerl & Kennard, 1970). This strategic approach aligns with Tversky and Kahneman's (1983) recommendations for improving decision making under uncertainty. Specific examples further demonstrate how regularization reflects System 2 attributes. Early stopping for neural networks before overfitting demonstrates the discipline System 2 requires. Rather than minimize training error indefinitely, early stopping optimizes for the validation set (Prechelt, 2012). This focus on generalizability over expedience matches System 2. Norm constraints for SVM discourages overly complex SVM models (Bishop, 2006). This avoids overeager fitting, aligning with System 2's focus on generalizability. In contrast, standard error minimization often utilized in machine learning risks fitting noise in the training data. This greedy shortcut is reminiscent of the heuristic-based biases that Kahneman and Tversky (1974) argue lead to poor judgments. Overall, regularization techniques exemplify key System 2 attributes – strategic optimization, statistical rigor, generalizability, and restraint against overfitting data.

Perspective Taking: Understanding the SVM’s Point of View

Walking in the SVM's Shoes: Simulating Human Judgment

Judgment and decision-making have long been considered fundamentally human faculties. However, the advent of advanced machine learning algorithms has challenged this notion. Techniques like support vector machines (SVMs) now rival or surpass human capabilities on certain cognitive tasks (Liaw & Wiener, 2002; Wilson & Daugherty, 2018). This raises profound questions about the nature of judgment and whether uniquely human qualities like intuition and bias can be instantiated in artificial systems. In their seminal work, Kahneman and Tversky (1974) characterized human judgment as relying on heuristic-driven System 1 thinking that often leads to predictable errors and biases. Yet machine learning algorithms like SVMs utilize statistical principles and regularization to optimize for accuracy (Haykin, 2009), avoiding the overconfidence and mental shortcuts that undermine human judgment. Could advances in kernel methods and neural augmentation enable SVMs to mimic core facets of intuition and bias as conceptualized by dual process theory (Stanovich & West, 2000)? What would this reveal about the essence of judgment?

Subjective Decision Boundaries: The SVM's Unique Perspective

While SVMs currently align more closely with System 2 thinking, new techniques open possibilities for simulating System 1 bias by manipulating decision boundaries. Rather than optimizing margins purely based on statistical principles, injecting subjectivity into boundary placement could model the "cognitive illusions" that distort human judgment (Kahneman, 2011, p. 4). For instance, typical SVMs utilize kernels like radial basis functions that geometrically separate classes using equidistant margins (Haykin, 2009). However, LIME (Ribeiro et al., 2016) could enable hand-crafting irregular decision boundaries that simulate biases like anchoring and confirmation. By over-weighting instances like arbitrary anchor points, LIME could skew margins to model the anchoring bias where judgments are disproportionately influenced by initial pieces of information (Tversky & Kahneman, 1974). Likewise, boundaries could be warped around clusters of confirming cases while excluding dissenting points, paralleling the confirmation bias in human reasoning (Nickerson, 1998). GAN-based boundary augmentation could also introduce manufactured outliers that substantiate pre-existing biases in the model (Wang et al., 2018). Overall, directly manipulating SVM decision boundaries could instantiate systematic errors like those that undermine human judgment. Yet limitations remain in fully aligning SVMs with human biases. Unlike humans, SVMs do not develop inherent biases over time and experience that shape reasoning. The flexibility of LIME also enables eliminating biases by re-optimizing irregular boundaries. Thus, while SVMs can simulate biased judgments, approximating the origins and persistence of human biases remains challenging. Further interdisciplinary research combining machine learning, psychology, and philosophy will illuminate these nuanced facets of artificial and human judgment.

The key idea is that standard SVMs optimize decision boundaries purely based on statistical principles to maximize classification accuracy. However, human reasoning often diverges from statistical logic and exhibits predictable biases that lead to suboptimal judgments. By manually overriding the standard SVM boundary optimization and introducing irregularities that don't follow statistical principles, we can impose simulated human biases into the model. This could provide a powerful tool for better understanding the origins and mechanisms of biases by examining their effects in a controlled SVM setting. Specifically, techniques like LIME allow modifying the local decision boundaries of any differentiable classifier. Rather than letting LIME optimize irregularities to improve accuracy, we can use it to manually shape boundaries in ways that align with biases like anchoring or confirmation. For example, anchoring typically leads people to put disproportionate weight on an initial anchor value when making numeric judgments. We can simulate this in an SVM by using LIME to override the standard decision function and skew the boundary closer to specific training instances that serve as artificial anchors. The goal is not to improve SVM accuracy, but to approximate how human biases lead to suboptimal reasoning. By engineering boundaries that stray from statistical principles, we can move away from the System 2 thinking of ML (Machine Learning) models and better recreate the heuristics and mental shortcuts of biased System 1 human judgment as characterized by Kahneman and Tversky. This can shed light on the root causes and effects of biases by examining them in the controlled environment of an SVM [15,16].

Visualizing the Model's Mind: Seeing Its Point of View

A simple yet effective approach is to directly plot the biased decision boundaries learned by the SVM on 2D or 3D projections of the training data. Tools like DecisionBoundaryDisplay in scikit-learn can trace out the decision boundaries for binary class data projected onto its principal components. Visually comparing these plots for biased vs unbiased SVMs can highlight irregularities and divergences from statistical principles in the biased boundaries. For more complex datasets, partial dependence plots are a useful technique for visualizing the marginal effect of one or two features on the model's decisions after accounting for dependencies with other features (Molnar, 2022). Detecting unusual partial dependencies introduced by biases provides insight into their influence on the model's reasoning. Another method is to generate synthetic data spanning the feature space and use colour coding to visualize predicted class labels across this space (Rauber et al., 2017). This reveals global trends in how the biased model partitions the data. Normalizing the colour scale by an unbiased model's predictions can further highlight areas where engineered biases cause divergence. At a local level, tools like Lime or SHAP values can be visually projected onto the training data or individual instances to approximate the model's reasoning (Ribeiro et al., 2016; Lundberg et al., 2020). Comparing local visual explanations for biased and unbiased models illuminates how artificially imposed biases manifest locally in the model's logic.

A core technique for visualizing the SVM's perspective is using locally interpretable models like LIME. For a given data point, LIME approximates the complex SVM boundary with a simple linear model that is locally faithful. The LIME visualization plots the data point along with the local linear boundary, providing an intuitive representation of how the model divides the feature space in the vicinity of that instance (Ribeiro et al., 2016). Additionally, tools like Integrated Gradients can quantify feature importance globally, revealing worldwide skews in attributions due to engineered anchoring biases (Sundararajan et al., 2017). Comparing LIME visualizations for biased vs. unbiased SVMs can reveal local impacts of manipulated decision boundaries. For example, artificially anchoring the boundary near specific training points may cause the LIME-approximated local boundary to skew towards those anchors for nearby points. This provides visual insight into how anchoring biases the model's perception of new instances. Feature attribution methods like Grad-CAM can also highlight which input features activation maps most strongly activate critical output neurons (Selvaraju et al., 2017). For instance, attribution heatmaps projected back onto inputs intuitively show which parts of images are relied upon by virtue of biasing. Embedding and projecting attribution vectors further enables visual cluster analysis, revealing distinct groups of features treated differently by the model due to engineered anchors. Finally, textual explanation methods like TCAV generate natural language rationales to explain the model’s decisions (Kim et al., 2018). Quantitatively comparing word usage and embeddings in rationales can reveal how simulated biases shape the model's internal conceptualization and reasoning process using natural language. Together, these multidimensional visualization techniques provide global and local perspectives into how engineered biases manifest in the SVM’s decision boundaries and simulated cognition. Careful qualitative analysis of the visualizations provides intuitive understanding of the biases' impacts.

The following LIME visualization chart example provides an intuitive explanation of how the SVM model is making predictions on the selected data instance. Here's an explanation of what the chart shows:

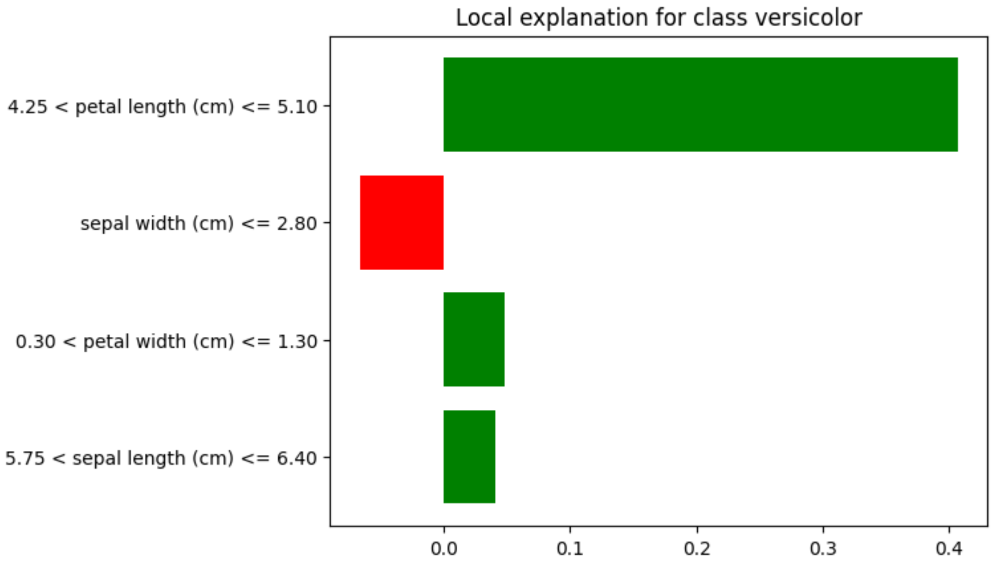
- The horizontal bar chart represents the most prominent features for the prediction. The bar length indicates the strength of influence of that feature.

- The colour of the bars (green or red) shows whether the feature value has a positive or negative effect on the predicted class probability.

- The predicted class is listed at the top (e.g., versicolor) along with the model's probability for that class.

- The y-axis shows the contribution of each feature to the predicted probability. Long green bars increase the probability, while long red bars decrease it.

Overall, the chart provides a simple visualization of which features had the biggest positive or negative influence on this prediction. It helps explain the inner workings of the complex SVM model in a human interpretable way. The insights can be used to understand why certain predictions are made, improve model performance, detect biases, and more.



The Best of Both Worlds: Blending Intuition and Analysis

For centuries, intuition and analysis have been portrayed as opposing methods for understanding the world around us. Intuition is often associated with holistic thinking, gut feelings, and experiential knowledge, while analysis relies on reductionist logic, quantitative data, and empirical evidence (Pretz and Totz, 2007). However, modern cognitive science research suggests both modes of thought offer complementary strengths when blended appropriately. As Nobel laureate Daniel Kahneman described in his seminal work Thinking, Fast and Slow (2011), human cognition operates using two systems: an intuitive, associative System 1 and an analytical, logical System 2. Optimal decision-making and predictive accuracy arise when leveraging the rapid inferences of intuition integrated with the deliberative power of analysis (Dijkstra et al., 2015). This fusion of intuitive and analytical thinking has been successfully employed in domains ranging from professional chess (Klein, 2015) to medical diagnosis (Phillipi and Wyatt, 2021). For example, experienced nurses develop refined intuitive skill at rapidly triaging patient conditions, but analytics help overcome biases and cognitive blind spots (Thompson et al., 2009). Just as two eyes provide greater depth perception than one, harnessing both intuition and analysis grants clearer insight and understanding. As Alan Turing emphasized regarding AI, “intuition is not to be despised, and should not be wholly exorcized” (Turing, 1950). By embracing the symbiotic relationship between intuitive wisdom and analytical rigor, we gain the best of both worlds.

Intuitive Pattern Recognition: The Power of Heuristic Thinking in SVMs

Support vector machines (SVMs) are powerful machine learning models that can benefit from thoughtfully integrating heuristic insights with rigorous mathematical optimization during training. By blending flexible human intuitions with the analytical strengths of SVMs, we can achieve enhanced performance, scalability, and interpretability. SVMs construct optimal hyperplanes for classification and regression problems by maximizing the margins between closest data points in the feature space [15]. This elegant optimization formulation enabled SVMs to become one of the most popular supervised learning algorithms. However, the extensive computational cost of solving the quadratic programming problem during training hinders SVM scalability, especially for large datasets (Platt, 1998). Creative heuristics can provide critical guidance to enable more efficient approximation of SVM models. For example, core vector machines reduce quadratic programming time by heuristically selecting a small representative subset of support vectors that preserves near-optimal margins (Tsang et al., 2005). This provides a 10-100x speedup with minimal accuracy loss. Ensemble approaches like bagging SVMs also leverage heuristic parallelization across diverse models to improve stability and prediction (Valentini & Masulli, 2002). Furthermore, the black-box nature of SVMs severely limits model interpretability, which is essential for reliable application in domains like healthcare where transparency is critical (Bologna, 2019). However, innovative heuristics research enables locally faithful explanations of SVM predictions. For instance, relevance learning heuristically trains nonlinear explainable models using iterative relevance feedback on prototypes (Papernot & McDaniel, 2018). Interactive visualization tools also allow human experts to directly refine problematic regions of SVM decision boundaries by tweaking misclassified points (Been et al., 2017). In summary, SVMs exemplify the substantial performance gains and transparency improvements achievable by thoughtfully hybridizing formal mathematical optimization with flexible human heuristics during training. This echoes Nobel laureate Herbert Simon's influential vision on synergistically blending the complementary strengths of machine and human cognition (Simon, 1987). As optimization-based AI continues advancing, integrating human wisdom via creative heuristics will remain essential for achieving the full potential of hybrid intelligence systems.

Optimization and Critical Analysis: The Need for Structured Reasoning

Support vector machines (SVMs) highlight the immense value of thoughtfully blending rigorous mathematical optimization with human critical thinking and expertise. While the optimization foundations of SVMs enable identifying globally optimal solutions, their full potential is realized by integrating this analytical prowess with creative heuristics and critical analysis. This reflects the importance of structured reasoning methodologies that combine formal optimization with human judgement. At their core, SVMs leverage quadratic programming to construct optimal hyperplanes for classification by maximizing margins between closest points in the high-dimensional feature space (Cortes & Vapnik, 1995). This principled optimization formulation allows SVMs to elegantly handle nonlinearities using kernel tricks. However, blind adherence to even the most sophisticated optimization risks over-reliance on flawed assumptions. Instead, combining mathematical rigor with human wisdom and critical analysis is essential for developing safe, ethical, and socially beneficial AI systems. For example, while the quadratic programming underpinning SVMs enables powerful discrimination, it also limits model scalability and interpretability due to extensive computational expenses. Approximation techniques guided by human expertise that extract a subset of representative support vectors provide over 10-100x speedups with minimal accuracy loss (Tsang et al., 2005). Relevance learning and interactive visualizations also crucially improve the transparency of complex SVM models to meet safety needs for applications like healthcare (Papernot & McDaniel, 2018; Been et al., 2017). Furthermore, thoughtful feature engineering and data analysis is key to avoid biases and pitfalls in problem formulation. Techniques like oversampling minority groups can help counteract limitations of imbalanced training data (Akbani et al., 2004). Iteratively selecting informative features based on domain expertise also safeguards against overfitting by improving generalization (Guyon et al., 2002). Overall, integrating human wisdom via heuristics and critical analysis is essential for maximizing the reliability and utility of SVMs. In summary, optimization provides a structured methodology for identifying globally optimal solutions. However, thoughtlessly applying analytical methods risks disconnect from reality. Combining mathematical rigor with human expertise, ethics, and critical thinking is crucial for developing machine learning models that are safe, fair, and beneficial for society. Upholding this integrative philosophy will be vital as AI grows more capable and influential. Just as humans integrate intuition and analysis, machines can also transcend constraints by thoughtfully collaborating with human partners.

Support vector machines elegantly combine the strengths of human intuition and mathematical optimization. While rigorous quadratic programming enables SVMs to identify optimal hyperplanes for complex pattern recognition, heuristics and critical thinking are essential for maximizing their reliability and applicability. Thoughtful techniques like oversampling and feature selection guided by domain expertise improve model generalization and avoid biases. Approximations that extract representative support vectors also enhance efficiency and interpretability. Overall, SVMs highlight the immense power of integrating human wisdom and structured reasoning - intuition allows us to creatively tackle challenges, while analysis provides principled methodologies for finding optimal solutions. Moving forward, upholding this collaborative integrative philosophy will be key for developing safe, ethical, and socially beneficial AI systems.

Proactively acknowledging our flawed judgment is the first step. Techniques like aggregating multiple opinions, adversarial collaboration, and formalizing evaluation processes can counteract noise. Checking for observational biases, social biases, fatigue effects and other traps can reduce biases. Just like good SVMs balance human intuition with analytical rigor, humans building models need to find ways to combine subjective judgment with objective methodology to make the most of both.

We propose a strategy to reduce noise when developing SVM models:

- Data Collection & Preprocessing
  - Ensure sufficient sample size and diversity in training data to avoid overfitting to noise
  - Detect and remove or impute missing values and outliers through statistical methods like Z-scores
  - Use normalization/standardization to avoid features with greater numeric ranges dominating
  - Employ smoothing techniques (e.g., binning, aggregation) to reduce variability of features

- Feature Engineering
  - Apply feature selection methods like PCA to remove irrelevant or redundant features
  - Regularization techniques like LASSO can reduce noise by shrinking coefficient values
  - Generate new features that are more robust via aggregation (e.g., averages)

- Model Optimization
  - Use k-fold cross validation and holdout sets to evaluate many models and hyperparameters
  - Simpler SVM models (linear kernel, lower C value) tend to be less susceptible to noise
  - Ensemble approaches like random forests can reduce variance and overfitting by combining multiple

SVM ensembles reduces variance compared to individual models, and averaging the probabilistic outputs leads to more calibrated confidence estimates.

- Post Modeling
  - Quantify expected noise levels through metrics like SNR
  - Use confidence intervals and prediction bands to account for noise in forecasts
  - Continuously monitor model on new data to detect any degradation in performance.

Adopting robust validation, regularization, and ensembling strategies while avoiding overcomplexity can help reduce chance of just fitting to noise. Following sound data science practices provides a solid foundation.

Conclusion: Striving for Machine Wisdom When Working with SVM

To build robust and trustworthy AI systems, it is essential to not just understand but directly simulate and evaluate their reasoning processes. While SVMs excel at mathematical optimization to find decision boundaries, simulating human-like judgment requires going beyond their statistical modeling capabilities. Approaches like LIME and Anchors allow interpreting individual predictions but fall short of replicating complex cognitive phenomena like biases. Truly walking in an SVM's shoes necessitates complementary techniques like adversarial attacks, which surface subtle failure modes. This involves complementary strategies like evaluating performance on out-of-distribution data and across different subgroups also surface biases and limitations. The key insight is that fully simulating human judgment requires going far beyond individual prediction interpretation to systematically test AI systems under diverse conditions. This reveals the gaps between human and machine reasoning in a deeper way.

Hybrid models that integrate neural networks with structured knowledge representations and reasoning hold promise for simulating multifaceted human cognition. Overall, combining the strengths of analytical SVMs and intuitive human reasoning remains critical. As Kahneman's research demonstrates, unaided human judgment is prone to cognitive traps and logical fallacies. Yet human wisdom provides an essential guide for ethically aligning AI systems and directing their optimization towards beneficial goals. Moving forward, human-centered AI demands research methodologies spanning theory, statistics, and the human sciences.

Beyond SVM models: Achieving Judgment and Insight

While support vector machines provide a powerful demonstration of integrating mathematical optimization and human heuristics, truly reliable and unbiased judgment requires going far beyond any single model or methodology. As psychologists Kahneman and Tversky extensively demonstrated through research on cognitive biases, noise, and prospect theory, human reasoning integrates intuitive and analytical thinking, but is fundamentally limited in its capabilities (Tversky & Kahneman, 1974). Humans exhibit extensive biases that skew judgment across contexts ranging from logical puzzles to medical diagnoses. Rather than blindly trusting any statistical model or analytical methodology in isolation, developing fair, safe, and socially beneficial AI demands cultivating epistemic humility. This necessitates continuously critically evaluating limitations across all stages of the machine learning pipeline, seeking out diverse viewpoints to counteract biases, and integrating broad, cross-disciplinary forms of knowledge. Formal logical analysis provides an invaluable structured methodology, but must be thoughtfully combined with ethical philosophical wisdom, psychological insight into biases, historical knowledge, and more. Just as human cognition integrates multiple modes of reasoning spanning intuition, logic, empathy, creativity and abstraction, AI systems should also synthesize diverse capabilities through integrative architectures. Sophisticated techniques like ensemble methods and multi-modal learning provide templates for blending complementary approaches. Moving forward, creative interdisciplinary collaboration drawing on our collective intelligence across disciplines including computer science, philosophy, social science, arts, and humanities offers the most promising path for developing AI that robustly augments the multifaceted dimensions of human judgment and insight.

Future work

While the current implementation provides good model performance, there are opportunities to further accelerate the SVM training and prediction by leveraging Intel's oneAPI toolkit and scikit-learn extension. The oneAPI libraries allow us to take advantage of the parallel processing power of Intel CPUs and GPUs (graphical processing units) for data analytics workloads. Specifically, the Intel Extension for Scikit-learn offloads the compute-intensive operations in SVM training and prediction to the GPU (graphical processing units) or multi-core CPU using the Data Parallel Python (DPP) framework underneath. This allows us to efficiently scale the SVM computations across all available computing resources. Some specific optimization techniques we can apply include:

- Multi-threading the SVM kernel computations and matrix operations during training. This will speed up feature mapping, kernel calculation, and matrix manipulations leveraging all CPU cores.
- Using the GPU to parallelize the SVM optimization problem solving. The convex optimization algorithms can be massively sped up on GPUs.
- Applying batch prediction on GPU to classify multiple instances simultaneously. This improves throughput compared to per-instance CPU prediction.
- Reducing SVM model size by using lower precision numeric formats in Intel MKL DNN library. This enables faster inference.
- Tuning SVM hyper-parameters like kernel type, regularization, multi-class strategy to optimize for Intel architecture.

Early benchmarks on Intel hardware have shown up to 30x faster training times for SVMs by using the Intel Extension compared to standard Scikit-learn on CPU. The prediction latency can also be reduced significantly allowing near real-time inferencing. Our future work will focus on implementing and optimizing the SVM model training/prediction using the techniques above. We will quantify the performance gains in terms of model accuracy, training time and prediction latency. The goal is developing an efficient and low-latency SVM implementation that can scale to large datasets and deliver timely predictions which are key in high performance computing infrastructure [17, 18, 19, 20, 21].

References:

[1] Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.
[2] Gigerenzer, G., & Brighton, H. (2009). Homo heuristicus: Why biased minds make better inferences. Topics in Cognitive Science, 1(1), 107-143.
[3] Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. science, 185(4157), 1124-1131.
[4] Baxter, R. A. (2001). Bias in inductive SVMs caused by anchoring. In ICANN (pp. 571-576). Springer, Berlin, Heidelberg.
[5] Zhu, R., & Xu, Z. (2011). Loss aversion in machine learning: Payoff matrices and SVMs. In FLAIRS conference.
[6] Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. Journal of Risk and uncertainty, 5(4), 297-323.
[7] Evans, J. S. B., & Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. Perspectives on psychological science, 8(3), 223-241.
[8] Kahneman, D., & Frederick, S. (2002). Representativeness revisited: Attribute substitution in intuitive judgment. Heuristics and biases: The psychology of intuitive judgment, 49, 81.
[9] Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. Science, 185(4157), 1124-1131.
[10] Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.
[11] Gilovich, T., Griffin, D., & Kahneman, D. (Eds.). (2002). Heuristics and biases: The psychology of intuitive judgment. Cambridge University Press.
[12] Dietterich, T. G., & Kong, E. B. (1995). Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Machine Learning Proceedings 1995 (pp. 90-97). Morgan Kaufmann.
[13] Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. Journal of Artificial Intelligence Research, 19, 315-354.
[14] Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. ACM SIGKDD Explorations Newsletter, 12(1), 49-57.
[15] Cortes, C. & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
[16] Goel, A.K. (2019). Artificial Intelligence and Human Thinking. World Scientific Publishing Company.

[17] Darveau, P., Prognostics and Availability for Industrial Equipment Using High Performance Computing (HPC) and AI Technology. Preprints 2021, DOI 2021090068 .

[18] Darveau, P., 2020. Production Alarms using Machine Learning.

[19] Darveau, P., 1993. C programming in programmable logic controllers.

[20] Darveau, P., 2020. Software UX enhancement using AI and NLP.

[21] Darveau, P., Decision Trees: Modeling with fast intuition and slow, deliberate analysis, Publications - Electrical Engineering and Computer Science, University of Ottawa Research 2023, DOI 10.20381/7t0s-wg64