MACHINE LEARNING IN WIRELESS SENSOR NETWORKS

Clay McLeod

October 2015

Paper

M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies and Applications," CoRR, vol. abs/1405.4463, 2014. [Online]. Available: http://arxiv.org/pdf/1405.4463v2.pdf

Viewable at `http://bit.ly/wsn-ml`

1. Brief introduction to Machine Learning
2. Applications of ML to Wireless Sensor Networks.
3. Drawbacks of using ML in Wireless Sensor Networks.
4. Two concrete solutions where ML is applied.

MACHINE LEARNING

Summary of ML

1. The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems.

Summary of ML

1. The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems.

2. The adoption of computational methods for improving machine performance by detecting and describing consistencies and patterns in training data.

Summary of ML

1. The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems.

2. The adoption of computational methods for improving machine performance by detecting and describing consistencies and patterns in training data.

Summary: exploit historical data to improve WSN performance at a given task

Supervised ML

Uses labelled data to build a predictive model (answer is known).

Supervised ML

Uses labelled data to build a predictive model (answer is known).

Examples

1. k-NN: Querying processing subsystem
2. Decision Tree: Assess link reliability (loss rate, corruption rate, mean time to failure, mean time to restore)
3. Neural networks: Any number of applications with high complexity (Big data tuning of parameters and dimensionality reduction)
4. Support vector machines: Security and localization
5. Bayesian statistics: Assessing event consistency, investigating knowledge about environment

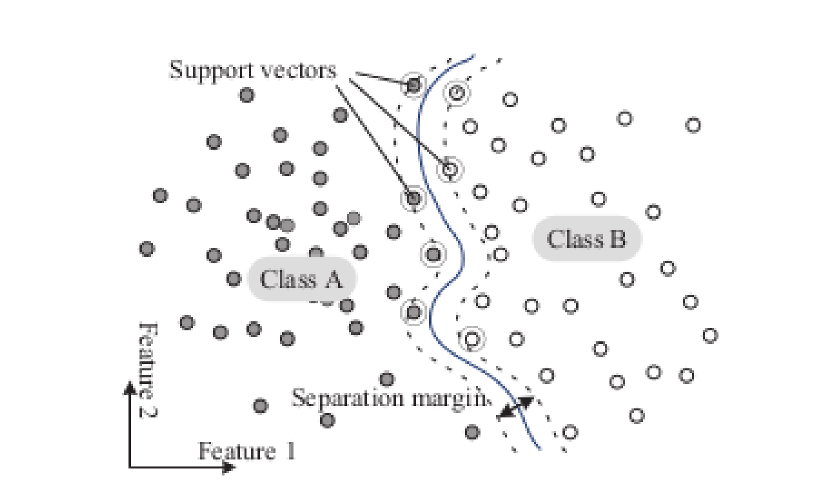

Figure 1: Non-linear support vector machine

Unsupervised ML

Tries to find underlying patterns in data (answer not known).

Unsupervised ML

Tries to find underlying patterns in data (answer not known).

Examples

1. K-means clustering: Clustering of nodes within a WSN
2. Principle component analysis (PCA): Reduce amount of transmitted data through dimensionality reduction.

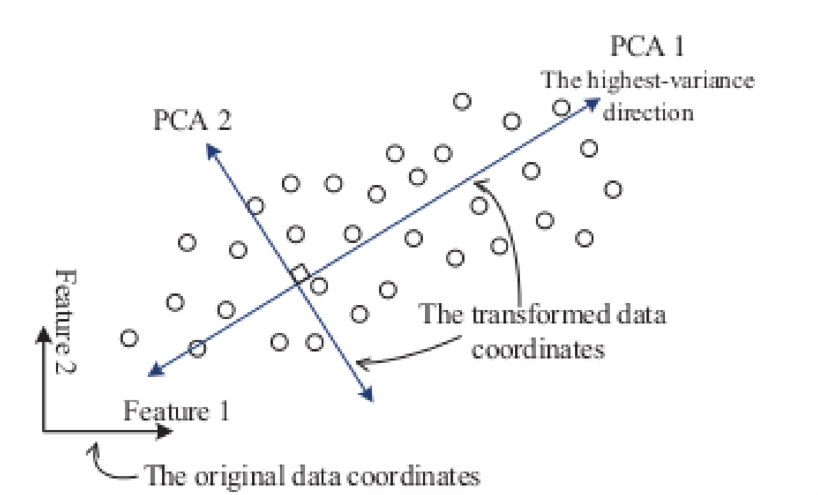

Figure 2: Principal component analysis

Reinforcement ML

Maximizes long-term rewards by using its own experience (find answers along the way).

Reinforcement ML

Maximizes long-term rewards by using its own experience (find answers along the way).

Example:

1. Q-Learning: Routing optimization problems (multicast routing, geographic routing, feedback routing, Q-probabilistic routing).

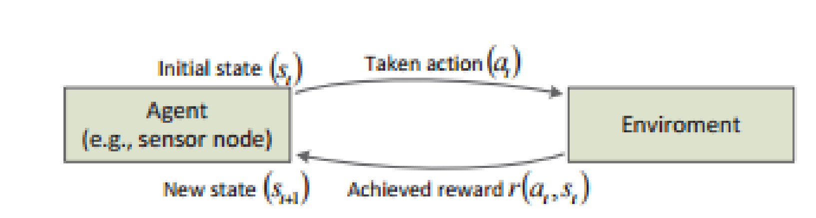

Figure 3: Visualization of Qlearning

APPLICATIONS

- Monitor dynamic environments that change rapidly over time. (Ex. monitor a node's geographical location).

APPLICATIONS

- Monitor dynamic environments that change rapidly over time. (Ex. monitor a node's geographical location).
- Calibrate a WSN to new knowledge about the environment (Ex. volcano eruption and waste water monitoring).

- Monitor dynamic environments that change rapidly over time. (Ex. monitor a node's geographical location).
- Calibrate a WSN to new knowledge about the environment (Ex. volcano eruption and waste water monitoring).
- Modelling systems with high complexities/dimensions that make it difficult for mathematicians to model.

- Monitor dynamic environments that change rapidly over time. (Ex. monitor a node's geographical location).
- Calibrate a WSN to new knowledge about the environment (Ex. volcano eruption and waste water monitoring).
- Modelling systems with high complexities/dimensions that make it difficult for mathematicians to model.
- Extract important correlations between sensor data to propose further improvements to the system.

- Monitor dynamic environments that change rapidly over time. (Ex. monitor a node's geographical location).
- Calibrate a WSN to new knowledge about the environment (Ex. volcano eruption and waste water monitoring).
- Modelling systems with high complexities/dimensions that make it difficult for mathematicians to model.
- Extract important correlations between sensor data to propose further improvements to the system.
- Intelligent decision making and autonomous control.

DETRIMENTS

DETRIMENTS

- *Power consumption*: Many ML techniques are computationally expensive. Utilizing them in a wireless sensor network could have negative effects depending on how often the calculations are run.

- *Power consumption*: Many ML techniques are computationally expensive. Utilizing them in a wireless sensor network could have negative effects depending on how often the calculations are run.
- *Data purity*: Generally speaking, ML perform better as the amount of data you have increases. Since WSNs are generally deployed in unpredictable environments, we cannot be sure what data we have coming into the system (except in supervised learning).

BRIEF NOTE ON BAYESIAN STATISTICS

Process

- Model inputs/outputs as random variables, build model based on the distributions

Process

- Model inputs/outputs as random variables, build model based on the distributions
- Update bayesian model using existing data (prior)

Process

- Model inputs/outputs as random variables, build model based on the distributions
- Update bayesian model using existing data (prior)
- Run thousands of random simulations to determine true distrobution of inputs/outputs.

Benefits for WSNs

1. Very fast execution, straightforward implementation

Benefits for WSNs

1. Very fast execution, straightforward implementation

2. Low power consumption

Benefits for WSNs

1. Very fast execution, straightforward implementation
2. Low power consumption
3. **Works well on small amounts of sparse data/small amount of data**

Benefits for WSNs

1. Very fast execution, straightforward implementation
2. Low power consumption
3. **Works well on small amounts of sparse data/small amount of data**
4. Output is generally probabilities (except in ML paradigms, like Bayes network).

SOLUTIONS

Routing in WSNs

Routing protocols	Topology	Machine learning algorithm(s)	Overhead	Scalability	Delay	Distributed / centralized	QoS
Distributed regression [69]	Flat / multi-hop	kernel linear regression	Low	Limited	High	Distributed	No
SIR [70]	Flat / multi-hop	SOM	High	Limited	Moderate	Hybrid	Yes
Q-MAP multicast [65]	Flat /multi-hop	Q-learning	Low	Moderate	High	Distributed	No
RLGR [66]	Hierarchical / geographic routing	Q-learning	Low	Good	Low	Distributed	No
Q-Probabilistic [68]	Flat / geographic routing	Q-learning	Low	Limited	High	Distributed	Yes
FROMS [67]	Flat - multi-hop	Q-learning	High	Limited	Moderate	Distributed	No

Figure 4: Overview of ML routing techniques

Route Optimization

Solution: Q-Probabilistic Routing

QoS enabled routing protocol that uses reinforcement learning (specifically, Q-learning) and a Bayesian decision model to route packets.

1. Each node maintains a simple lookup table containing information on neighbors expected delivery rate (based on previous actions) and power constraints.

2. Bayesian model uses information in this lookup table, the importance of the message, and other metadata about a node (such as activity rate) to asses which node has the highest probability of giving us the fastest transmission time.

3. Node sends the packet and periodically communicates with its neighbor node to find out delivery times between the two nodes.

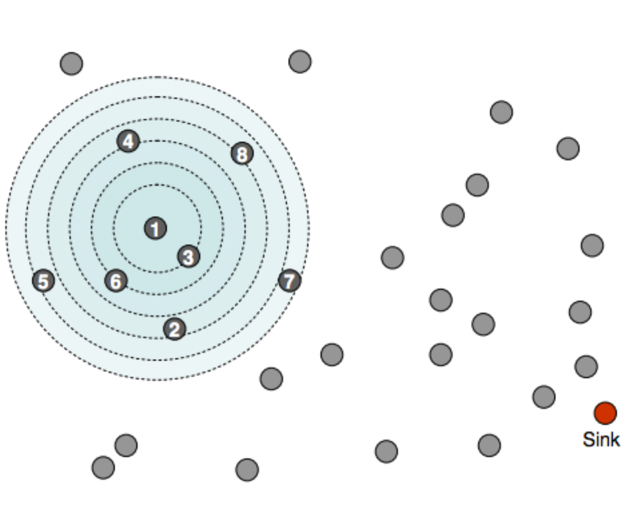

Figure 5: QPR visualization of nodes

	$N_j \in \phi_1^+$			
Cost metric estimation for N_1	N_2	N_3	N_7	N_8
Distance to destination: $\|\mathbf{z}_j - \mathbf{z}_0\|$	7	8	5	6
ETX to Destination: $Q_1(j)$	2.5	5.2	2.1	4.1
ETX/Progress: $\frac{Q_1(j)}{\|\mathbf{z}_1 - \mathbf{z}_0\| - \|\mathbf{z}_j - \mathbf{z}_0\|}$	1.25	5.2	0.52	1.37

Figure 6: QPR visualization of table

Security Intrusion Detection

Solution: Distributed Bayesian Belief Network

Idea is simple:

1. Given that the majority of nodes will have similar sensor readings, each node constantly monitors each of its neighbor nodes and builds a model of what an "average" node should look like (at each iteration).

2. This becomes the prior, causing the BBN to infer conditional relationships in the data.

3. Outlier detection is as trivial as setting a probability threshold for an outlier and communicating that to some centralized datastore.

Bonus - Quorum decision function: have each of the neighbor nodes vote on whether or not they think a certain node is an outlier. This prevents local outliers from being reported.

FUTURE APPLICATIONS

- *Compressive Sensing and Sparse Coding*: data compression and dimensionality reduction to reduce communications across network, saving energy.
- *Distributed and Adaptive ML Techniques*: Distributed learning methods use less energy than centralized due to less communication. Spreading out computations across nodes will spread out energy consumption across the WSN, leading to longer overall life.
- *Resource Management*: Using ML to cease "nonfunctional and energy wasteful" activities, such as looping of packets.
- *Detect Spatial and Temporal Correlations*: Cluster based on temporal and spatial conditional correlations, decreasing communications and saving energy.

QUESTIONS?