Principal Component Analysis Preprocessing with Bayesian Networks for Battery Capacity Estimation

Liessman E. Sturlaugson, John W. Sheppard
Department of Computer Science
Montana State University
Bozeman, MT 59717
{liessman.sturlaugson, john.sheppard}@cs.montana.edu

***Abstract*—Bayesian networks (BNs) are a common data-driven approach for representing and reasoning in the presence of uncertainty. Inference in a BN can quickly become intractable as the complexity of the network increases, specifically in the number of nodes and the number of states for each node. We demonstrate the benefit of preprocessing cyclic time-series measurements using principal component analysis (PCA), evaluating the technique with the BN to perform diagnostics on a set of lithium-ion batteries that have undergone repeated charging/discharging cycles. The results show how PCA preprocessing can result in simpler Bayesian network models than those from the raw data while still achieving higher accuracy.**

I. Introduction

Sensor networks and monitoring systems can generate large amounts of measurement data very quickly. Turning these measurements into usable information, on the other hand, can be a difficult challenge [1]. For example, diagnostic models may attempt to incorporate usage information and environmental conditions as they are measured to perform fault detection and isolation. More frequent system measurements potentially gives more information, but the models often become more computationally expensive as they try to cope with the finer resolution in the data. Data preprocessing becomes important to convert the raw measurement values into more compact and more abstract information that the models can more easily handle. Dimensionality reduction techniques are applicable, because they compact the data while attempting to preserve the greatest amount of information. We examine the extent to which PCA aids the diagnostic capabilities of the BN model using the life-cycle measurements from sets of Li-ion batteries that have been run to failure.

Section II gives the necessary background information and related work. Section III describes the experimental design, including a description of the dataset, the steps taken for data preprocessing, and the design choices for network topology and parameter estimation. Section IV compares the results of the BN diagnostics with and without the PCA preprocessing. Section V presents concluding remarks.

II. Background and Related Work

In this section, we review principal component analysis (PCA), Bayesian networks (BNs), and the supervised discretization technique based on average class entropy and minimum description length (MDL). We end the section with a survey of related work.

A. Principal Component Analysis

Principal component analysis (PCA) is a popular dimensionality reduction technique that linearly projects n-dimensional data into a smaller d-dimensional coordinate system in which each successive axis is aligned in the direction of the maximum remaining variance [2].

Suppose that we have a matrix of data $\mathbf{D}$ such that the n columns correspond to the dimensions of the data and the rows correspond to sets of measurements in each dimension. We assume that the columns of $\mathbf{D}$ have been mean-centered and normalized so that $\mathbf{\Sigma} = \mathbf{D}^T\mathbf{D}$ is the covariance matrix. $\mathbf{W}$ is the $n \times n$ matrix of eigenvectors of the covariance matrix $\mathbf{\Sigma}$.

Then the transformed data matrix $\mathbf{D}'$ can be represented as the projection of $\mathbf{D}$ onto $\mathbf{W}$ as

$$\mathbf{D}' = \mathbf{D}\mathbf{W}.$$

To decrease dimensionality, we choose $d < n$ of the topmost eigenvectors (called principal components) of $\mathbf{W}$ for the projection, resulting in an approximation of the data matrix $\mathbf{D}'$ using only d dimensions.

B. Bayesian Networks

Bayesian networks are probabilistic graphical models that use nodes and arcs in a directed acyclic graph to represent a joint probability distribution over a set of variables [3]. Formally, suppose that $P(\mathbf{X})$ is a joint probability distribution over n variables $x_1, \ldots, x_n \in \mathbf{X}$, and that $Parents(x_i)$ denotes the parents of the node x_i in the network. Then the graphical structure of the network factors the joint probability distribution as,

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(x_i | Parents(x_i)).$$

The full joint probability distribution is exponential in the number of nodes and number of states for these nodes. By factoring the joint probability distribution into only the relevant variable interactions, represented by the parent-child relationships in the network, the complexity of the network can

be better managed. Nevertheless, exact and even approximate inference in general Bayesian networks is NP-hard [4]. Thus, the the number of nodes, states, and arcs in the model must be balanced to make inference tractable while maintaining a relatively accurate model. One of the ways to accomplish this is by improving the quality of the data that the model is using.

The Tree Augmented Naïve (TAN) Bayes model is a special type of Bayesian network that selects one node as the parent of all other nodes. This serves as the class node, that is, the state to be predicted, given evidence assigned to its children. Further dependencies are allowed between the children of the class node, as long as the dependencies form a tree structure from these child nodes.

C. Supervised, Entropy-Based Discretization

The Bayesian network definition presented here is a discrete-state model, meaning the nodes in the network represent conditional probability tables that define the probabilities for all the node's states conditioned on the parents' states. The data used by the models in the following experiments are real-valued voltage, current, and temperature measurements. Thus, these real-valued measurements need to be discretized into a discrete number of states capable of being represented by the model.

If the class label is known, the discretization can be done with respect to this class label, referred to as supervised discretization [5]. In particular, we use a supervised discretization technique that recursively splits intervals based on average class entropy and uses the minimum description length as the stopping criterion [6].

Suppose that a set S of $N = |S|$ instances has k classes, labeled as $C_1, C_2, \ldots C_k$. Let $P(C_i, S)$ be the proportional of samples in S having class C_i. The entropy $H(S)$ of a set S is

$$H(S) = -\sum_{i=1}^{k} P(C_i, S) \log P(C_i, S).$$

Suppose that an interval boundary T of continuous-valued attribute A partitions S into S_1 and S_2. The class information entropy of the partition induced by T, is defined as

$$E(A, T; S) = \frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2).$$

The interval boundary is chosen that minimizes $E(A, T; S)$. Each attribute is recursively partitioned into two subsets until the information gain, $H(S) - E(A, T; S)$, falls below a threshold that is based on the number of samples, the number of classes, and the class information entropy of the current partition.

D. Related Work

PCA has been used with several different types of machine learning models, such as neural networks [7], [8] and support vector machines [9]. Examples of PCA with Bayesian networks include [10] and [11].

However, used as a preprocessing step, these papers are less concerned with quantifying or analyzing the improvement that PCA can offer over just using the raw values. By using a discrete Bayesian network and supervised discretization on data with and without PCA preprocessing, we can compare the resulting number of states, one of the driving factors of Bayesian network computational complexity. Thus, while the PCA preprocessing step can be time-consuming up-front, it makes model creation and inference much more efficient. By comparing model performance with and without the PCA preprocessing, we can show that PCA still preserves and even improves model accuracy.

III. Experiments

This section performs diagnostic comparisons between the raw measurements and the values projected onto only the first principal component after the PCA preprocessing. The Structural Modeling, Inference, and Learning Engine [12] was used to construct TAN models from the data and perform model inference.

A. Dataset

The dataset used in the experiments, provided by the Prognostics Center of Excellence at NASA Ames [13], consists of a total of 34 lithium-ion batteries that underwent accelerated aging. Each battery was run under a cycle of 3 operational profiles: charge, discharge, and impedance testing. The charge/discharge cycles performed accelerated battery aging, while the impedance testing provided the damage criterion, or health estimate, of the battery. The batteries cycled through these operational profiles (although impedance testing was not performed in every cycle) until they reached their end-of-life (EOL).

During the charge cycle, the features of the dataset were battery terminal voltage, battery output current, battery temperature, current measured at charger, and voltage measured at charger. The charge cycles occurred under constant current at 1.5A until the battery voltage reached 4.2V. The charging continued under constant voltage until the charge current dropped to 20mA. The Bayesian networks were learned from these charging cycles both with and without the PCA preprocessing. Given the characteristics of each charging the cycle, the models were used to predict the capacity during the subsequent discharge cycle.

During the discharge cycle, the features of the dataset were battery terminal voltage, battery output current, battery temperature (as above), current measured at load, and voltage measured at load. The capacity value of each discharging cycle was derived from these measurements and was used as the health estimate to be predicted by the models. The sets of batteries were discharged under different profiles that varied the current loads and the voltage thresholds. Thus, different Bayesian networks were learned for each battery set.

Because in our scenario, diagnosis occurs at the charger, the discharging data is not used and so the discharging profiles are

TABLE I
NUMBER OF BINS FROM MDL DISCRETIZATION ON BATTERY SETS, BEFORE AND AFTER PCA ON CHARGING CYCLES

		Median		Average	
	Variables	Before	After	Before	After
2-Bin Capacity	Cycle	8	2	28.6	2.2
	Input Current	33	3	36.2	2.9
	Input Voltage	122	2	99.0	2.9
	Temperature	27	2	31.9	2.6
	Current at load	350	3	304.7	3.4
	Voltage at load	134	2	142.3	2.6
3-Bin Capacity	Cycle	15	2	34.7	2.8
	Input Current	39	3	45.0	3.3
	Input Voltage	206	3	181.0	3.3
	Temperature	30	3	42.6	2.9
	Current at load	602	3	613.7	3.6
	Voltage at load	192	3	191.1	3.0
4-Bin Capacity	Cycle	17	3	43.6	3.2
	Input Current	44	3	49.6	3.3
	Input Voltage	209	4	208.0	4.2
	Temperature	47	2	49.6	3.0
	Current at load	760	4	687.7	4.4
	Voltage at load	247	2	235.4	3.7
5-Bin Capacity	Cycle	21	2	44.4	3.1
	Input Current	50	3	54.1	3.6
	Input Voltage	228	4	233.0	3.9
	Temperature	50	2	54.1	2.8
	Current at load	836	4	846.2	4.1
	Voltage at load	254	3	264.1	3.3

not listed. The BN should be able to detect degradation during charging regardless of the battery's discharging profile.

During the impedance testing, the features of the dataset included the current in sense branch, current in battery branch, battery impedance, estimated electrolyte resistance, and estimated charge transfer resistance. However, the data from the impedance cycles were not used in the models.

B. Discretization

The health variable (capacity) was first discretized by equal-width binning into 2, 3, 4, and 5 bins. For each of these number of bins, the remaining variables then underwent supervised discretization using the entropy-based binning described above.

The resulting bins influenced both model accuracy and the tractability of inference. In general, as discretization becomes more granular, models may be able to achieve better accuracy. However, the models run a risk of "memorizing" the data, mapping a small set of unique real values to specific class labels, and losing the ability to generalize across larger intervals. Furthermore, the data also becomes more sparse, because there are fewer examples for each discretization. Lastly, the number of states in the models (corresponding to the number of bins) also control the computational complexity of running inference on the model. The more states, the higher the computational burden becomes.

TABLE II
AVERAGE ACCURACY OF THE TWO BAYESIAN NETWORK CLASSIFIERS WITH AND WITHOUT PCA PREPROCESSING

	Battery Set IDs	**Without PCA**	**With PCA**
2-Bin Capacity	{5-7, 18}	88.6%	99.7%
	{25-28}	99.2%	99.2%
	{29-32}	92.9%	90.4%
	{33,34,36}	89.6%	91.3%
	{38-40}	90.6%	98.6%
	{41-44}	97.2%	98.0%
	{45-48}	77.5%	95.8%
	{49-52}	57.3%	64.6%
	{53-56}	99.3%	99.3%
3-Bin Capacity	{5-7, 18}	73.1%	97.1%
	{25-28}	98.3%	98.3%
	{29-32}	93.6%	84.6%
	{33,34,36}	96.6%	82.8%
	{38-40}	98.6%	97.1%
	{41-44}	99.0%	98.7%
	{45-48}	71.8%	91.2%
	{49-52}	92.9%	85.6%
	{53-56}	92.7%	92.7%
4-Bin Capacity	{5-7, 18}	69.7%	95.9%
	{25-28}	96.7%	96.7%
	{29-32}	87.8%	78.8%
	{33,34,36}	88.9%	92.2%
	{38-40}	100%	96.4%
	{41-44}	96.3%	99.0%
	{45-48}	67.6%	88.4%
	{49-52}	90.2%	55.8%
	{53-56}	73.8%	73.8%
5-Bin Capacity	{5-7, 18}	70.3%	94.8%
	{25-28}	90.8%	90.8%
	{29-32}	73.1%	80.1%
	{33,34,36}	85.9%	71.3%
	{38-40}	97.1%	95.7%
	{41-44}	96.3%	97.6%
	{45-48}	85.6%	82.7%
	{49-52}	84.0%	76.7%
	{53-56}	15.5%	10.2%

IV. RESULTS

Two Bayesian network classifiers were tested. The first used the raw discharging measurements, without PCA preprocessing. The second used the preprocessed data, in which PCA was performed on the discharging cycles aggregated across all battery datasets, and each cycle was projected onto the first principal component. The two datasets then underwent their own supervised discretization.

Table I shows the number of bins for each variable after MDL discretization, averaged across the sets of batteries, before and after PCA was performed on the raw measurements. As can be seen from the table, the PCA preprocessing significantly reduced the number of bins returned by the supervised discretization for all variables. This is because PCA

extracted the most important features from the data, resulting in fewer bins when discretizing the measurements with respect to the capacity state.

However, the goal of the discretization is to create Bayesian network classifiers to accurately predict a battery's capacity. Thus, the number of bins, while impacting the computational complexity of the models, still need to be sufficient to allow the models to achieve reasonable accuracy.

To keep the accuracy results comparable between the two models, the number of bins of the raw data was constrained to match that of the PCA data. This was to test whether the PCA data was able to train more accurate models than the raw data, given similar complexity of each dataset's networks. After the supervised discretization was performed on the raw data, the bins went through a merging process. Adjacent bins were merged when the merge resulted in the least increase in class information entropy. This process was repeated for each variable and for each battery set until the number of bins matched that the corresponding number of bins of the discretized PCA data.

After structure learning and parameter estimation, the models were tested on each charging cycle of each battery set. Given the six observed variables of the charging cycles, the models were to predict the state of the capacity during the subsequent discharging cycle. For the model training from raw data, the model performed inference on each set of measurements in the discharging cycle and the most-predicted state over the cycle was used as the prediction. Table II shows the accuracy of the two classifiers across the sets of batteries. The model with the PCA data tended to outperform the model with the raw data on the majority of battery sets.

V. Conclusion

This paper shows how PCA preprocessing can increase the tractability of Bayesian networks without degrading classification accuracy by applying the PCA preprocessing technique to estimate Li-ion battery capacity using measurements of the battery discharge cycles.

Although preprocessing the data with PCA may entail extra computation when training the networks, the resulting models can have a better trade-off between complexity and accuracy when performing inference.

References

[1] A. Jacobs, "The pathologies of big data," *Commun. ACM*, vol. 52, no. 8, pp. 36–44, Aug. 2009.

[2] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2010.

[3] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[4] G. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial intelligence*, vol. 42, no. 2, pp. 393–405, 1990.

[5] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," in *International Conference on Machine Learning*, 1995, pp. 194–202.

[6] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," 2004.

[7] C. Li, P. Heinemann, and R. Sherry, "Neural network and Bayesian network fusion models to fuse electronic nose and surface acoustic wave sensor data for apple defect detection," *Sensors and Actuators B: Chemical*, vol. 125, no. 1, pp. 301–310, 2007.

[8] F. Aminian and M. Aminian, "Fault diagnosis of analog circuits using Bayesian neural networks with wavelet transform as preprocessor," *Journal of electronic testing*, vol. 17, no. 1, pp. 29–36, 2001.

[9] A. Widodo and B. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[10] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 124 –132, march 2006.

[11] N. Nariai, S. Kim, S. Imoto, and S. Miyano, "Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks," in *Pacific Symposium on Biocomputing (PSB03)*, 2003, pp. 336–347.

[12] M. J. Druzdzel, "SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a development environment for graphical decision-theoretic models," in *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference*, ser. AAAI '99/IAAI '99. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1999, pp. 902–903. [Online]. Available: http://dl.acm.org/citation.cfm?id=315149.315504

[13] B. Saha and K. Goebel, "Battery Data Set. NASA Ames Prognostics Data Repository, [http://ti.arc.nasa.gov/project/prognostic-data-repository], NASA Ames, Moffett Field, CA," 2007.