Support Vector Machines for Prostate Lesion Classification in the ProstateX Challenge

Andy Kitchen[a] and Jarrel Seah[b]

[a]Silverpond Pty. Ltd., 2/382 Little Collins Street, Melbourne VIC 3000, Australia
[b]STAT Innovations Pty. Ltd., PO Box 274, Ashburton VIC 3147, Australia

ABSTRACT

Support vector machines (SVM) are applied to the problem of prostate lesion classification for the SPIE ProstateX Challenge 2016, achieving a score of 0.82 AUC on held-out test data. Square 5mm transverse image patches are extracted around each lesion center from aligned MRI scans. Three MRI modalities are simultaneously analyzed: T2-weighted, apparent diffusion coefficient (ADC) and volume transfer constant (K^{trans}). Extracted patches are used to train a binary classifier to predict clinical significance. The machine learning algorithm is trained on 76 positive cases and 254 negative cases (330 total) from the challenge. The method is conceptually simple, trains in a few seconds and yields competitive results.

Keywords: ProstateX, MRI, prostate, support vector machine, machine learning, computer-aided diagnosis

1. INTRODUCTION

The Support Vector Machine[1] (SVM) is applied to the problem of prostate lesion classification for the SPIE ProstateX Challenge 2016. This machine learning algorithm is trained on MRI scans to classify new unseen lesions as clinically significant or not. The method described achieves competitive performance with a score of 0.82 area under the curve (AUC) assessed on held-out test data with correct answers kept hidden by competition organizers. This approach is conceptually simple. There are no involved processing steps or complex computer vision algorithms. The method is efficient, due to its simplicity and the availability of highly optimized implementations of SVMs. The implementation uses free and open source software tools and libraries, so there are no encumbrances to further research or reproduction; all code is extensible and auditable.

The data provided for the challenge is a collection of patients each with MRI images in multiple modalities and metadata provided as comma separated value (CSV) files. For each patient one or more prostate lesions and their locations have been identified. For patients in the training set, each lesion is labeled with its clinical significance (true or false). The task is to predict the hidden labels for the patients in the test set. Three MRI modalities are simultaneously analyzed by this method: T2-weighted, apparent diffusion coefficient (ADC) and volume transfer constant (K^{trans}); which are all shown to be related to clinical significance.[2] These modalities are all aligned and processed together.

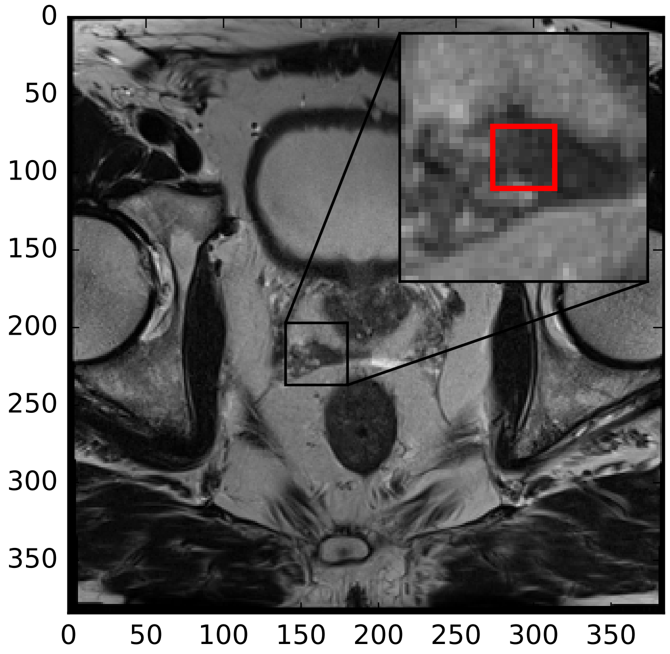
2. METHOD

2.1 Preprocessing

The ProstateX input data is complex, including multiple data structures and formats. Processing and reconciling inconsistent or overlapping metadata is a major challenge. Multiple rules were applied on each patient to heuristically extract the most relevant T2, ADC and K^{trans} images. Metadata was collated from both Comma Separated Value (CSV) files and information embedded within image files themselves. Significant effort was expended in this area, rivaling that spent on machine learning models.

Further author information:
Andy Kitchen: E-mail: andy.kitchen@silverpond.com.au, Telephone: +61 432 514 287
Jarrel Seah: E-mail: jarrelscy@gmail.com, Telephone: +61 450 681 551

Figure 1. T2-weighted MRI image with red rectangle showing patch area with surrounding lesion



2.2 Patch Extraction

For each patient, and each lesion, a centered 5mm × 5mm patch is extracted at a resolution of 1px/mm (see figure 1) in 3 modalities T2, ADC and K^{trans}. Only transverse image slices are used. Images are flattened into 75 ($5 \times 5 \times 3$) dimensional vectors. We further augment this vector with zone information by encoding it as dummy variables[3] which are then concatenated with the image vector. See table 1 for performance comparison without zone information.

Patch extraction subroutines were validated using pixel-by-pixel comparison to the reference images released by the challenge organizers and were found to be in close agreement. All processing is internally carried out with 32-bit floating point pixel values. This preserves large dynamic range and subtle contrast differences important for later analysis.

2.3 Normalization

Each input dimension has the mean subtracted and is divided by the standard deviation. This ensures that the distribution of each dimension is approximately normal. K^{trans} values are also transformed with a log function to correct for large skew.

2.4 Example Weighting

ProstateX training labels are highly unbalanced, with 76 positives and 254 negatives. It is necessary to weight the classes appropriately,[4] the following formula is used:

$$\text{class weight} = \frac{\text{total number of examples}}{\text{number of classes} \times \text{number of examples in class}} \tag{1}$$

The positive class is given a weight of $330/(2 \times 76) = 2.17$ and negative class a weight of $330/(2 \times 254) = 0.65$. This can be interpreted as making it approximately three times worse to incorrectly label a positive case than a negative case during training.

Table 1. Comparison of cross validation performance for differing configurations

	AUC	C	γ
Image patch only	.782	50	10^{-5}
With zone	.806	30	10^{-5}
With class weights	.811	30	10^{-3}

2.5 Kernel Selection

The SVM kernel selected is the radial basis function (RBF); together called the RBF-SVM. This is a non-linear kernel so increases/decreases in one dimension do not necessarily cause proportional changes in score output. The authors believe that this is a desirable characteristic. Empirically, linear models also performed much worse and were abandoned early on.

2.6 Hyperparameter Selection

RBF-SVMs require tuning of two hyper-parameters a regularization parameter C and a kernel parameter γ. The highest scoring settings achieved with 3-fold cross validation[3] (CV) are used. Final AUC is calculated by averaging AUC over every fold. A simple grid search is carried out, where each combination of $C \in \{0.1, 0.5, 1, 2, 5, 10, 20, 30, 50\}$ and $\gamma \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ is scored and the best combination selected, see table 1 for selected values.

2.7 Scoring

The ProstateX challenge required entries to provide a continuous significance score for each test lesion, while SVMs produce a discrete binary classification. However a score can be easily derived for SVMs by using the decision function value directly instead of just the sign. This score increases in magnitude as the example moves further away from the SVM decision boundary. E.g. A score near zero indicates that small changes to this input would cause the prediction to change; while conversely, a large positive or negative value indicates that a small change in this input would not lead the prediction to change. Deriving the score in this way is practical and effective for this task.

2.8 Implementation

This competition entry is reproducible and implemented using only open source software including Python, PyDICOM, SimpleITK,[5] Scikit-Learn[6] and NumPy.[7] Source code is available from the authors under an open source license.

3. ALTERNATIVE APPROACHES

While the method descried here is self-contained, during development many different models and ideas were implemented and compared. Generally, multiple approaches were worked on in parallel. Ideas generated or problems encountered on one front would lead to progress on another. Some may be a path for future work.

Logistic regression[3] was implemented based on several hand coded features and on features derived from fully-connected autoencoders and convolutional autoencoders.[8] But these methods never surpassed .75 AUC in cross validation.

Convolutional neural networks[9] (CNN) were implemented, varying in depth from 3 to 20+ layers. Multiple architectures were tried including fully convolutional networks and residual networks. Generally it was very hard to prevent these large models from overfitting even with intense regularization and data augmentation. Models were heavily regularized using standard techniques, including dropout, gaussian activation noise and L2.

Semi-supervised learning with Generative Adversarial Networks (GAN)[10] was implemented to try and augment the training data with the unlabeled test data. While GANs were able to generate high quality fake lesion images, the CV performance was disappointing, hovering around .65 AUC despite being an extensive and sophisticated model.

Specialized data augmentation was implemented. Extra synthetic data is generated by applying small modifications to existing data, for example, randomly cropping an area from an existing scan or generating a small random 3D rotation and resampling MRI scan data along new transformed axes. This technique gave a small performance increase for some neural network models.

Extra metadata including age, weight and sex was extracted from inside the training DICOM files and input into the model, this did seem to modestly increase performance in some circumstances. Although this extra metadata was not used in the final competition submission.

4. DISCUSSION

These results demonstrate that with care and correct application, well understood and reliable machine learning tools apply well to computer-aided diagnosis. It is empirically shown that relatively simple models can compete with more complex or heavily engineered models. While not the most accurate model, simplicity, flexibility and computational efficiency have their own benefits and may make this model much easier to certify and deploy in real-world environments.

REFERENCES

[1] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine Learning* **20**(3), 273–297 (1995).

[2] Langer, D. L., van der Kwast, T. H., Evans, A. J., Plotkin, A., Trachtenberg, J., Wilson, B. C., and Haider, M. A., "Prostate tissue composition and mr measurements: investigating the relationships between ADC, T2, K-trans, ve, and corresponding histologic features 1," *Radiology* **255**(2), 485–494 (2010).

[3] Hastie, T., Tibshirani, R., and Friedman, J., [*The Elements of Statistical Learning*], Springer Verlag, New York, second ed. (2009).

[4] Yang, C.-Y., Yang, J.-S., and Wang, J.-J., "Margin calibration in SVM class-imbalanced learning," *Neurocomputing* **73**(1), 397–411 (2009).

[5] Lowecamp, B. C., Chen, D. T., Ibáñez, L., and Blezek, D., "The design of SimpleITK," *Frontiers in Neuroinformatics* **7**, 45 (2013).

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[7] Walt, S. v. d., Colbert, S. C., and Varoquaux, G., "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering* **13**(2), 22–30 (2011).

[8] Hinton, G. E. and Salakhutdinov, R. R., "Reducing the dimensionality of data with neural networks," *science* **313**(5786), 504–507 (2006).

[9] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet classification with deep convolutional neural networks," in [*Advances in Neural Information Processing Systems 25*], Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., eds., 1097–1105, Curran Associates, Inc. (2012).

[10] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., "Improved techniques for training GANs," in [*Advances in Neural Information Processing Systems*], 2226–2234 (2016).