
Comprehensive Examination

Question 3: Roger Day

Branislav Kveton
Intelligent Systems Program
University of Pittsburgh
bkveton@cs.pitt.edu

Introduction

At first, we present a short overview of all questioned machine learning techniques together with a discussion on their applicability in the classification of proteomic profiles. The overview is followed by a proposal of the data analysis plan, which does not attempt to be exhaustive. Proposed solutions can be characterized as intuitively offering high expected value within reasonable time.

PCA and PPCA

Principal component analysis (PCA) is a standard technique for dimension reduction and data compression. The method reduces the dimension of the original data k by producing a set of q principal vectors, which are perpendicular and represent the components of the data preserving the highest variance. The simplicity of the reduction and easily performed linear transformation between the subspaces contribute to the widespread use of the method. In addition to the solid underlying mathematical theory, there exist a lot of empirical evidence that point out to the scalability of the technique in several fields: authoritative sources in hyperlinked environment [8, 4], information retrieval [3], or spectral analysis of data.

Probabilistic principal component analysis (PPCA) [16] is an iterative algorithm that emerges as a special case of the latent variable model. As opposing to the standard approach for finding principal components, PPCA is based on a generative probabilistic model. This formulation offers several advantages.

First, PPCA can handle missing data, which is a problem for PCA. Second, the computational complexity of PPCA per one iteration is only $O(nkq)$, where n is the number of data samples, while the computation of the sample covariance matrix in PCA takes $O(nk^2)$ [9]. Linear dependence on q is an important feature of PPCA because the dimensionality of the input data

k is usually huge, while we look only for a small set of explanations $q \ll k$. Finally, PPCA models can be combined into a mixture of PPCA models [17], and thus naturally allow modelling of more than one data cluster.

However, PPCA is an iterative procedure, and the number of iterations before it converges may depend both on k and n . An empirical study of the convergence rate was performed by Roweis [9], and his results showed that the computation time for finding the principal eigenvector stayed almost constant while the dimensionality of the input data k varied from several attributes up to 450.

Both PCA and PPCA are suitable methods for performing dimensionality reduction of proteomic profiles. After the first q eigenvectors u_j are computed, data samples x_i are projected to the subspace spanned by the principal components

$$z_{ij} = u_j^T x_i, \quad (1)$$

where z_{ij} is a new j -th feature of the i -th data sample. PPCA is even more suitable for this task because the computation of the eigenvectors is only linearly dependent on the input space dimension k , which is huge in proteomic profiles.

Kernel trick

Kernel trick is a transformation that allows machine learning algorithms to operate in larger feature spaces, mostly nonlinearly related to the input space, while the computations are performed in the original input space. More concretely, if $\phi : X \rightarrow F$ is a transformation of the input space X to a larger feature space F , and the machine learning algorithm operates only on the dot products $\langle \phi(x), \phi(x') \rangle$, the issue is the construction of a kernel $k(x, x')$ such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle.$$

Among the most popular choices of kernels belong linear kernel

$$k(x, x') = \langle x, x' \rangle,$$

n -th degree polynomial kernel

$$k(x, x') = (1 + \langle x, x' \rangle)^n, \quad (2)$$

and radial basis kernel

$$k(x, x') = e^{-\frac{1}{2}\|x-x'\|^2}. \quad (3)$$

Even if reproducible kernels have become increasingly popular with the introduction of support vector machines, the kernel trick is not limited to only this machine learning technique: Bach [2] utilized kernels in independent component analysis and Scholkopf [14] extended principal component analysis to nonlinear spaces. Moreover, the notion of kernels goes far beyond the dot product operator, and kernels can be considered as a generalized distance metric in some feature space [13]. Therefore, kernels have been defined over large classes of structured data objects, including trees and Markov chains.

Kernels, viewed as powerful add-ins to support vector machines, are definitely the approach of our choice in the classification of proteomic profiles. By using higher order polynomial kernels, we can verify whether the data support more than linear decision boundaries without expansion of the input space. Moreover, the time complexity of the computation increases only by a multiplicative factor corresponding to the computation of the kernel.

Variational methods

Variational methods offer a deterministic framework for performing inference in the graphical models in which exact inference is not feasible [6, 7]. In general, by introducing a new variational parameter λ , variational methods create a lower (upper) bound of the original convex space, which lower (upper) bounds all further computations with the quantity. Fitting of the variational parameter λ is performed with respect to individual data samples x_i .

For example, multiplication of the probabilities of negative findings

$$P(f_i = 0|d) = e^{-\sum_{j \in \pi(i)} \theta_{ij} d_j - \theta_{i0}}$$

in the QMR-DT noisy-OR model [7] yields a term that is linear in the exponents of e . However, this nice coupling does not occur if the probabilities of positive findings

$$P(f_i = 1|d) = 1 - e^{-\sum_{j \in \pi(i)} \theta_{ij} d_j - \theta_{i0}}$$

are multiplied, which renders terms that are computationally expensive. By introducing a new variational parameter λ , it can be shown that

$$\begin{aligned} 1 - e^{-x} &\leq e^{\lambda x - f^*(\lambda)} \\ f^* &= -\lambda \ln \lambda + (\lambda + 1) \ln(\lambda + 1), \end{aligned}$$

where the first inequality serves as an upper bound on the quantity $P(f_i = 1|d)$. The upper bounded probabilities of positive findings are linear in the exponents of e , and thus their product can be evaluated without enumerating all combinations of parents $\pi(i)$.

Performing tractable inference in graphical models is the major application of variational methods in machine learning. As opposing to their sampling counterparts, variational methods are independent of randomization factor. However, produced bounds may be loose, especially if the variational approximations are used for multiple random variables. Therefore, variational methods should not be viewed as an ultimate replacement for exact inference, but rather as a technique for decomposing hard-to-solve graphs such that their subgraphs can be tackled by exact inference.

Variational methods are mainly optimization techniques, and therefore of very little use in the classification of proteomic profiles. If we had a graphical model capable of classifying proteomic profiles, and none of exact inference methods would be feasible, variational methods could be used to simplify the inference. However, this does not seem to be the case.

MCMC

Markov chain Monte Carlo (MCMC) algorithms form a large class of sampling methods that have played a significant role in solving integration and optimization problems [1]. Moreover, for some high dimensional optimization problems, MCMC is the only known technique that provides solutions within a reasonable time.

In machine learning, more specifically in the context of Bayesian inference, it is in general hard to compute normalization factors

$$p(x|y) = \frac{p(y|x)p(x)}{\int_x' p(y|x')p(x')dx'}$$

perform marginalization

$$p(x|y) = \int_z p(x, z|y)dz,$$

or compute an expectation

$$E_{p(x|y)}[f(x)] = \int_x f(x)p(x|y)dx.$$

The common problem in all three tasks is that we can usually evaluate a function or density in individual points, but there is no closed form solution for the integral. MCMC can solve these types of problem by defining an additional auxiliary distribution.

The fundamental idea in all MCMC methods is to explore a state space X by generating samples $x^{(i)}$ from a Markov chain. The construction of the chain is the key element that allows drawn samples to mimic a target distribution $p(x)$. For example, in the $(i + 1)$ -th step of the Metropolis-Hastings algorithm, Markov chain moves to a sample x^* only if

$$u < \min \left\{ 1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})} \right\},$$

where u is a random number from $[0, 1]$ and x^* is randomly drawn from a sampling distribution $q(x^*|x^{(i)})$. Intuitively, x^* has a higher chance to be accepted if either $p(x^*) > p(x^{(i)})$ or the chance of the backward transition $q(x^{(i)}|x^*)$ is higher than the probability of the forward transition $q(x^*|x^{(i)})$.

Metropolis algorithm and independent sampler are special cases of the Metropolis-Hastings algorithm. Gibbs sampler, widely used for resampling in Bayesian networks, samples random variables x_j^* given their complement $x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i)}, \dots, x_n^{(i)}$, and can be rewritten as a special case of the Metropolis-Hastings algorithm as well.

MCMC methods are optimization and density estimation techniques, and therefore of very little use in the classification of proteomic profiles. If we had a graphical model capable of classifying proteomic profiles, and none of exact inference methods would be feasible, MCMC can be used to implement a robust approximate inference algorithm. However, this does not seem to be the case.

SVM

Support vector machine (SVM) [18] classifier stems from the idea to separate binary classes by a linear boundary that has the maximum distance from the closest points of both classes. The problem of finding the optimal separating hyperplane between the two classes $y_i \in \{-1, 1\}$ can be reformulated as a quadratic optimization problem

$$\begin{aligned} \text{minimize}_{\beta, \beta_0} : & \quad \frac{1}{2} \|\beta\|^2 \\ \forall i : & \quad y_i(x_i^T \beta + \beta_0) \geq 1 \end{aligned}$$

and solved by utilizing Lagrange multipliers. In the classification phase, the label of a sample x is decided

by the sign of

$$x^T \beta + \beta_0 = \sum_{i: \alpha_i \neq 0} \alpha_i y_i \langle x, x^i \rangle + \beta_0,$$

where the Lagrange multipliers α_i are non-zero only for the samples on the decision boundary, called support vectors.

The popularity of SVMs is mostly caused by the fact that our results can be generalized far beyond linearly separable classes. A nice feature of SVMs is that both learning and prediction steps operate rather on dot products $\langle x, x' \rangle$ than single observations x . This allows introduction of various reproducible kernels, so we can easily obtain nonlinear decision boundaries in the input space X for no additional computational power. Moreover, ideas similar to those for the classification lead into the derivation of support vector regression models [15].

For the proteomic profile classification task, support vector machines seem to be one of the natural choices. First, the number of support vectors in high dimensional spaces can be low, which may result into learning of a robust classifier. Second, various reproducible kernels can help to examine nonlinear decision boundaries in the data for no additional price.

Boosting

Almost 15 years ago, Schapire [10] showed that the class of weak learners, which performs only slightly better than random guessing, is equal to the class of strong learners, which can produce correct hypothesis with the probability approaching 1. These ideas, which belong without any doubt to one of the major breakthroughs in the field of machine learning, gave rise to the adaptive version of boosting AdaBoost [11], which is nowadays a standard technique for the combination of weak learners. The main idea of the algorithm is an iterative creation of new classifiers that focus on the training examples that were misclassified by previous classifiers. In the prediction phase, the label of a data sample is decided by a weighted voting method among all classifiers.

One of the reason why boosting has attracted so much attention is because the empirical evidence presents the algorithm as resilient to overfitting, which is unfortunately not true [12]. However, it can be proved that the generalization error of boosting can be bound in terms of classifier complexity, sample size, and required precision [12]. Unfortunately, this bound is not very tight and do not explain the excellent performance of boosting. Friedman [5] offered another explanation of boosting as an additive logistic regression model. Moreover, it seems that boosting is even

capable of reducing the bias of a classifier [12]. In the light of the most recent research in the field, boosting is perceived as a power technique, albeit carrying a lot of controversy.

AdaBoost, an adaptive method for boosting of weak binary classifiers, is another machine learning algorithm suitable for the classification of proteomic profiles. Based on the experimental results of Shapire [12], we have a choice of two well-performing weak learners: trees and decision stumps. Our expectation is that boosting applied on decision stumps can reduce the bias of the learner and produce low testing errors. At the same time, the weight assigned to a decision stump learner can be interpreted as a discriminative importance assigned to a specific mass to charge value.

Data analysis plan

In the data analysis plan, we focus on two issues that seem to arise in the classification of proteomic profiles: (1) discrimination between cancer and non-cancer patients on the basis of proteomic profiles, which is the primary goal related to the choice of a machine learning method, and (2) dimensionality reduction, which seems to be an issue whenever one has more than ten thousand features and only few hundred learning examples.

This plan does not attempt to propose a method that would reach perfect discrimination between cancer and non-cancer patients. We suppose that in the first stage of developing a classifier, it is more important to try standard machine learning techniques and see how well they perform. Consequently, we can see how much can more sophisticated algorithms improve over this baseline. Finally, most of this proposal can be implemented and experimentally tried in less than a week.

Feature selection

There are many ways how to perform feature selection, starting from ad-hoc methods that do not seem to have any theoretical background, up to very well understood dimensionality reduction techniques. In the next paragraphs, we give an example of three methods that cover this ad-hoc to theory spectrum.

Peak selection belongs among less theoretical techniques supported mainly by empirical evidence. The main idea is to generate average profiles for each of the classes and extract features that correspond to the peaks in the profiles. Consequently, reduced feature set is formed as the union of peak features. Peaks naturally reflect our notion that something interesting is happening, and thus the intensity of the corresponding mass to charge value is higher. Moreover, the hope

is that the locations of peaks differ between both average profiles, and thus carry certain discriminative power. Smoothing of the original signal or average profiles may be necessary if a huge amount of peaks is identified.

Random selection of features is another way of reducing the dimensionality of the problem. Even if the technique may look nonsense, it can give us a lot of useful information. For example, if the classification algorithm performs well on randomly reduced feature set, the discriminatory signal may be scattered all over the profiles.

PCA is a well understood dimensionality reduction technique. The method is applied on mass to charge values from both types of profiles and identifies q principal components that correspond to the highest variance in the data. The hope is that the high variance is correlated with discriminatory power, and thus the projection to the principal components is the most discriminatory one. As the time complexity of PCA is quadratic in the number of features k , we can use PPCA, which computes the same transformation in the time linear in k . New feature set is obtained by a linear transformation of the input data (Equation 1).

Any of these feature reduction methods can be applied either offline, when the feature reduction is performed before the learning starts, or online, when new features are added adaptively during learning phase to improve the performance of the classification algorithm.

Classification

For binary classification task, there is a handful choice of statistical models. In the rest of this section, we assume that feature reduction has been already performed. In addition to SVM and boosting, both of which were primarily developed for binary classification, we would try even two simple machine learning classifiers: logistic regression and Naïve Bayes¹. Justification for using each of the classifier follows.

Both Naïve Bayes and logistic regression pose seemingly unrealistic conditions on their inputs: Naïve Bayes assumes that the features are independent of each other given the class label, and logistic regression models linear decision boundary between the posteriors of the classes. However, both methods perform extremely very in the classification of real world data, are easy to implement, and thus may serve as a reasonable baseline for other machine learning algorithms. Moreover, the complexity of inference and learning in Naïve Bayes grows only linearly in the size of input

¹Distribution of relative intensities for a specific mass to charge value is assumed to follow normal distribution.

space, and thus the method is suitable for running on the non-reduced set of features.

After obtaining preliminary results based on linear decision boundaries, SVMs can be easily adapted to estimate nonlinear decision boundaries by using polynomial (Equation 2) or radial basis (Equation 3) kernels. The advantage of utilizing kernels is that the input space does not have to be expanded. Moreover, the time complexity of learning and classification grows only by a multiplicative factor corresponding to the computation of the kernel. Experiments with SVMs can give us a notion of how much nonlinearity can be in the decision boundary before the model is overfitted.

Boosting, having ability to drive training error to zero without overfitting, is a good candidate for a classification method that would perform the best. In the role of weak learner, we can choose either decision stumps or trees. Our preference for stumps is guided by their lower complexity and performance that is only slightly worse the performance of decision trees [12]. Moreover, boosting with decision stumps is suitable for running on the non-reduced set of features. The technique can be viewed as an introduction of nonlinearities by a weighted voting system.

References

- [1] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.
- [2] Francis Bach and Michael Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, 2003.
- [3] Michael Berry, Zlatko Drmac, and Elizabeth Jessup. Matrices, vector spaces, and information retrieval. *SIAM*, 41(2):335–362, 1999.
- [4] Alan Borodin, Gareth Roberts, Jeffrey Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structures on the World Wide Web. In *World Wide Web*, pages 415–429, 2001.
- [5] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. Technical report, Stanford University, 1998.
- [6] Tommi Jaakkola. *Advanced Mean Field Methods: Theory and Practice*, chapter Tutorial on Variational Approximation Methods. MIT Press, 2000.
- [7] Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.
- [8] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [9] Sam Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, 1998.
- [10] Robert Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
- [11] Robert Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.
- [12] Robert Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the 14th International Conference on Machine Learning*, pages 322–330, 1997.
- [13] Bernhard Scholkopf. The kernel trick for distances. In *Advances in Neural Information Processing Systems*, volume 12, pages 301–307, 2000.
- [14] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
- [15] Alex Smola and Bernhard Scholkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT, 1998.
- [16] Michael Tipping and Christopher Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Aston University, 1997.
- [17] Michael Tipping and Christopher Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.
- [18] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1996.