Fast and accurate population structure prediction using a greedy support vector machine clustering algorithm

Usman Roshan, Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102

Abstract

Identification of population structure is a central problem in population and medical genetics. The model-based approach implemented in STRUCTURE and the k-means clustering on the principal component projection are two techniques that have been shown to accurately separate admixed populations. We introduce a greedy support vector machine clustering algorithm for solving this problem. Support vector machines are discriminative classification methods with strong theoretical guarantees. We demonstrate its performance on several real datasets. We show that our method obtains comparable accuracies than STRUCTURE and k-means applied to the principal component projection. Further, our algorithm is fast and can incorporate known ancestry of individuals.

Introduction

The problem of identifying population structure arises in the fields of medical and population genetics. In medical genetics identifying sub-populations and assigning individuals to them is a key step in conducting disease association studies. However, uncorrected population structure can produce false positives, as discussed in the literature [1,2,3,4]. In population genetics understanding of the structure can be used to uncover demographic history and address related scientific questions [5]. Consequently, several methods have been developed for identifying and correcting stratification in a given dataset (see [6] for references).

A fast and accurate approach is to compute the principal components, visualize the projection of the data on the largest principal components, and then apply the k-means clustering method on the projected data [7]. This is called principal component analysis (PCA) followed by k-means. This method has been shown to separate inter and intra-continental admixtures with very high accuracy [7]. STRUCTURE [9] is another popular and accurate program designed for this purpose. It estimates the joint probability distribution of population origin and allele frequencies using MCMC. Ancestry can then be predicted by the estimated probabilities. STRUCTURE is considered to be highly accurate; however, it has a slow running time.

The support vector machine (SVM) is a discriminative classification method introduced in [10]. To understand its application in the context of population structure, assume we are given an admixture of two sub-populations. Further, assume that each individual in the admixture is represented by its *d* dimensional principal component projection. These are also called *feature vectors*. Suppose we know the ancestry of some individuals from each sub-population. We refer to these individuals as *training data*. Given a training dataset the support vector machine is defined by the optimal hyperplane separating them, i.e., the hyperplane that maximizes the minimum distance of all points to the plane. We show how training data can be estimated for SVMs if no prior ancestry is available.

SVMs carry strong theoretical guarantees from a statistical learning viewpoint: the SVM classifier has minimum error on the training set and at the same time minimizes overfitting [11]. SVMs have had impressive empirical performance in various bioinformatics problems (see [11,12,13,14] for references). Furthermore, using the kernel trick [11], SVMs can compute non-linear separators instead of a hyperplane.

We describe in this paper a greedy SVM clustering solution for identifying population structure. We compare it to PCA followed by k-means and the popular STRUCTURE program on several admixtures of multiple sub-populations. We show that the SVM-clustering algorithm is highly accurate compared to the two methods and runs fast on large admixtures.

Results

Experimental design

We consider several real admixtures in this study obtained from various sources. We run both SVM-cluster and k-means on the PCA projection of each admixture. Each PCA projection was computed using the smartpca program of EIGENSOFT version 2.0 [15]. We use the top *k* principal components where *k* is the true number of clusters. Instead of the standard Euclidean k-means we use spherical k-means [25] that we find it to produce higher accuracies. Spherical k-means is similar to the original one except that each input vector is normalized to Euclidean norm one and the dot product is used as a measure of closeness. Following convention, the initial clustering for k-means is randomly chosen. Therefore we run k-means a 1000 times and report the accuracy of the run with the highest objective function value.

Due to running time considerations we run STRUCTURE only on the smallest admixture. We used STRUCTURE version 2.2 from the author's website. The output of STRUCTURE is a set of probabilities of the population origin of each individual. We compute a clustering from this by assigning each individual to the population with the higher probability.

Intuitively, we designed greedy SVM-clustering to add data points to an initial training set on which an initial SVM classifier is trained. With an initial SVM classifier it iteratively adds test points with the highest SVM discriminant values into the training set from the previous iteration. Throughout our experiments we use the e-sensitive loss function [11] with the tube width set to 0.1 and a polynomial degree four kernel. The initial training set is set to the k-tuple where (i.e. one individual from each predicted k-means cluster) that maximizes the classifier margin. The maximum number of iterations is set to the size of the smallest k-means cluster on the admixture. SVM-clustering is fully described in the Methods Section.

We use the rand coefficient to measure accuracy of a clustering [27,28]. Given a clustering we can represent it by a matrix C in which C_{ij}=1 iff i and j are in the same cluster and 0 otherwise. Let T and C be the matrices for the true and computed clusterings. Define n_{11} as the number of pairs (i,j) in which both T_{ij}=1 and C_{ij}=1 (and n_{00} similarly), and n_{10} and n_{01} where T_{ij}=1(0) and C_{ij}=0(1). The rand coefficient is the ratio $(n_{11}+n_{00})/(n_{11}+n_{00}+n_{01}+n_{10})$. It ranges from zero to one; one indicates 100% accuracy while zero is 0% accuracy.

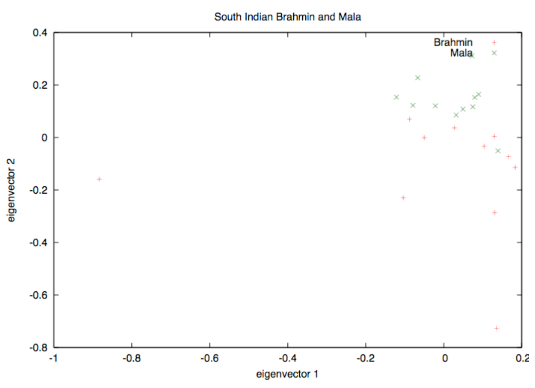
Finally, we run all analyses on dedicated 64 bit AMD Opteron 2.4 GHz processors.

Admixtures containing two sub-populations

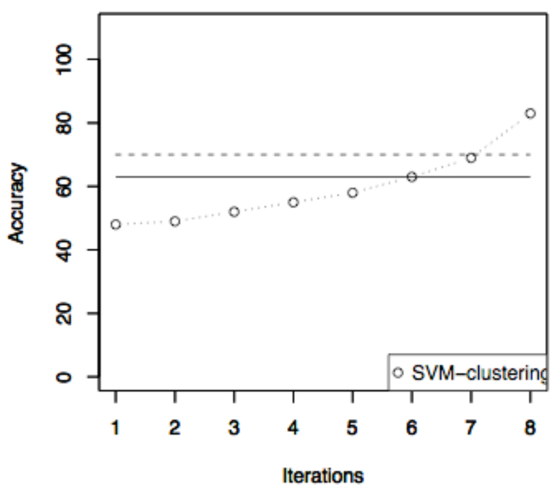
We first illustrate the performance of SVM-clustering on some binary admixtures before studying the general case of separating k sub-populations. Consider first an admixture of South Indian Mala and Brahmins obtained from a large dataset provided by Mark Shriver [16]. This admixture has 8805 SNPs after removing missing entries. The PCA projection on the two largest principal components is shown in Figure 1(a).

Figure 1: South Indian Mala and Brahmins

a: Projection on the two largest principal components.



b. Accuracies of SVM-clustering, k-means, and STRUCTURE



We apply all three methods, SVM-clustering, k-means, and STRUCTURE on this admixture. As indicated by the solid line in Figure 2, k-means has an accuracy of 63%. We ran STRUCTURE several times and found a mean accuracy of 70% with standard deviation 11%.

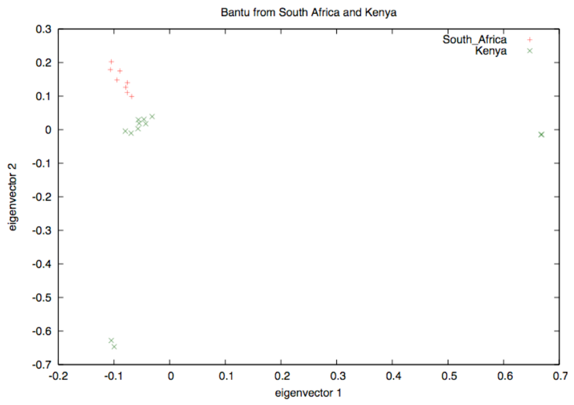
Given the relatively few number of SNPs in this admixture we apply SVM-clustering to the genotype data instead of the PCA projection. Let g be each individual's set of sequenced SNPs and let g_i represent the i^{th} SNP. g_i can be AA, AB, or BB where A and B are alphabetically ordered SNP bases. Following the encoding in [7] we define the feature vector x for g by setting x_i to +1 if g_i is AA, 0 if AB, and -1 if BB.

SVM-clustering reaches 75% accuracy at the last iteration. Further, it finishes in 8 seconds whereas STRUCTURE takes on average 342 seconds (with a very small standard deviation).

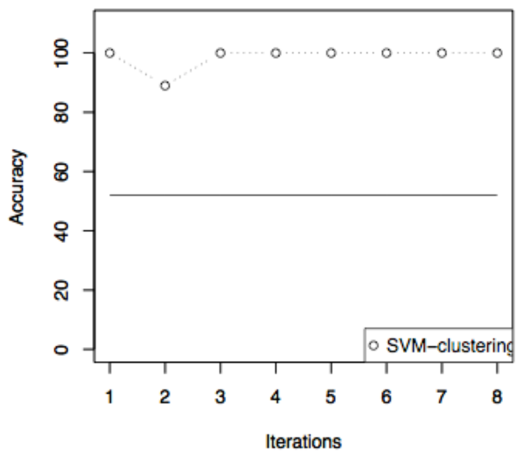
Now consider a Bantu admixture of 8 individuals from South Africa and 12 from Kenya, This was obtained from the HGDP dataset [17] and contains 499,748 SNPs after removing missing entries. Figure 2 shows the projection of the data on the two largest principal components.

Figure 2: South African and Kenyan Bantu

a: Projection on the two largest principal components.



b. Accuracies of SVM-clustering and k-means



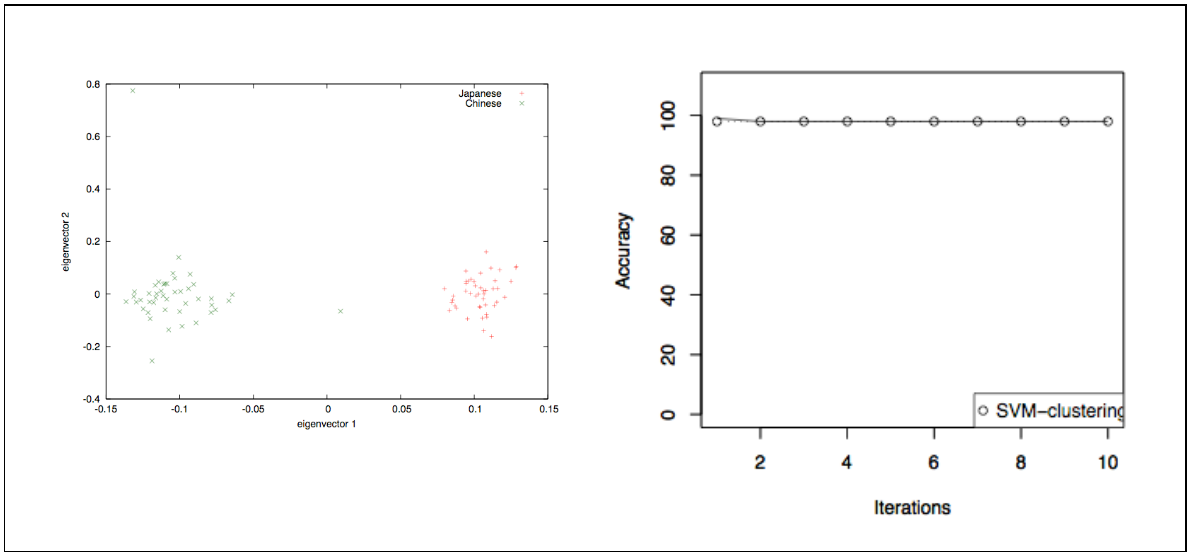
Most of the individuals are clustered in two sets in the upper left of Figure 2(a). However, there are four individuals that are not part of the two main clusters. As Figure 2(b) shows k-means attains 67% accuracy and SVM-clustering 90% in the last iteration.

Our last binary admixture consists of 45 Japanese and 45 Chinese with 1,621,467 SNPs after removing missing entries. This was extracted from the HAPMAP project [26] and has been used in a previous study [7]. Figure 3 shows the PCA projection and the accuracy of k-means and SVM-clustering when applied to the PCA projection. Both k-means and SVM-clustering attain the same accuracy of 98% due to the misclassification of a single outlier.

Figure 3: Chinese and Japanese from HAPMAP project

a: Projection on the two largest principal components.

b: Accuracies of SVM-clustering and k-means



Admixtures with several sub-populations

We now look at three large admixtures containing several sub-populations. We begin with the admixture of all four East Asian sub-populations from the HGDP dataset [17]. This admixture contains 10 individuals from Cambodia, 15 from Siberia, 49 from China, and 16 from Japan, each with 459,188 SNPs after removing missing entries. The PCA projection on the top two principal components is illustrated in Figure 4(a).

Figure 4: East Asian admixture of four sub-populations

a: Projection on the two largest principal components. **b:** Accuracies of SVM-clustering and k-means

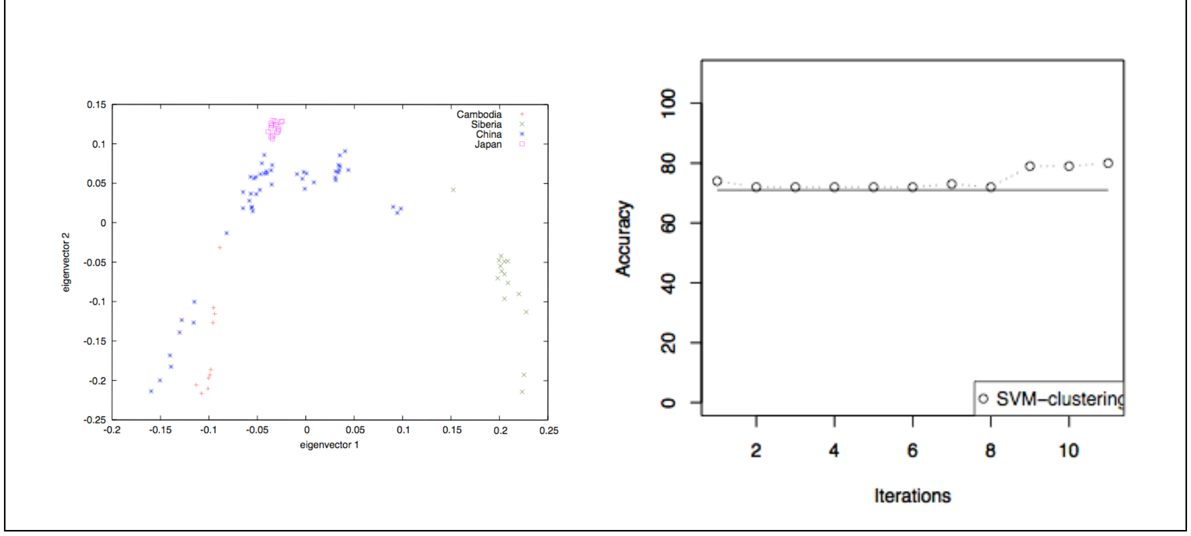
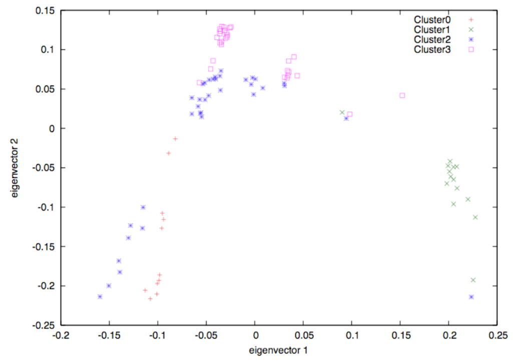


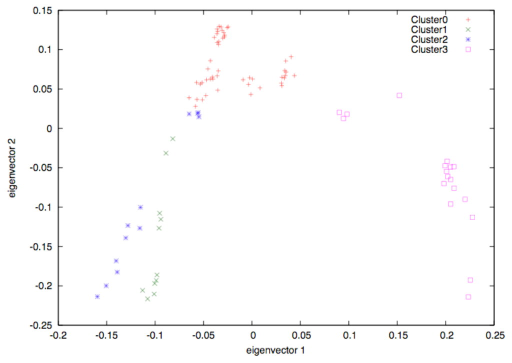
Figure 4(b) shows the accuracy of SVM-clustering and k-means when applied to the projection on the top four principal components. SVM-clusering attains 95% accuracy. The high accuracy of SVM-clustering on this admixture can be illustrated by visualizing its clustering on the PCA projection. In Figure 5(b) we see that k-means misclassifies most of the individuals of Chinese origin by grouping them together with Japanese. SVM-clustering on the other hand correctly classifies most of the Chinese individuals.

Figure 5: SVM-clustering and kmeans clusterings illustrated on the two dimensional PCA projection

a: SVM-clustering



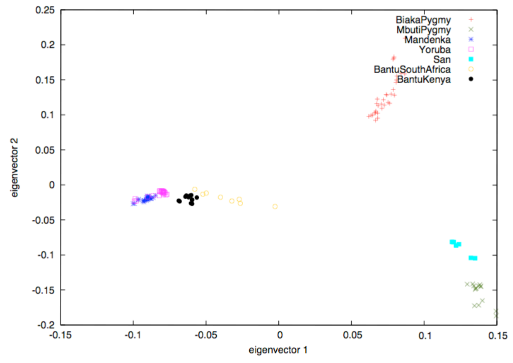
b. k-means



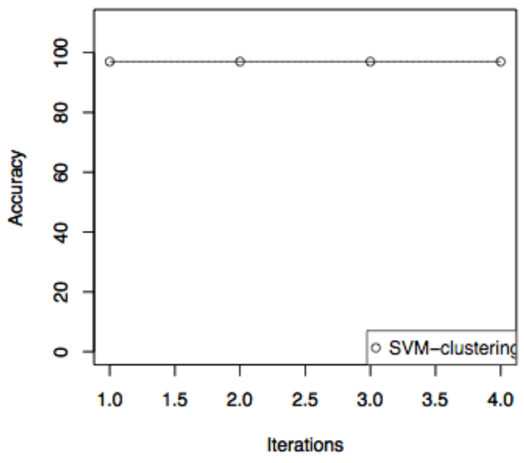
We now consider the admixture of all individuals of African origin in the HGDP dataset [17]. This admixture contains 32 Biaka Pygmy individuals, 15 Mbuti Pygmy, 24 Mandenka, 25 Yoruba, 7 San from Namibia, 8 Bantu of South Africa, and 12 Bantu of Kenya. After removing missing entries there are 454,732 SNPs remaining. The PCA projection on the top two principal components is illustrated in Figure 6(a). The accuracies of both k-means and SVM-clustering on the top seven principal components is graphed in Figure 6(b).

Figure 6: African admixture of seven sub-populations

a: Projection on the two largest principal components.



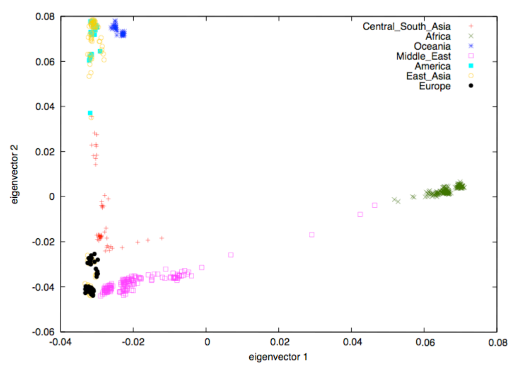
b. Accuracies of SVM-clustering and k-means

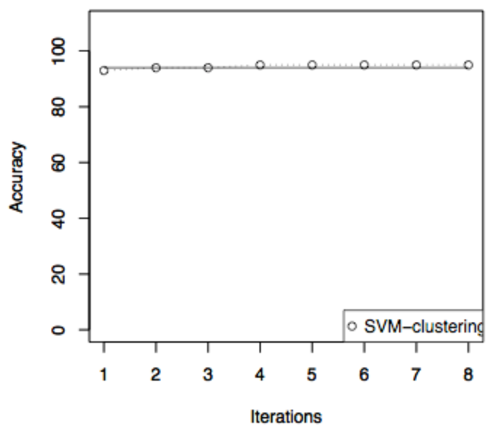


Finally, we look at the entire HGDP dataset. After removing all missing entries we are left with 333,396 SNPs. HGDP has seven regions: 50 individuals from Central South Asia, 159 from Africa, 33 from Oceania, 146 from the Middle East, 31 from America, 90 from East Asia, and 88 from Europe. The PCA projection on the top two principal components is shown in Figure 7(a) and the accuracies of k-means and SVM-clustering on the top seven principal components in Figure 7(b).

Figure 7: Worldwide population of seven regions

a: Projection on the two largest principal components. **b.** Accuracies of SVM-clustering and k-means





Discussion

Our selection of initial training data for SVM-clustering is not guaranteed to be always correct, i.e. each individual is selected from a separate sub-population. It depends upon the PCA projection and the k-means clustering. However, empirically it performs very well on the admixtures considered here.

In practice the prior ancestry of several individuals from each sub-populations may be known in advance. The prior ancestry may also be estimated using non-automatic methods such as visualizing the PCA projection. This can be used to produce better initial SVM models for SVM-clustering, which in turn can be expected to perform better than the initial estimates that we use in this study.

The application of SVM-clustering to other problem domains remains to be determined. We can expect kernels specifically designed for this problem to perform better than polynomial ones used here. We also expect kernel PCA to produce more illustrative projections for separating clusters and consequently improve the accuracy of clustering algorithms. We intend to explore both of these questions in future work.

Methods

Support vector machine background

Optimally separating hyperplane

Suppose we are given a set of *d* dimensional vectors x_i with labels y_i. We use the term feature space to describe the vector space that contains x_i. Each label y_i is +1 or -1 denoting the class x_i belongs to. Since the class membership of all x_i is known in advance, we call the set of x_i the training data. The support vector machine is defined by the optimally separating hyperplane between training data points (vectors) belonging to two classes. The direction of the hyperplane is defined by a vector *w* of dimension *d* and the distance to origin by a real number w_0. Both the parameters *w* and w_0 can be found by solving the following problem using Lagrange multipliers and the KKT conditions [11]:

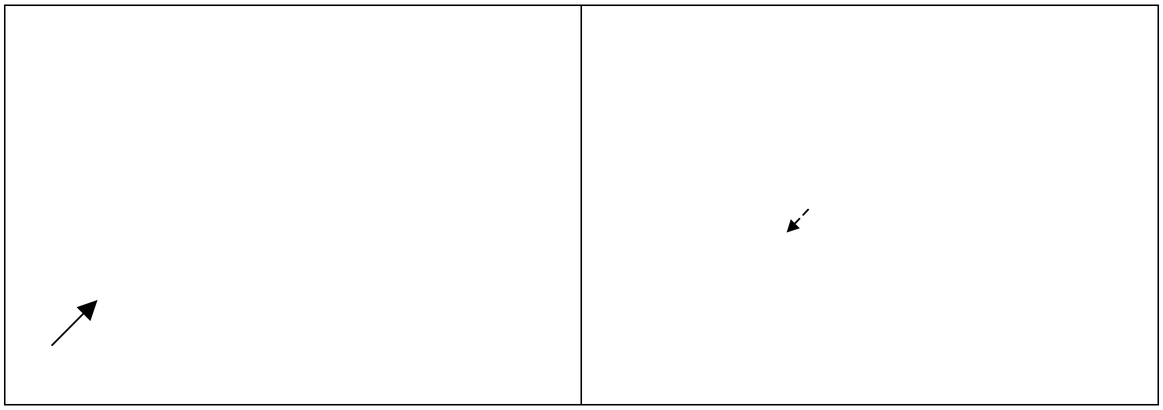
$$\underset{w\in R^d, w_0\in R}{\text{minimize}} \ \frac{1}{2}\|w\|^2 \text{ subject to } y_i(w^T x_i + w_0) \geq 1 \qquad (1)$$

The SVM classifier is then described by the function

$$g(x) = w^T x + w_0 \qquad (2)$$

Given a point *x* whose class is unknown, the sign of *g(x)* predicts which class, i.e. which side of the hyperplane, *x* belongs to. The margin of the classifier is defined by *2/||w||*. Maximizing the margin is the same as minimizing the *capacity* of the classifier (to avoid overfitting). The optimization problem in (1) ensures this happens but at the same time is constrained to minimize the empirical error. Figure 7 illustrates the geometrical interpretation of SVMs for two-dimensional data [11,18].

Figure 7: Toy example of an optimal hyperplane separating points on a plane (illustrated by squares and circles). In the left example 2/||w|| denotes the margin of the classifier. Points on the margin are at a distance of 1/||w||, i.e. *g(x)=1* or *-1* depending upon which side of the hyperplane they lie in. Maximization of the margin can be thought of as minimizing the capacity of the classifier. The example on the right shows one square misclassified and one circle inside the margin. This is the case when no hyperplane can separate the data points and therefore some points will be necessarily misclassified.



Linearly inseparable data

So far we have assumed that the training data is linearly separable by a hyperplane.

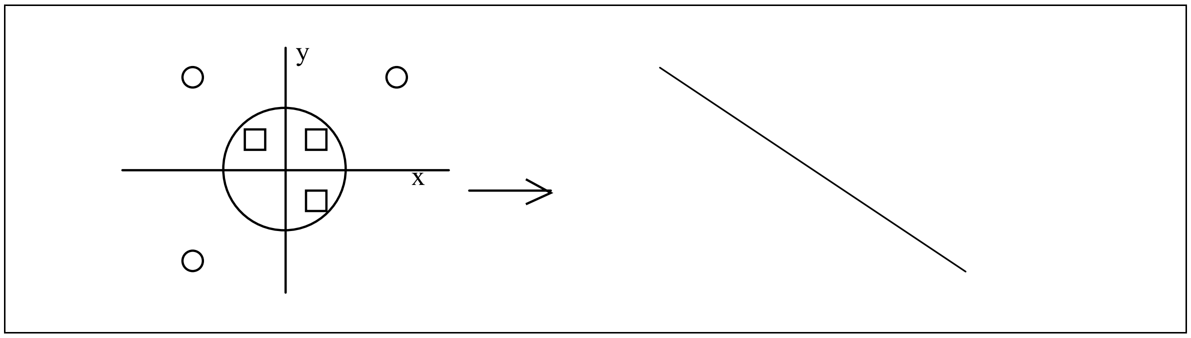
If this is not the case (see Figure 7), as one may expect in practice, we add error terms and minimize them. This gives the following optimization problem that is normally solved in practice [11,18].

$$\underset{w \in R^d, w_0 \in R}{\text{minimize}} \left(\frac{1}{2} \|w\|^2 + C \sum_i \xi_i \right) \text{ subject to } y_i(w^T x_i + w_0) \geq 1 + \xi_i \text{ and } \xi_i \geq 0 \qquad (3)$$

Here *C* is a regularization parameter that we set to 10^6. We can also consider mapping each x_i in our feature space to a higher dimensional one where it is linearly separable.

Suppose we have a function $\Phi(x)$ that maps each vector *x* to *z* where *z* may have more dimensions than *x*. This allows us to transform the feature space where there is no linear separator to a higher dimensional one where a linear separator may exist. Figure 8 illustrates a simple example. While this approach is very attractive for non-linear separation, it requires computing dot products in potentially many more dimensions and thus increases the running time. The *kernel trick* [11,18] allows us to keep the running time manageable.

Figure 8: The left figure shows data points that are linearly inseparable. The separator in the example is a circle. However, by the mapping $(x,y) \dashrightarrow (x^2,y^2)$ we can turn this into a linearly separable problem.



The polynomial kernel is defined by $K(x_i,x_j)=(x_i^Tx_j+1)^d$ where *d* is the degree of the polynomial. We use a degree three polynomial kernel throughout this study.

Support vector regression

Finding the optimal hyperplane can also be viewed as finding *w* and w_0 that minimize the regularized risk functional

$$\frac{C}{n} \sum_{i=1}^{n} \max(0, 1 - y_i g(x)) + \|w\|^2 \qquad (4)$$

The first term in (4) is called the *loss function* and defines the *empirical error* on the training set [11]. The loss function in (4) is also called the *hinge loss* and is used in the support vector classifier described above. *C* is the regularization parameter described earlier (3). The support vector regression problem is defined by using the *ε-sensitive* loss function instead of the hinge loss in (4). In regression we want to find *w* and w_0 that minimize

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - g(x_i)|_\varepsilon + \lambda\|w\|^2 \tag{5}$$

where

$$|x|_\varepsilon = \begin{cases} 0 & \text{if } |x| < \varepsilon \\ |x| - \varepsilon & \text{otherwise} \end{cases}$$

and *g(x)* is the same as defined earlier. We use the ε-sensitive loss function throughout this study with ε set to zero. In practice we find e-sensitive to be slower than

Support vector machine clustering algorithm

Previous clustering algorithms based on support vector machines explicitly search for the labeling of individuals that maximize the margin ($1/\|w\|$) [19,20,21]. We applied the best of these methods, an iterative maximum margin algorithm [21], to the problem considered here and found that it did not lead to comparable performance to SVM-clustering or STRUCTURE and PCA-kmeans.

Our procedure learns an SVM classifier from an initial training dataset and greedily improves it. At each step it adds test data points to the initial training set that have the highest discriminant values (in absolute terms). Consider an admixture of two sub-populations. In each iteration we pick *i* individuals with the highest SVM discriminant values and *i* with the lowest values. We set the ancestry of the top *i* individuals to one sub-population and the bottom *i* to the other one and replace the original training set with these *2i* individuals. Note that our ancestry settings may be different from the predicted ancestry of these individuals by the SVM classifier. Figure 9 describes the algorithm in full detail.

Our algorithm requires that some individual ancestry be known a priori. If prior ancestry of some individuals in the population is known then that can be specified. Let us look at the parameters of SVM-clustering in relation to this study. Variable *s*, which specifies the maximum number of iterations, is set to the size of the smallest k-means cluster on the PCA projection. Variable *inc*, which specifies the increment parameter, is set to one. The initial training dataset is set to the k-tuple of individuals (i.e. one individual from each predicted k-means cluster) out of 10^5 randomly selected k-tuples that minimize the SVM model complexity (i.e. $\|w\|$) under the polynomial degree three kernel.

We implement our algorithm using Perl scripts. All datasets and our code is available online at http://www.cs.njit.edu/usman/SVMclustering. Several packages exist for computing SVMs [23]. We use a fast C implementation called SVM_light [24] for learning the model and classifying the data

Figure 9: The SVM clustering algorithm for separating admixtures of *k* sub-populations

```
SVM clustering algorithm for admixture of k sub-populations
Input:
  • $k$ (number of sub-populations)
  • $U_t$ (training dataset; contains $t$ training points from each sub-population)
  • $T$ (set of all individuals to be clustered)
  • $inc$ (increment parameter; $inc < s$)
  • $s$ (stop parameter)
Output: clustering of individuals into their respective sub-populations
Algorithm:
  • For each $j=1...k$ form the training set $U_j$ as follows:
      o For each point in $U_t$, set its label to +1 if it is from sub-population $j$ and -1 otherwise.
  • For each $j=1...k$ learn SVM models $M_j$ from training data $U_j$.
  • Initialize $i=t+inc$ and repeat until $i=s$
      o Sort SVM discriminant values in each of the $k$ SVM models obtained previously.
      o For each $j=1...k$ form the new training set $U_j$ as follows:
          ▪ Set label of points with $i$ highest SVM discriminant values from the $j^{th}$ SVM
            model to +1
          ▪ Similarly, find points with the $i$ highest SVM discriminant values in each of
            the other SVM models (excluding the $j^{th}$ one) and set their labels to -1. (Note
            that our assigned labels of these points may be different from their predicted
            label by the classifier.)
      o For each $j=1...k$ learn SVM models $M_j$ from training data $U_j$.
      o Set $i=i+inc$
  • Form a final clustering from the $k$ final SVM models as follows:
      o Compute $k$ classifications on $T$ each using one of the $k$ SVM models
      o Assign point $p$ to sub-population $j$ if its discriminant value is largest under the $j^{th}$
        SVM model
```

The algorithm is fast in practice in practice because the baseline SVM programs for learning and classification are both fast. This also makes the estimation of the initial training set very fast. Furthermore, SVM-clustering is applied on the PCA projection instead of the full genotype data that has a much higher dimension.

Conclusion

We present an SVM-clustering algorithm and demonstrate its performance on real data. Our results show that SVM-clustering is a fast and accurate method for uncovering structure of large and hard admixtures.

Acknowledgements

Computational experiments were performed on the CIPRES cluster supported by NSF award 0331654. We thank the CIPRES team and system administrators at the San Diego Supercomputing Center for their support. We also thank Eran Halperin for motivating discussions on the population structure prediction problem.

References

1. Ziv E, Burchard E (2003) Human population structure and genetic association studies. Pharmacogenomics 4: 431–441.
2. Marchini J, Cardon L, Phillips M, Donnelly P (2004) The effects of human population structure on large genetic association studies. Nat Genet 36: 512–517.
3. Campbell C, Ogburn E, Lunetta K, Lyon H, Freedman M, et al. (2005) Demonstrating stratification in a European American population. Nat. Genet 37: 868–872
4. Devlin B, Roeder K (1999) Genomic control for association studies. Biometrics 55: 997–1004
5. Cavalli-Sforza L, Feldman M (2003) The application of molecular genetic approaches to the study of human evolution. Nat Genet 33: 266–275.
6. Tsai H, Choudhry S, Naqvi M, Rodriguez-Cintron W, Burchard E, et al. (2005) Comparison of three methods to estimate genetic ancestry and control for stratification in genetic association studies among admixed populations, *Hum Genet* 118: 424–433.
7. Paschou P, Ziv E, Burchard EG, Choudhry S, Rodriguez-Cintron W, et al. (2007) PCA-Correlated SNPs for Structure Identification in Worldwide Human Populations. *PLoS Genetics* 3(9): e160 doi:10.1371/journal.pgen.0030160
8. Allocco D.J., Song Q, Gibbons G.H., Ramoni M.F., Kohane I.S. (2007) Geography and genography: prediction of continental origin using randomly selected single nucleotide polymorphisms. *BMC Genomics*, 8:68.
9. Pritchard J, Stephens M, Donnelly P (2000) Inference of population structure using multilocus genotype data. *Genetics* 155: 945–959.
10. Vapnik, V.N., 1998. Statistical Learning Theory. Wiley, New York.
11. Scholkopf, B., & Smola, A. (2002). Learning with kernels. Cambridge, MA, MIT Press.
12. Support Vector Machine Classification of Microarray Gene Expression Data, Michael P. S. Brown William Noble Grundy, David Lin, Nello Cristianini, Charles Sugnet, Manuel Ares, Jr., David Haussler
13. Gene Selection for Cancer Classification using Support Vector Machines, I. Guyon, J. Weston, S. Barnhill and V. Vapnik, *Machine Learning* 46(1/3): 389-422, January 2002
14. A discrimitive framework for detecting remote protein homologies, T. Jaakkola, M. Diekhans, and D. Haussler, *Journal of Computational Biology*, Vol. 7 No. 1,2 pp. 95-114, (2000)
15. Patterson N, Price A, Reich D (2006) Population structure and eigenanalysis. PLoS Genet 2: e190. doi:10.1371/journal.pgen.0020190 Sacramento river chinook salmon. Genetics 151: 1115–1122.
16. Shriver MD, Mei R, Parra EJ, Sonpar V, Halder I, et al. (2005) Large-scale SNP analysis reveals clustered and continuous patterns of human genetic variation. Hum Genomics 2: 81–89.
17. M Jakobsson, SW Scholz, P Scheet, JR Gibbs, JM VanLiere, H-C Fung, ZA Szpiech, JH Degnan, K Wang, R Guerreiro, JM Bras, JC Schymick, DG Hernandez, BJ Traynor, J Simon-Sanchez, M Matarin, A Britton, J van de Leemput, I Rafferty, M Bucan, HM Cann, JA Hardy, NA Rosenberg, AB Singleton (2008) Genotype, haplotype and copy-number variation in worldwide human populations. *Nature* 451: 998-1003.
18. Alpaydin, E. (2004) Introduction to Machine Learning, MIT Press

19. Valizadegan, H., & Jin, R. (2007). Generalized maximum margin clustering and unsupervised kernel learning, *Advances in Neural Information Processing Systems* 19. Cambridge, MA: MIT Press.
20. Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2005). Maximum margin clustering, *Advances in Neural Information Processing Systems* 17. Cambridge, MA: MIT Press.
21. Kai Zhang, Ivor W. Tsang, James T. Kwok (2007) Maximum Margin Clustering Made Practical, *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, Oregon, USA
22. Perl scripts for SVM-clustering [www.cs.njit.edu/usman/svmclustering]
23. T. Joachims (1999) Making large-scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press
24. Ovidiu Ivanciuc (2007) Applications of Support Vector Machines in Chemistry, *Rev. Comput. Chem*. 23, 291-400
25. I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143–175, 2001.
26. The International HapPap Project [www.hapmap.org]
27. W. M. Rand (1971) Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, 66: 846–850.
28. K. Y. Yeung and W. L. Ruzzo (2001) Principal component analysis for clustering gene expression data, *Bioinformatics* 17 (9): 763–774.