Published in final edited form as:

J Appl Stat. 2013 February 1; 40(2): 358–367. doi:10.1080/02664763.2012.743977.

Sparse cluster analysis of large-scale discrete variables with application to single nucleotide polymorphism data

Baolin Wu*

Division of Biostatistics, School of Public Health, University of Minnesota, Minneapolis, MN, USA

Abstract

Current extremely large scale genetic data presents significant challenges for cluster analysis. Most existing clustering methods are typically built on Euclidean distance and geared toward analyzing continuous response. They work well for clustering, e.g., microarray gene expression data, but often perform poorly for clustering, e.g., large scale single nucleotide polymorphism data. In this paper, we study the penalized latent class model for clustering extremely large scale discrete data. The penalized latent class model takes into account the discrete nature of the response using appropriate generalized linear models and adopts the lasso penalized likelihood approach for simultaneous model estimation and selection of important covariates. We develop very efficient numerical algorithms for model estimation based on the iterative coordinate descent approach and further develop the Expectation-Maximization algorithm to incorporate and model missing values. We use simulation studies and applications to the international HapMap single nucleotide polymorphism data to illustrate the competitive performance of the penalized latent class model.

Keywords

Clustering; Expectation-Maximization algorithm; Latent class model; Lasso; K-means; Principal components; Single nucleotide polymorphism; Sparse clustering

1. Introduction

Cluster analysis has been extensively studied and many methods have been proposed (see, e.g., Kaufman and Rousseeuw [1] for an excellent introduction). Increasingly available ultrahigh dimensional and large-scale biomedical data, e.g., gene expression and genetic marker data, called for development of new clustering methods with built in variable selection, which can often improve the classification performance and model interpretability. Most existing clustering methods are typically built on Euclidean distance and geared toward analyzing continuous variables. Some model based methods have been proposed for variable selection in the context of clustering large-scale continuous response data including gene expression microarrays [see 2–4, e.g.]. Recently Witten and Tibshirani [5] proposed a very novel and general framework for variable selection in clustering applicable to various types of data. Through simulation and application studies, they have found that the proposed approach in general works very well for clustering large-scale continuous data but performs unsatisfactorily for large-scale discrete data. For analyzing discrete data, Houseman *et al.* [6] studied a penalized latent class model for analyzing small

scale loss of heterozygosity genomic data. And DeSantis *et al.* [7] further extended the model to analyze relatively small scale ordinal data. However for extremely large-scale discrete data, e.g., the HapMap data with hundreds of thousands of single nucleotide polymorphisms (SNPs), the latent class model poses numerical difficulty and has not been well studied for its clustering performance in the literature.

In this paper, we explore the penalized latent class model specifically tailored for clustering large scale discrete data. We will develop very efficient numerical algorithms for model estimation based on the iterative coordinate descent approach and further develop the Expectation-Maximization algorithm to incorporate and model missing values.

The rest of the paper is organized as follows. In Section 2 we discuss the penalized latent class model for clustering large-scale discrete data and develop computational algorithms for efficient model estimation. In Section 3 we conduct simulation studies to investigate the performance of the penalized latent class model. In Section 4 we analyze the international HapMap SNP data for illustration. Concluding remarks are provided at Section 5. All technical details are provided in the Appendix.

2. Statistical methods

Denote a set of m discrete variables measured on n samples as

$$x_{ij} \in \{1, \dots, K\}, \quad i=1, \dots, n, \quad j=1, \dots, m,$$

where K is the number of categories for each variable. Let $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$. Here for simplicity of notation and discussion, we have assumed that each discrete variable has the same number of categories. We note that the following model can be easily generalized to handle different number of categories.

Assuming there are G clusters, we fit the data with a finite mixture of independent multinomial distributions (also known as latent class model, see Goodman [8], e.g.)

$$\Pr(\mathbf{x}_i) = \sum_{g=1}^G \pi_g \prod_{j=1}^m \prod_{k=1}^K \theta_{gjk}^{S_{ijk}}, \quad \sum_{g=1}^G \pi_g = 1, \quad (1)$$

where S_{ijk} is the indicator of variable j having category k for sample i : $S_{ijk} = I(x_{ij} = k)$, θ_{gjk} is the corresponding category probability of cluster g : $\sum_{k=1}^K \theta_{gjk} = 1$, $\theta_{gjk} > 0$, and π_g is the mixing proportion representing the overall proportion of cluster g . The order selection for finite mixture model (i.e., choice of G) is a long standing and very difficult problem. For the following discussion, we assume that the number of clusters G is known. We will investigate the selection of G in future research. The latent class model has assumed a conditional independence structure: given the latent cluster membership, all variables are independent. Marginally without knowing the cluster membership, all variables are still modeled dependently. This conditional independence assumption can simplify the model estimation. Specifically we can develop Expectation-Maximization algorithms for the maximum likelihood estimation of model parameters.

We parametrize the multinomial distribution parameters using the logit transformation and model the cluster difference with a multinomial logistic regression model. To select

important features, we adopt the penalized latent class model with the lasso penalty [9]. Specifically for variable j , the cluster probabilities are parameterized symmetrically as

$$\theta_{gjk} = \frac{\exp(\alpha_{jk} + \beta_{gjk})}{\sum_{l=1}^K \exp(\alpha_{jl} + \beta_{gjl})}, \quad k=1, \dots, K, \quad \alpha_{j1} = \beta_{gj1} = 0. \quad (2)$$

Overall we fit a lasso penalized latent class model

$$\max_{\alpha_{jk}, \beta_{gjk}} \frac{1}{n} \sum_{i=1}^n \log \left\{ \sum_{g=1}^G \pi_g \prod_{j=1}^m \prod_{k=1}^K \theta_{gjk}^{S_{ijk}} \right\} - P(\beta, \lambda),$$

where

$$P(\beta, \lambda) = \lambda \sum_{j=1}^m \sum_{g=1}^G \sum_{k=1}^K |\beta_{gjk}|, \quad \lambda > 0.$$

The symmetric multinomial parameterization has its redundancy naturally removed by the use of lasso penalty. It also removes the arbitrariness of choosing one of the clustering group as reference.

With a suitably chosen large λ , $\{\beta_{gjk}\}_{g,k}$ could be shrunk down to zero for variable j , and hence it will not contribute to the cluster difference. Those variables with $\sum_{g,k} |\beta_{gjk}| > 0$ will be selected as important features for clustering samples.

We maximize the cross validated likelihood to select an optimal penalty parameter λ . In the following we discuss the Expectation-Maximization (EM) algorithm [10] for maximum likelihood estimation of the penalized latent class model.

2.1 EM algorithm for penalized latent class model

For sample i , we augment the data with cluster membership indicators z_{ig} following the Bernoulli distribution, $\Pr(z_{ig} = 1) = \pi_g$, and conditionally assume $\Pr(S_{ijk} = 1 | z_{ig} = 1) = \theta_{gjk}$. Then marginally $\mathbf{x}_i = \{S_{ijk}\}$ will follow the latent class model (1). Conditionally z_{ig} still follows the Bernoulli distribution with probability

$$\Pr(z_{ig} = 1 | \mathbf{x}_i) = \frac{\Pr(z_{ig} = 1) \Pr(\mathbf{x}_i | z_{ig} = 1)}{\Pr(\mathbf{x}_i)} = \frac{\pi_g \prod_{j=1}^m \prod_{k=1}^K \theta_{gjk}^{S_{ijk}}}{\sum_{l=1}^G \pi_l \prod_{j=1}^m \prod_{k=1}^K \theta_{ljk}^{S_{ijk}}}.$$

The penalized log likelihood for the complete data can be written as

$$\frac{1}{n} \sum_{i=1}^n \sum_{g=1}^G z_{ig} \left\{ \log \pi_g + \sum_{j=1}^m \sum_{k=1}^K S_{ijk} \log(\theta_{gjk}) \right\} - P(\beta, \lambda).$$

In the E-step, denote the conditional expectation as

$$\tau_{ig} = \mathbb{E}(z_{ig} | \mathbf{x}_i) = \frac{\pi_g \prod_{j=1}^m \prod_{k=1}^K \theta_{gjk}^{S_{ijk}}}{\sum_{l=1}^G \pi_l \prod_{j=1}^m \prod_{k=1}^K \theta_{ljk}^{S_{ijk}}}, \quad g=1, \dots, G. \quad (3)$$

To avoid rounding error, we will compute everything on the log scale and then center and normalize them to probabilities.

In the M-step, we can easily check that $\hat{\pi}_g = \sum_{i=1}^n \tau_{ig} / n$. For updating β_{gjk} , we need to solve a penalized multinomial logistic regression model separately for each variable. Specifically for variable j , we need to maximize

$$\frac{1}{n} \sum_{g=1}^G \sum_{k=1}^K n_{gjk} \log(\theta_{gjk}) - \lambda \sum_{g=1}^G \sum_{k=1}^K |\beta_{gjk}|, \quad n_{gjk} = \sum_{i=1}^n \tau_{ig} S_{ijk},$$

where n_{gjk} can be interpreted as the effective sample size for cluster g . We can employ a local quadratic approximation to iteratively solve this lasso penalized latent class model (see Appendix for details). We use multiple random K-means clustering results [1] as initial values for the EM algorithm, and choose the one that maximizes the cross validated likelihood. We can also naturally incorporate and model missing values in the proposed EM algorithm (see Appendix for details). In practice we run the EM algorithm with multiple initial values for a fixed λ and then very efficiently estimate the model for a sequence of λ using warm starting values based on the idea of iterative coordinate descent [11]. We can further improve the convergence of the EM algorithm based on the ECME algorithm [12]. Specifically note that given θ_{gjk} , the mixing proportion π_g can be very efficiently estimated with a simple recursion rule

$$\pi_g \leftarrow \frac{1}{n} \sum_{i=1}^n \frac{\pi_g \prod_{j=1}^m \prod_{k=1}^K \theta_{gjk}^{S_{ijk}}}{\sum_{l=1}^G \pi_l \prod_{j=1}^m \prod_{k=1}^K \theta_{ljk}^{S_{ijk}}}, \quad g=1, \dots, G.$$

In the next section, we conduct simulation studies to investigate the performance of the penalized latent class model.

3. Simulation study

In the simulation study, we compare the penalized latent class model (denoted as plcm) to the standard K-means clustering, and the sparse K-means clustering approach (denoted as sparcl) proposed by Witten and Tibshirani [5]. For standard K-means method, we run multiple random starting points and choose the clustering results with the smallest within sum of squares. For sparcl, the gap statistic is used to tune the model. For plcm, we use multiple random K-means clustering results as initial values for the penalized EM algorithm and 5-fold cross validation to choose optimal λ .

We simulate m Bernoulli random variables for 100 case and 100 control samples. We assume the first 100 variables are important features with case-control log odds ratio equal to 1. For the control group, the Bernoulli probabilities are simulated from a Beta

distribution, Beta(5,5). In the simulation, we investigate the impact of noise variables on the clustering performance. Specifically we did the simulation for the total of $m = 500, 1000, 2000, 3000, 4000, 5000$ variables. For each fixed m , we report the average clustering error, Rand index [13], sensitivity (i.e., true positive rate, which can be interpreted as power) and 1-specificity (i.e., false positive rate, which can be interpreted as Type-I error) over 100 simulations.

Table 1 and 2 report the average clustering errors and Rand indices. Overall we can see that plcm has the best performance. In general, noise variables have a large impact on the k-means clustering algorithm since it has no builtin feature selection. And plcm performs consistently well over all settings and is less affected by noise variables owing to its builtin feature selection with the lasso penalty. The sparcl method did not have a consistent performance over all settings: it could have larger clustering errors with less noise variables and more signals in our limited simulation studies. This is partly due to its difficulty of tuning parameter selection using the gap statistics, which agrees with the observations of Witten and Tibshirani [5] for clustering discrete variables. Compared to the k-means clustering algorithms, the main advantage of the sparcl and plcm method is the builtin important feature selection using the lasso penalized estimation approach. Table 3 summarizes the average sensitivity and specificity for the sparcl and plcm method. In general, both methods can correctly identify most significant variables for clustering when there are relatively small number of noise variables. Overall sparcl tends to select relatively more false positives compared to plcm.

We empirically compare the computing time of sparcl (using the ‘sparcl’ R package by Witten and Tibshirani [14]) and plcm. We implemented the plcm algorithm in C that is called from R. We used the R ‘system.time’ function for timing in a Linux workstation with 3 GHz Intel CPU and 8 GB memory. We evaluate the time used to compute all solutions for both methods with 25 lasso penalty parameter values under each simulation setting with m variables. We repeat 10 times and report the average time in seconds. Figure 1 shows the computing time. Overall we can see that both methods are very efficient and roughly scale linearly with the number of variables.

4. Application to HapMap SNP data

We analyzed the publicly available international HapMap data, which measured the single nucleotide polymorphism (SNP) variation of eleven worldwide populations [15]. A SNP is a site of common variation in DNA sequence where a single nucleotide differs between people: some people may have an C while others have a T instead. Each of the two possibilities is called an allele (denoted as A/B). We code the three genotypes (combination of two alleles), {AA, AB, BB} as {0, 1, 2} in the analysis. We studied whether we can use the measured SNP data to correctly classify populations and also identify those SNPs that differ across different populations. We download the Phase III HapMap SNP data from http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/2009-01_phaseIII/hapmap_format/consensus. We used those SNPs with all three genotypes being observed.

We restricted our analysis to chromosome 22 SNPs and three African populations: 90 samples from Luhya in Webuye of Kenya (denoted lwk), 171 samples from Maasai in Kinyawa of Kenya (denoted mkk), and 167 samples from Yoruba in Ibadan of Nigeria (denoted yri). The resulting data has in total 428 samples and 17083 SNPs. For plcm, we use random K-means clustering results as initial values to the penalized EM algorithm and 10-fold cross validated likelihood maximization to tune the lasso penalty parameter λ . We also use solutions from smaller λ as warm starting points for the next larger λ in the estimation,

which can dramatically improve the model convergence and reduce computation time. For sparcl, the gap statistic is used to tune the sparse K-means algorithm.

Figure 2 shows the first two principal components computed using all SNPs. In general we can see that all three population samples are mixed together with large overlap between lwk and yri. Table 4 shows the clustering results. All methods have good separation between mkk and yri population. K-means clustering using all SNPs yields 26.9% classification error rate with Rand index being 0.55. For k-means and sparcl, lwk samples are completely misclassified: 84/83 lwk samples are clustered into yri population and 6/7 lwk samples into mkk population. The sparcl performs better than k-means for clustering mkk population. It selected 14443 important SNPs and has 24.3% classification error with Rand index being 0.59. The plcm selected 2831 important SNPs and has the best performance with 8.4% classification error and Rand index being 0.81. Figure 4 shows the first two principal components based on the selected SNPs by sparcl and plcm. Overall, sparcl has a similar pattern as using all SNPs, while plcm has a much better separation of three populations.

5. Discussion

In this paper we have studied statistical methods for clustering extremely large-scale discrete data. Instead of relying on Euclidean distance based clustering methods, we have adopted the multinomial distribution based latent class model and the lasso penalized generalized linear model approach for simultaneous model estimation and variable selection. We have illustrated its competitive performance through simulation studies and applications. Throughout the discussion, we have assumed fixed number of clustering components and focus on the variable selection and classification performance. How to select the number of components in a finite mixture model is a very difficult and long standing problem. Most commonly used approaches, e.g., cross validating marginal likelihood or some model selection criterion (AIC or BIC), are known not working very well. Chen and Khalili [16] made important progress recently for order selection in the univariate finite normal mixture model under the large sample size setting. The key idea is to penalize the ordered cluster mean differences. We will investigate its generalization to clustering large-scale discrete data, and report the results elsewhere in the future.

Another important and challenging problem for clustering discrete variables has been the modeling of variable interactions. It has been a long standing problem since there are no convenient multivariate discrete distributions. A typical approach is adopting some working covariance matrix commonly used in the estimation equations. Alternatively the generalized linear mixed effects model can be used to model the interactions, which however poses great challenges for the maximum likelihood estimation of the extremely large-scale discrete data. Zhang [17] studied a hierarchical model framework addressing the variable interaction modeling, which is a promising approach though with very intensive computations. We will explore these approaches in the context of penalized clustering of discrete data and report the results in the future.

Acknowledgments

This research was supported in part by NIH grant GM083345 and CA134848. I would like to thank two anonymous referees for their constructive comments that have dramatically improved the presentation of the paper.

References

1. Kaufman, L.; Rousseeuw, P. An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics). Wiley-Interscience; 2005. Finding Groups in Data.

2. Pan W, Shen X. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*. 2007; 8:1145–1164.
3. Wang S, Zhu J. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*. 2008; 64:440–448. [PubMed: 17970821]
4. Xie B, Pan W, Shen X. Variable selection in penalized model-based clustering via regularization on grouped parameters. *Biometrics*. 2008; 64:921–930. [PubMed: 18162109]
5. Witten DM, Tibshirani R. A framework for feature selection in clustering. *Journal of the American Statistical Association*. 2010; 105 (490):713–726. [PubMed: 20811510]
6. Houseman EA, Coull BA, Betensky RA. Feature-specific penalized latent class analysis for genomic data. *Biometrics*. 2006; 62 (4):1062–1070. [PubMed: 17156280]
7. DeSantis SM, Houseman EA, Coull BA, Stemmer-Rachamimov A, Betensky RA. A penalized latent class model for ordinal data. *Biostatistics*. 2008; 9 (2):249–262. [PubMed: 17626225]
8. Goodman LA. Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*. 1974; 61 (2):215–231.
9. Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B (Methodological)*. 1996; 58:267–288.
10. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*. 1977; 39 (1):1–38.
11. Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*. 2010; 33:1. [PubMed: 20808728]
12. Liu C, Rubin DB. The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence. *Biometrika*. 1994; 81:633–648.
13. Hubert L, Arabie P. Comparing partitions. *Journal of Classification*. 1985; 2 (1):193–218.
14. Witten, DM.; Tibshirani, R. sparcl: Perform sparse hierarchical clustering and sparse k-means clustering. R package version 1.0.2. 2011. <http://CRAN.R-project.org/package=sparcl>
15. The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*. 2007; 449 (7164):851–861. [PubMed: 17943122]
16. Chen J, Khalili A. Order selection in finite mixture models with a nonsmooth penalty. *Journal of the American Statistical Association*. 2009; 104 (485):187–196.
17. Zhang NL. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*. 2004; 5:697–723.

Appendix

Lasso penalized latent class model

For simplicity of notation, we drop the index for variable j in the following discussion since the maximization algorithm is essentially the same for all variables.

The cluster probabilities are modeled as

$$\theta_{gk} = \frac{\exp(\alpha_k + \beta_{gk})}{\sum_{l=1}^K \exp(\alpha_l + \beta_{gl})}, \quad k=1, \dots, K, \quad g=1, \dots, G.$$

The conditional expected log likelihood can be written as

$$L = \frac{1}{n} \sum_{g=1}^G \left[\sum_{k=1}^K n_{gk} (\alpha_k + \beta_{gk}) - n_g \log \left\{ \sum_{k=1}^K \exp(\alpha_k + \beta_{gk}) \right\} \right], \quad n_g = \sum_{k=1}^K n_{gk}.$$

We consider the following lasso penalized likelihood maximization

$$\max_{\alpha_k, \beta_{gk}} L - \lambda \sum_{g=1}^G \sum_{k=1}^K |\beta_{gk}|.$$

We can check that

$$\frac{\partial L}{\partial \alpha_k} = \frac{1}{n} \sum_{g=1}^G (n_{gk} - n_g \theta_{gk}), \quad \frac{\partial^2 L}{\partial \alpha_k^2} = -\frac{1}{n} \sum_{g=1}^G n_g \theta_{gk} (1 - \theta_{gk}),$$

$$\frac{\partial L}{\partial \beta_{gk}} = \frac{n_{gk} - n_g \theta_{gk}}{n}, \quad \frac{\partial^2 L}{\partial \beta_{gk}^2} = -\frac{n_g}{n} \theta_{gk} (1 - \theta_{gk}).$$

We adopt the sequential Newton-Raphson update

$$\alpha_k \leftarrow \alpha_k + \frac{\sum_{g=1}^G (n_{gk} - n_g \theta_{gk})}{\sum_{g=1}^G n_g \theta_{gk} (1 - \theta_{gk})},$$

$$\beta_{gk} \leftarrow \text{sign}(\widehat{\beta}_{gk}) \left(|\widehat{\beta}_{gk}| - \frac{n\lambda}{n_g \theta_{gk} (1 - \theta_{gk})} \right)_+,$$

where the subscript plus means positive part and

$$\widehat{\beta}_{gk} = \beta_{gk} + \frac{n_{gk} - n_g \theta_{gk}}{n_g \theta_{gk} (1 - \theta_{gk})}.$$

Missing value modeling

We can naturally incorporate missing values in the EM algorithm. Specifically in the model the indicator S_{ijk} will be treated as random variable if x_{ij} is not observed. Denote O_i as the indices of variables without missing values for sample i . Here the observed data $\mathbf{x}_i$ only consists of those variables without missing values for sample i , i.e., $\mathbf{x}_i = \{x_{ij} : j \in O_i\}$. For variable $j \notin O_i$ with missing value, we can compute

$$\Pr(z_{ig} S_{ijk} = 1, \mathbf{x}_i) = \Pr(z_{ig} = 1) \Pr(S_{ijk} = 1 | z_{ig} = 1) \Pr(\mathbf{x}_i | z_{ig} = 1, S_{ijk} = 1) = \pi_g \theta_{gjk} \eta_{ig},$$

where

$$\eta_{ig} = \prod_{l \in O_i} \prod_{k=1}^K \theta_{gik}^{S_{ilk}}.$$

Therefore

$$\gamma_{igjk} = \mathbb{E}(z_{ig} S_{ijk} | \mathbf{x}_i) = \frac{\pi_g \eta_{ig} \theta_{gjk}}{\sum_{l=1}^G \pi_l \eta_{il}} = v_{ig} \theta_{gjk},$$

where

$$v_{ig} = \mathbb{E}(z_{ig} | \mathbf{x}_i) = \sum_{k=1}^K \gamma_{igjk} = \frac{\pi_g \eta_{ig}}{\sum_{l=1}^G \pi_l \eta_{il}}, \quad g=1, \dots, G,$$

and

$$\mathbb{E}(S_{ijk} | \mathbf{x}_i) = \frac{\sum_{g=1}^G \pi_g \eta_{ig} \theta_{gjk}}{\sum_{g=1}^G \pi_g \eta_{ig}} = \sum_{g=1}^G v_{ig} \theta_{gjk}, \quad k=1, \dots, K,$$

which can also be used to impute the missing value for variable j of sample i . Denote M_j as the indices of samples with missing value for variable j , and we then compute

$$n_{gjk} = \sum_{i \notin M_j} \tau_{ig} S_{ijk} + \sum_{i \in M_j} v_{ig} \theta_{gjk}.$$

Figure 1.
Computing time with respect to number of covariates for `sparcl` and `plcm`

Figure 2.
First two principal components computed using all SNPs

Figure 3.
First two principal components computed based on selected SNPs

Table 1

Average percentage of clustering errors over 100 simulations (listed within parenthesis are standard errors).

m	500	1000	2000	3000	4000	5000
k-means	3.3 (0.15)	6.3 (0.26)	14.5 (0.57)	26.5 (0.88)	33.1 (0.90)	36.6 (0.78)
sparcl	18.3 (1.30)	8.0 (0.75)	12.2 (0.93)	27.1 (1.32)	32.6 (1.11)	36.4 (0.98)
plem	2.6 (0.13)	3.3 (0.16)	4.8 (0.26)	11.5 (1.21)	19.4 (1.63)	26.6 (1.77)

Table 2

Average Rand indices over 100 simulations (listed within parenthesis are standard errors).

m	500	1000	2000	3000	4000	5000
k-means	0.87 (0.01)	0.77 (0.01)	0.51 (0.01)	0.25 (0.01)	0.14 (0.01)	0.09 (0.01)
sparcl	0.47 (0.03)	0.73 (0.02)	0.60 (0.02)	0.28 (0.02)	0.17 (0.02)	0.11 (0.01)
plem	0.90 (0.01)	0.87 (0.01)	0.82 (0.01)	0.65 (0.03)	0.48 (0.03)	0.34 (0.04)

Table 3

Average sensitivity and specificity (%) over 100 simulations (listed within parenthesis are standard errors).

m	500		1000	
	sensitivity	1-specificity	sensitivity	1-specificity
sparcl	90.8 (1.66)	60.1 (3.95)	83.8 (1.70)	16.4 (1.76)
plcm	97.2 (0.19)	34.4 (0.63)	93.4 (0.32)	18.8 (0.49)
m	2000		3000	
	sensitivity	1-specificity	sensitivity	1-specificity
sparcl	82.0 (2.12)	18.1 (1.42)	55.2 (3.05)	19.8 (1.60)
plcm	83.3 (0.80)	8.0 (0.42)	62.5 (2.25)	4.3 (0.35)
m	4000		5000	
	sensitivity	1-specificity	sensitivity	1-specificity
sparcl	36.3 (2.43)	13.7 (1.42)	25.5 (2.33)	9.89 (1.00)
plcm	47.0 (2.65)	4.2 (0.33)	33.6 (2.76)	3.95 (0.23)

Table 4

Clustering comparison of HapMap SNP data.

	clustering membership											
	k-means			spare1			plcm					
	1	2	3	1	2	3	1	2	3	1	2	3
lwk	0	6	84	0	7	83	60	3	27			
mkk	21	146	4	10	157	4	6	165	0			
yri	0	0	167	0	0	167	0	0	167			
clustering error			26.9%			24.3%			8.4%			
Rand index			0.55			0.59			0.81			