Appendix A. Introduction to Binary Support Vector Machines

Below we summarize the main ideas behind binary Support Vector Machines via a short theoretical description followed by an example. For a detailed review of SVMs, refer to [Vapnik1998] and [Burges1998].

1. Linear Support Vector Machines

Given a set of n training points from the m-dimensional space $x_1,...,x_n \in \Re^m$ with positive and negative class labels $y_1,...,y_n \in \{-1,1\}$, linear SVMs solve an optimization problem that seeks a maximum margin classifier (i.e. a hyperplane with the maximum margin width that separates training instances of two classes). This classifier is defined by a subset of training data points called *support vectors*. Then unseen data instances (i.e. samples, which were not used for training) are classified based on which side of the hyperplane they fall into. Mathematically, the separation hyperplane is defined by the equation $x^T w + b = 0$ where $w \in \Re^m$ and b come from the solution of the optimization problem. The decision function is then defined by $f(x) = \text{sgn}(x^T w + b)$.

Consider an example cancer diagnostic problem with two possible outcomes, T-cell acute lympboblastic leukemia (ALL) and B-cell ALL, using expression levels of two genes: c-Myb and SNF2. Given a training set of patients (**Figure A1**), our goal is to build an SVM based diagnostic model. There is obviously an infinite number of hyperplanes (lines in this two-dimensional space) separating data points of two classes. Two possible hyperplanes are shown with dotted lines in **Figure A1**. Linear SVMs provide an "optimal" classifier (bold line) that has the maximum margin width. This classifier is based on 5 support vectors (highlighted with circles) – two ALL T-cell data points and three ALL B-cell data points. The decision function applied to the unseen instance 1 (depicted as a triangle with number 1 inside) will classify it as ALL T-cell because it is above the separation hyperplane. By the same token, the unseen instance 2 (triangle with number 2 inside) will be classified as ALL B-cell since it is below the separation hyperplane.

In practice, linear SVMs may be applied to the datasets that are non-separable (i.e. when it is impossible to come up with a hyperplane separating training instances of two classes without errors). The SVM algorithm can handle this case by using tradeoff (cost) parameter C and penalty parameters (slack variables).

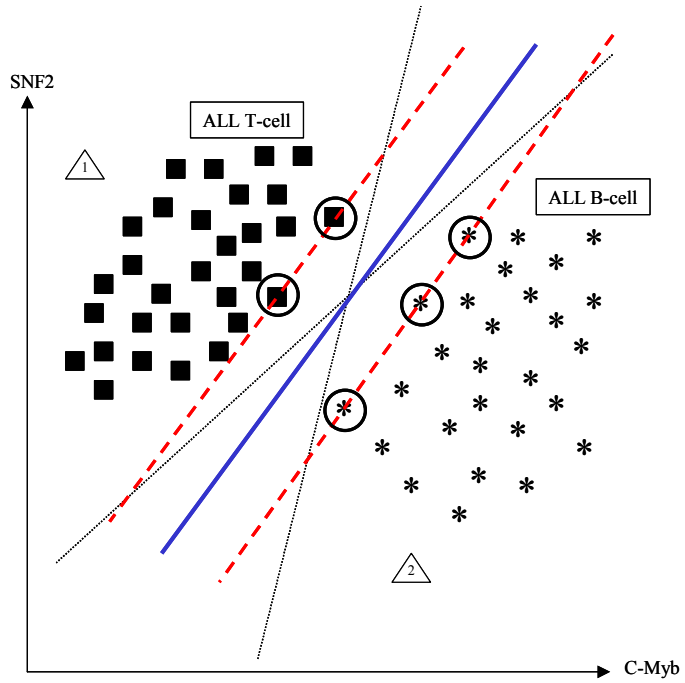

Figure A1. *Binary linear SVMs applied to the example diagnostic problem with two outcomes: ALL T-cell (■) and ALL B-cell (∗).*

2. Non-linear Support Vector Machines

The real power of SVMs comes into play by application of kernel functions, which allow to implicitly map data into a higher dimensional space, called *feature space*. This can be very useful if the training data are non-separable in the input space and become separable in the feature space. Among common kernel functions are linear, polynomial, and radial basis.

Consider an example cancer diagnostic problem involving two possible outcomes, T-cell acute lympboblastic leukemia (ALL) and acute myelogenous leukemia (AML), using expression levels of two genes: c-Myb and SNF2. As it can be seen from the **Figure A2**, data are non-separable in the input space. However, the application of polynomial kernel function maps the data into a higher dimensional feature space where they are separable (**Figure A3**). The resulting hyperplane separating two classes in also shown in the figure. Note that the axes in the feature space are kernel bases dependent on the input variables, expression levels of c-Myb and SNF2.

Similarly to the linear case, slack variables and cost parameter are applicable to non-linear SVMs and can be useful in the situations when the data are still non-separable in the feature space.

It is also important to mention that both linear and non-linear SVMs do not need to directly access to training data points and require only pairwise dot products of the data instances.

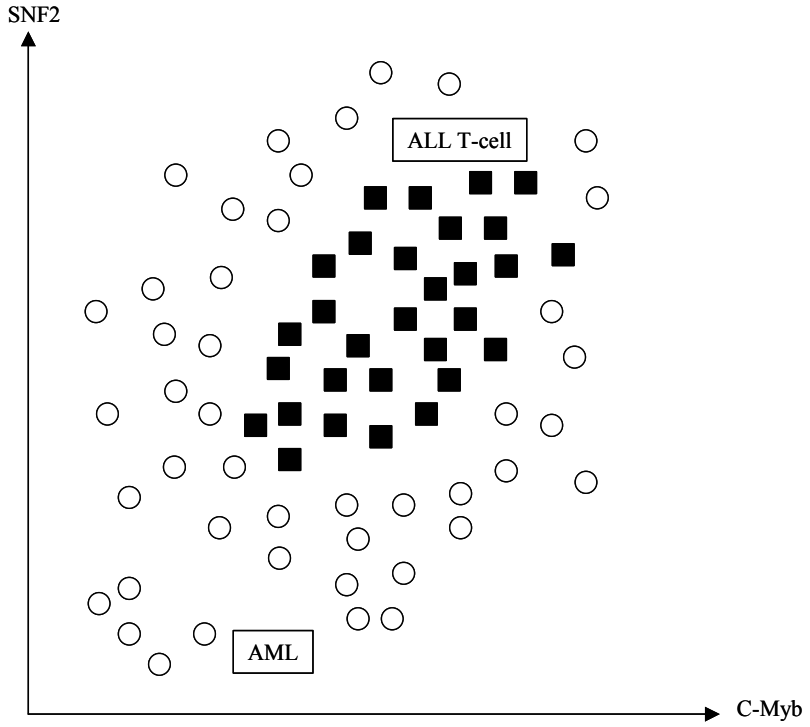

Figure A2. *Input space of the example diagnostic problem with two outcomes: ALL T-cell (■) and AML (○).*

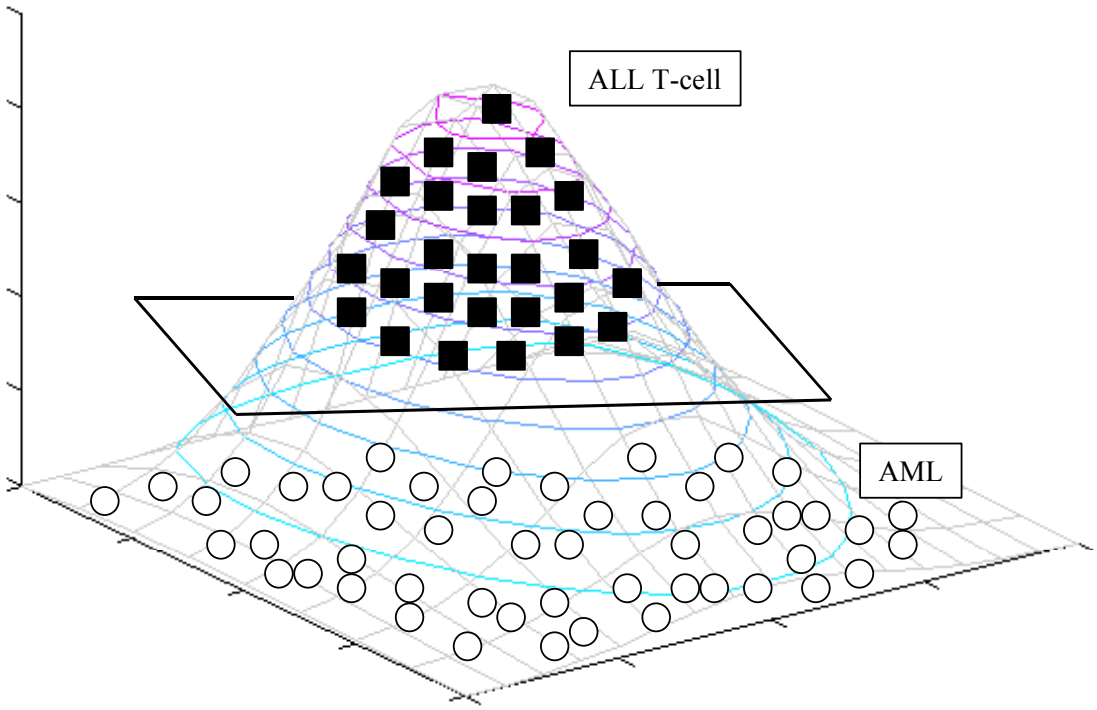

Figure A3. *Feature space of the example diagnostic problem with two outcomes: ALL T-cell (■) and AML (○). The resulting hyperplane is obtained by SVMs with polynomial kernel.*

References

[Vapnik1998] Vapnik, V. "Statistical Learning Theory", Wiley-Interscience, 1998.

[Burges1998] Burges, C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

Appendix B. Mathematical Formulations of Binary and Multicategory SVMs

Below we summarize mathematical formulations of binary and multicategory Support Vector Machines. Refer to papers cited in text for full descriptions of the algorithms.

1. Binary SVMs

Given *n* training instances of dimension *m*: $x_i \in \Re^m$ and corresponding class labels $y_i \in \{-1,+1\}$ ($i = 1,2,..,n$), the margin between two classes is optimized via a solution of the following quadratic constrained optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{i=1}^{n} \xi_i \text{ subject to}$$

$$y_i((w^T \Phi(x_i)) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for } i = 1,2,..,n$$

where $w \in \Re^m$ is a vector of weights of training instances; *b* is a constant; *C* is a real-valued tradeoff (cost) parameter; ξ_i is a penalty parameter (slack variable); and Φ is a map from the input space $\Re^m$ to the typically much larger dimensional feature space $\Re^r$ ([Vapnik1998] and [Burges1998]). This optimization problem is quadratic in *n* variables and *n* constraints and depends on the data only through dot products $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$. Hence, there is no need to know mapping Φ explicitly, since one has to use only the function *K* (called a *kernel function*) to solve the optimization problem. Among commonly used kernel function are:

1. Linear kernel: $K(x_i, x_j) = x_i^T x_j$;
2. Polynomial kernel: $K(x_i, x_j) = (\gamma \cdot x_i^T x_j + r)^p$, where $\gamma, r \in \Re$;
3. Radial-basis kernel: $K(x_i, x_j) = \exp\left(-\sigma \left\| x_i - x_j \right\|^2\right)$, where $\sigma \in \Re$.

Given *w* and *b*, one can classify an instance *x* using the decision function:

$$f(x) = \text{sgn}[w^T \Phi(x) + b].$$

2. Multicategory SVMs

In formulations of multiclass SVM methods described below adopt the following notation: $x_i \in \Re^m$ are *m*-dimensional training instances and $y_i \in \{1,2,...,k\}$ ($i = 1,2,..,n$) are corresponding class labels.

One-vs-rest (OVR). The margins between each of *k* classes and the remaining classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w_p, b_p, \xi^p} \frac{1}{2} w_p^T w_p + C\sum_{i=1}^{n} \xi_i^p \text{ subject to}$$

$$w_p^T \Phi(x_i) + b_p \geq 1 - \xi_i^p, \text{ if } y_i = p,$$

$$w_p^T \Phi(x_i) + b_p \leq -1 + \xi_i^p, \text{ if } y_i \neq p,$$

$$\text{and } \xi_i^p \geq 0 \text{ for } i = 1,2,..,n$$

where $p \in \{1,2,...,k\}$; $w_p \in \Re^m$ is a vector of weights of training instances; $b_p \in \Re$; *C* is a real-valued tradeoff (cost) parameter; ξ_i^p is a penalty parameter (slack variable); and $\Phi : \Re^m \rightarrow \Re^r$ ([Kressel1999]). In total one needs to solve *k* constrained QP problems (for *k* values of *p*) with *n* variables and *n* constraints. Given optimal weights *w* and *b*, the following decision function is used for classification of an instance *x*:

$$f(x) = \arg\max_{p=1,...,k} [w_p^T \Phi(x) + b_p].$$

Note, that for the case *k=2*, this technique is equivalent (i.e. its hyperplane is identical) to the binary SVMs.

One-vs-one (OVO). The margins between each pair of *k* classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w_{pq},b_{pq},\xi^{pq}} \frac{1}{2} {w_{pq}}^T w_{pq} + C\sum_{i=1}^{n} \xi_i^{pq} \text{ subject to}$$

$${w_{pq}}^T \Phi(x_i) + b_{pq} \geq 1 - \xi_i^{pq}, \text{ if } y_i = p,$$

$${w_{pq}}^T \Phi(x_i) + b_{pq} \leq -1 + \xi_i^{pq}, \text{ if } y_i = q,$$

$$\text{and } \xi_i^{pq} \geq 0 \text{ for } i = 1,2,..,n_1$$

where $p \in \{1,2,...,k\}$; $q \in \{1,2,...,k\} \setminus p$; $n_1 \leq n$ is the number of training instances with class labels *p* and *q*; $w_{pq} \in \Re^m$ is a vector of weights of training instances; $b_{pq} \in \Re$; *C* is a real-valued tradeoff (cost) parameter; ξ_i^{pq} is a penalty parameter (slack variable); and $\Phi : \Re^m \to \Re^r$ ([Kressel1999]). In total one needs to solve $\binom{k}{2} = \frac{k(k-1)}{2}$ constrained QP problems (for all distinct pairs of *p* and *q*) with n_1 variables and n_1 constraints. If different classes have the same priors in the training dataset, then $n_1 = \frac{2n}{k}$. Given optimal weights *w* and *b*, individual decision functions for all distinct pairs of *p* and *q* are computed for an instance *x*:

$$f_{pq}(x) = \text{sgn}[w_{pq}^T \Phi(x) + b_{pq}].$$

There are various methods to combine votes of the individual decisions functions into a final decision. A common approach, so-called *Max Wins strategy*, is to assign an instance to a class which has the largest number of votes [Friedman1996].

Note, that for the case *k=2*, this technique is equivalent (i.e. its hyperplane is identical) to the binary SVMs.

DAGSVM. The margins between each pair of *k* classes are optimized via solution of the constrained QP optimization problem, same as for OVO. When all QP problems are solved, a new instance *x* is classified using DDAG, a rooted binary decision directed acyclic graph, constructed on the basis of $\binom{k}{2}$ individual classifiers (nodes) and *k* leaves corresponding to the classification decisions [Platt2000]. The choice of the class order in the DDAG list can be arbitrary as shown empirically in [Platt2000].

Method by Weston and Watkins (WW). The margins between all *k* classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} \sum_{p=1}^{k} w_p^T w_p + C\sum_{i=1}^{n} \sum_{p \neq y_i} \xi_i^p \text{ subject to}$$

$$w_{y_i}^T \Phi(x_i) + b_{y_i} \geq w_p^T \Phi(x_i) + b_p + 2 - \xi_i^p \text{ and } \xi_i^p \geq 0$$

$$\text{for } i = 1,2...,n \text{ and } p \in \{1,2,...,k\} \setminus y_i,$$

where $w_p \in \Re^m$ is a vector of weights of training instances; $b \in \Re^k$; *C* is a real-valued tradeoff (cost) parameter; ξ_i^p is a penalty parameter (slack variable); and $\Phi : \Re^m \to \Re^r$ [Weston1999]. This optimization problem is quadratic in *(k-1)n* variables and *(k-1)n* constraints. Given optimal weights *w* and *b*, the following decision function is used for classification of an instance *x*:

$$f(x) = \arg\max_{p=1,...,k} [w_p^T \Phi(x) + b_p].$$

Note, that for the case *k=2*, this technique is equivalent (i.e. its hyperplane is identical) to the binary SVMs.

In our experiments we used a modified formulation of this algorithm, called *bounded formulation*, which is obtained by adding a term $\sum_{m=1}^{k} b_m^2$ to the objective function of the optimization problem stated above. By doing so, the dual formulation of

SVM problem is simplified, which can lead to easier optimization problem solvable by robust decomposition techniques (see [Hsu2002] for details).

Method by Crammer and Singer (CS). The margins between all *k* classes are optimized via solution of the following constrained QP optimization problem:

$$\min_{w,\xi} \frac{1}{2}\sum_{p=1}^{k} w_p^T w_p + C\sum_{i=1}^{n}\xi_i \quad \text{subject to}$$

$$w_{y_i}^T \Phi(x_i) - w_p^T \Phi(x_i) \geq e_i^p - \xi_i \text{ , and } \xi_i \geq 0$$

$$\text{for } i = 1,2...,n \text{ and } p \in \{1,2,...,k\} \setminus y_i ,$$

where $w_p \in \Re^m$ is a vector of weights of training instances; *C* is a real-valued tradeoff (cost) parameter; ξ_i is a penalty parameter (slack variable); $\Phi : \Re^m \rightarrow \Re^r$; and $e_i^p = \begin{cases} 0 \; if \; y_i = p \\ 1 \; if \; y_i \neq p \end{cases}$ ([Hsu2002] and [Crammer2000]). This optimization problem is quadratic in *(k-1)n* variables and only *n* constraints. Given optimal weights *w*, the following decision function is used for classification of an instance *x*:

$$f(x) = \arg\max_{p=1,...,k} w_p^T \Phi(x) .$$

Similarly to WW, in our experiments we used a bounded formulation, which can lead to a significant speed-up in the solution of the optimization problem (see [Hsu2002] for details).

References

[Burges1998] Burges, C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

[Crammer2000] Crammer, K. and Y. Singer. "On the Learnability and Design of Output Codes for Multiclass Problems", Proceedings of the Thirteen Annual Conference on Computational Learning Theory (COLT), 2000.

[Friedman1996] Friedman, J. "Another approach to polychotomous classification", Technical report, Stanford Univeristy, 1996.

[Hsu2002] Hsu, Chih-Wei and Chih-Jen Lin. "A Comparison of Methods for Multi-class Support Vector Machines", IEEE Transactions in Neural Networks 13(2) 415-425, 2002.

[Kressel1999] Kressel, U. "Pairwise classification and support vector machines", In Advances in Kernel Methods: Support Vector Learning (Chapter 15), MIT Press, 1999.

[Platt2000] Platt, J., N. Cristianini, and J. Shawe-Taylor. "Large margin dags for multiclass classification", Advances in Neural Information Processing Systems 12, pages 547-553. MIT Press, 2000.

[Vapnik1998] Vapnik, V. "Statistical Learning Theory", Wiley-Interscience, 1998.

[Weston1999] Weston, J. and C. Watkins. "Support Vector Machines for Multi-Class Pattern Recognition", Proceedings of the Seventh European Symposium On Artificial Neural Networks, 1999.

Appendix C. A solution of example cancer diagnostic problem using MC-SVMs

Consider an example diagnostic problem with three possible outcomes: T-cell acute lympboblastic leukemia (ALL), B-cell ALL, and acute myelogenous leukemia (AML), using expression levels of two genes: c-Myb and SNF2. Given a training set of patients (see **Figure C1**), our goal is to build a multicategory linear SVM diagnostic model that can be later applied to new patients. Although this example is highly simplified for clarity, the ideas generalize to all kinds of classification problems and data characteristics.

One-vs-rest (OVR). Application of the three binary OVR classifiers (**Figure C2**) yields a decision surface divided by three separate hyperplanes (dashed lines). The shaded regions in the figure correspond to tie situations when two or none classifiers are active (i.e. vote) at the same time (**Table C1**).

Consider classification of a new sample (triangular shaped in the figure) in the ambiguous region 5. This sample receives votes from both AML and ALL T-cell classifiers, however its distance from the "ALL T-cell vs. ALL B-cell and AML" hyperplane is larger than one from "AML vs. ALL B-cell and ALL T-cell" hyperplane. Hence, this sample belongs to ALL T-cell class.

The combined OVR decision function separates the decision surface by a solid bold line (**Figure C2**). Notice that the final decision function differs significantly from the original one which corresponded to the solution of k QP optimization problems.

One-vs-one (OVO). Using the OVO technique, a decision surface is divided by three separate hyperplanes (dashed lines) obtained by binary SVMs corresponding to one versus one decisions (**Figure C3**). The application of *Max Wins strategy* (**Table C2**) results in division of decision surface into three regions (separated by bold dashed lines) and the small shaded ambiguous region in the middle of the figure. The tie-breaking strategy applied to the ambiguous region produces the final decision function depicted with solid bold lines and bold dashed lines. Notice that in this example the final decision function does not differ significantly from the initial one corresponding to the solution of $\binom{k}{2}$ optimization problems.

DAGSVM. The application of DAGSVM involves training of $\binom{k}{2}$ binary OVO classifiers. For the testing phase, we follow the DDAG shown in the **Figure C4**. We note that there is no unique subdivision of the decision surface for this MC-SVM technique. Depending on the path in the DDAG (**Figure C4**), the decision surface is divided by two out of three dashed lines (i.e. binary one-versus-one classifiers) in the **Figure C3**.

Method by Weston and Watkins (WW). An example of the WW decision surface obtained for the diagnostic problem example is shown in the **Figure C5**. Note that all the classes are separated with the maximum margin.

Method by Crammer and Singer (CS). The decision surface for the CS method roughly corresponds to one for the WW, although the methods differ algorithmically (**Figure C5**).

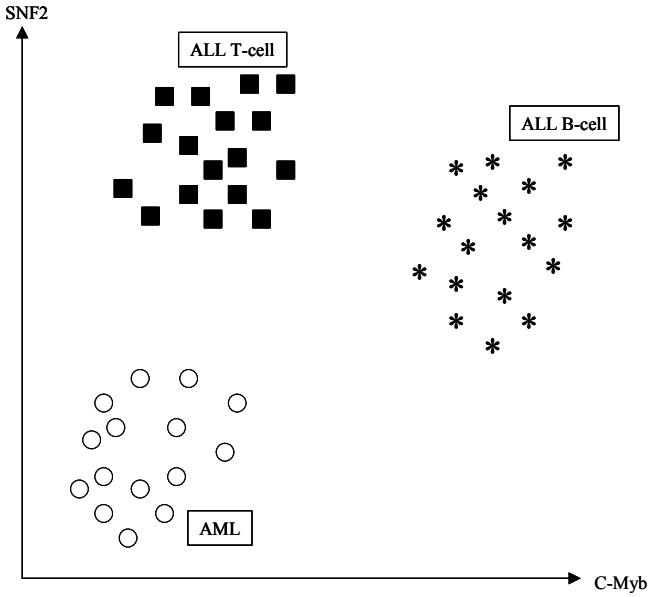

Figure C1. *Example diagnostic problem with three outcomes: ALL T-cell (■), ALL B-cell (✻), and AML (○).*

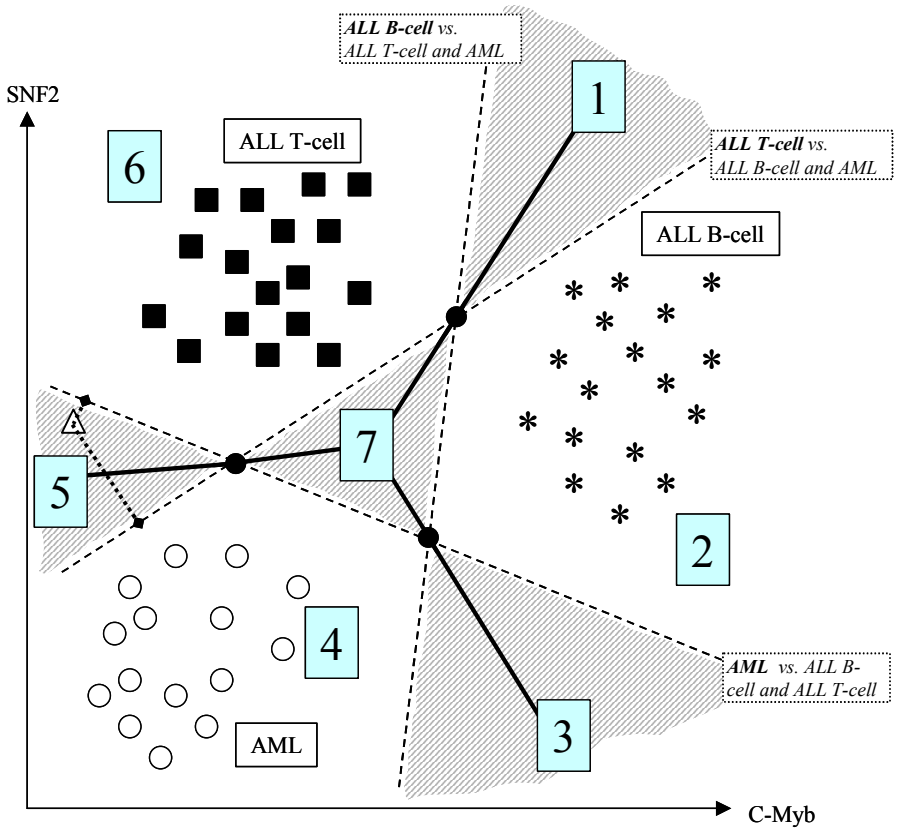

Figure C2. *OVR MC-SVM is applied to the example diagnostic problem with three outcomes: ALL T-cell (■), ALL B-cell (*), and AML (○).*

Region	Decision of the classifer			Resulting class
	AML vs. (ALL B-cell and ALL T-cell)	ALL T-cell vs. (ALL B-cell and AML)	ALL B-cell vs. (ALL T-cell and AML)	
1	-	*ALL T-cell*	*ALL B-cell*	*?*
2	-	-	*ALL B-cell*	*ALL B-cell*
3	*AML*	-	*ALL B-cell*	*?*
4	*AML*	-	-	*AML*
5	*AML*	*ALL T-cell*	-	*?*
6	-	*ALL T-cell*	-	*ALL T-cell*
7	-	-	-	*?*

Table C1. *Three binary OVR classifiers are applied to the example diagnostic problem (see **Figure C2**). The column "Resulting class" contains the resulting classification of each region. Cells with "?" correspond to tie situations when two or none classifiers are active (i.e. vote) at the same time. Please see text for how ties are resolved.*

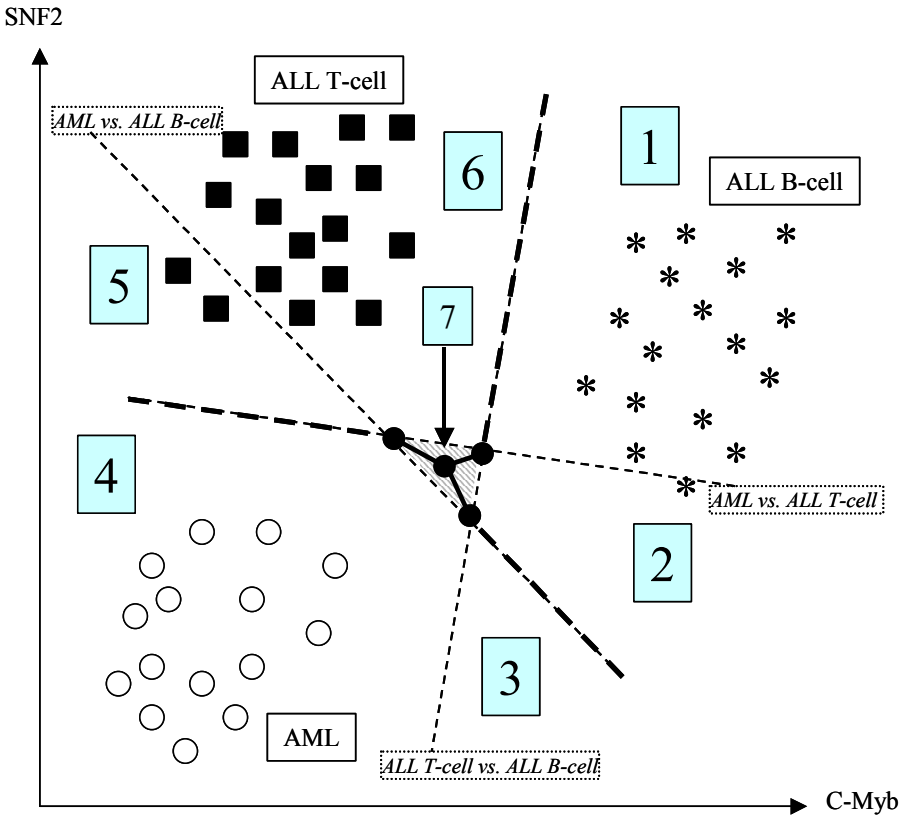

Figure C3. *OVO MC-SVM is applied to the example diagnostic problem with three outcomes: ALL T-cell (■), ALL B-cell (✻), and AML (○).*

Region	Decision of the classifier			Resulting class
	AML vs. ALL B-cell	ALL T-cell vs. ALL B-cell	AML vs. ALL T-cell	
1	***ALL B-cell***	***ALL B-cell***	*ALL T-cell*	*ALL B-cell*
2	***ALL B-cell***	***ALL B-cell***	*AML*	*ALL B-cell*
3	***AML***	*ALL B-cell*	***AML***	*AML*
4	***AML***	*ALL T-cell*	***AML***	*AML*
5	*AML*	***ALL T-cell***	***ALL T-cell***	*ALL T-cell*
6	*ALL B-cell*	***ALL T-cell***	***ALL T-cell***	*ALL T-cell*
7	*ALL B-cell*	*ALL T-cell*	*AML*	*?*

Table C2. *Three binary OVO classifiers are applied to the example diagnostic problem (see* ***Figure C3****). The column "Resulting class" contains the resulting classification of each region according to "Max Wins" strategy. Cell with "?" corresponds to tie situation when three classifiers vote at the same time for different classes). Please see text for how ties are resolved.*

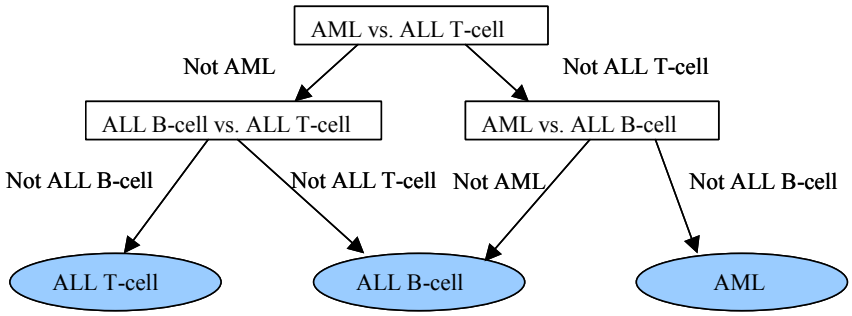

Figure C4. *DAGSVM algorithm: rooted binary decision directed tree (DDAG) for testing of unseen samples for the example diagnostic problem.*

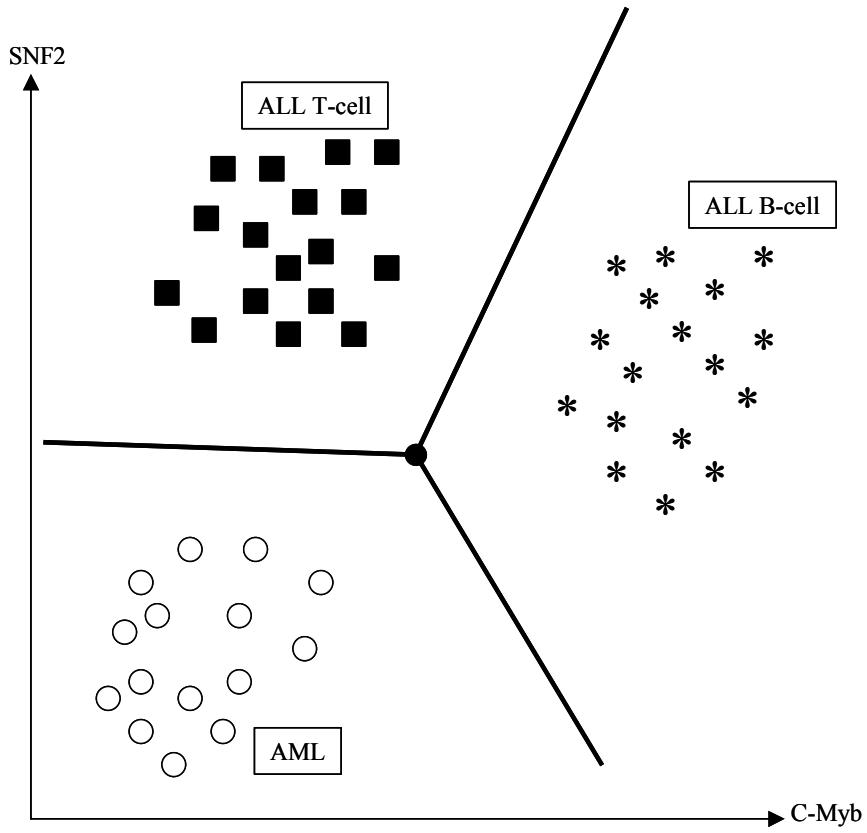

Figure C5. *WW MC-SVM is applied to the example diagnostic problem with three outcomes: ALL T-cell (■), ALL B-cell (*), and AML (○).*

Appendix D. *GEMS* user's manual

The graphics user interface of the *GEMS* system consists of a single form with a menu bar on top. Below we briefly describe the meaning of each labeled section in the form (**Figure D1**).

Section A. This section is used to specify input data files:

- Dataset (gene expression dataset in a tab/space separated ASCII format, with columns corresponding to genes/variables and rows to observations, the first column is the target variable which is encoded with integers starting from 0);
- Gene names (ASCII file with the list of gene names – one line per gene, first line is not used, line indices correspond to columns in the dataset);
- Gene accession numbers (ASCII file with the list of gene accession numbers – one line per gene, first line is not used, line indices correspond to columns in the dataset);

The user has to specify a dataset. Gene names and accession numbers (optional fields) will be used only for generation of experimental report in HTML format.

Section B. This section is used to select experimental design: either (1) N-fold cross-validation or (2) leave-one-out cross-validation (LOOCV). In case N-fold cross-validation is used, it is necessary to input the number of folds.

Section C. This section is used only when the experimental task is to estimate performance of the best model (see section M) and LOOCV design is employed. In that case, the user has to input the number of cross-validation folds in the inner loop of LOOCV design.

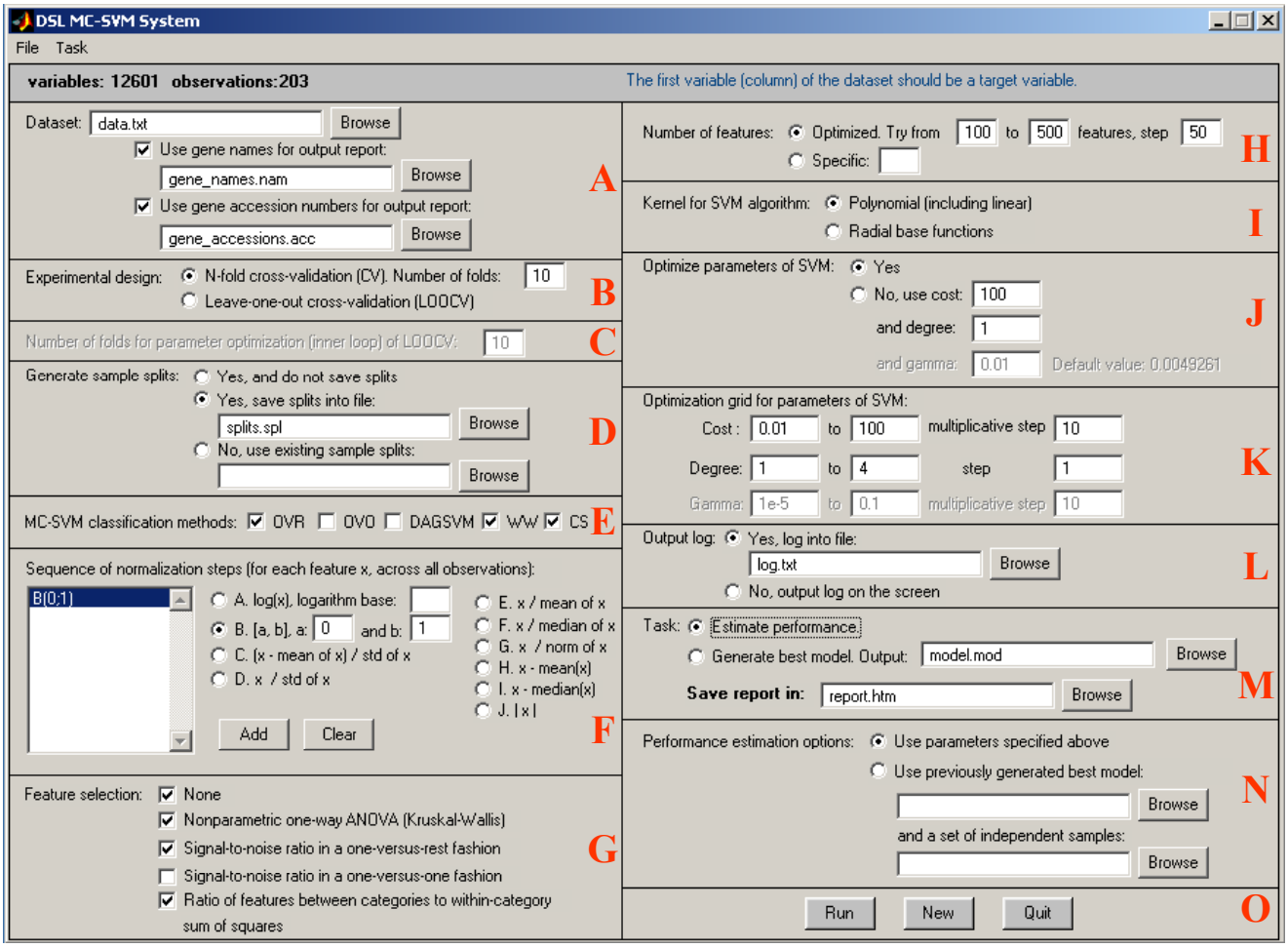

Figure D1. *Screenshot of the GEMS system. Many fields are automatically filled out with default values. Most experiments in this study can be replicated using the system with a few clicks of the mouse.*

Section D. Using this section one can either (1) generate randomly stratified sample splits for N-fold cross-validation and either discard them after experiments or save them into a file; or (2) load already generated sample splits. Sample splits are stored in an ASCII file, where each line contains indices of samples participating in a single fold (sample indices are delimited by spaces).

Section E. This section is used to select MC-SVM classification methods. If the user selects multiple classification algorithms, the system will perform optimization of the algorithms by cross-validation and derive a single algorithm yielding the largest cross-validation accuracy.

Section F. This section allows users to specify a sequence of data normalization steps for each gene x in the dataset. Normalization is always performed based solely on training dataset, so that the final results are not overfitted. It is suggested to use normalization "B" ($x \to [a, b]$ with $a = 0$ and $b = 1$) to speed up training of MC-SVM algorithms. Notice that one may need to apply normalization $|x|$ before applying $log(x)$ to ensure that the dataset does not contain negative values.

Section G. This section is used to specify gene selection algorithms. If the user selects multiple gene selection algorithms, the system will perform optimization of the algorithms by cross-validation and derive a single algorithm yielding the largest cross-validation accuracy.

Section H. If gene selection techniques are employed, this section allows selection of cardinalities of the gene subsets. One can either (1) use a gene subset of the fixed size, or (2) consider multiple gene subsets and the system will derive a single gene subset yielding the largest cross-validation accuracy.

Section I. This section is used to select a class of kernel functions for SVM algorithm – either polynomial or radial basis functions.

Section J. This section indicates if it is necessary to optimize SVM parameters by cross-validation. If optimization is not desired, the user needs to input values of cost and degree or gamma parameters. Otherwise, one has to input ranges for optimization in section K.

Section K. This section contains ranges for optimization of SVM parameters by cross-validation. The system will select a single instantiation of parameters cost and degree or gamma yielding the largest cross-validation accuracy.

Section L. This section is used to specify whether log is displayed on the screen or saved in a file.

Section M. This section is used to select an experimental task and specify the output report file. The user has two options: either (1) estimate performance, or (2) generate the best model and save it. This section also contains a field for the output report HTML file.

Section N. In case the user wants to estimate performance, this section allows either (1) to run the entire experiment with model selection (i.e. estimate performance) using Design I or II, or (2) to use already generated best model and apply it to new samples. In the latter case, one has to specify a model and a testing dataset. The testing dataset should have the same format (i.e. should contain the same variables in the same order) as the dataset in section A. If the user does not want the system to compute final accuracies, the true values of the first (target) column in the testing dataset should be substituted with an arbitrary integer number.

Section O. This is the control section of the user interface. It contains three buttons:

- Run (estimate complexity of the experiment and execute it);
- New (reset the form to default values);
- Quit.

In addition to sections described above, one can use menu bar to open and save project files which can be also created or edited with a simple text editor.

Appendix E. Supplementary Information

1. Classification without gene selection using Design II

Table E1. *Performance results (**accuracies**) without gene selection obtained using LOOCV with 10-fold cross-validation for parameter selection (Design II).*

		Multicategory classification					
Method		**9_Tumors**	**11_Tumors**	**14_Tumors**	**Brain_Tumor1**	**Brain_Tumor2**	**Leukemia1**
MC-SVM	**OVR**	**63.33%**	**95.40%**	74.35%	**91.11%**	76.00%	**97.22%**
	OVO	55.00%	89.66%	44.48%	**91.11%**	**78.00%**	**97.22%**
	DAGSVM	55.00%	89.66%	46.10%	**91.11%**	**78.00%**	95.83%
	WW	61.67%	94.25%	64.61%	**91.11%**	**78.00%**	**97.22%**
	CS	**63.33%**	94.83%	**75.32%**	90.00%	76.00%	**97.22%**
non-SVM	**KNN**	38.33%	75.29%	50.00%	86.67%	60.00%	80.56%

		Multicategory classification			*Binary classifcation*		**Averages**
Method		**Leukemia2**	**Lung_Cancer**	**SRBCT**	**Prostate_Tumor**	**DLBCL**	
MC-SVM	**OVR**	**98.61%**	96.06%	**100.00%**	**92.16%**	**97.40%**	**89.24%**
	OVO	95.83%	95.07%	**100.00%**	**92.16%**	**97.40%**	85.08%
	DAGSVM	95.83%	95.57%	**100.00%**	**92.16%**	**97.40%**	85.15%
	WW	97.22%	95.07%	**100.00%**	**92.16%**	**97.40%**	88.06%
	CS	97.22%	**96.55%**	**100.00%**	**92.16%**	**97.40%**	89.09%
non-SVM	**KNN**	86.11%	91.13%	84.34%	79.41%	88.31%	74.56%

2. Statistical comparison among classifiers

Table E2. *P-values of the statistical test that compares accuracies of all algorithms with ones of the best MC-SVM methods (CS, OVR, and WW) using classifications obtained by nested stratified 10-fold cross-validation design (Design I) for all 11 datasets <u>without gene selection</u>. Bold p-values correspond to cases when we cannot reject null hypothesis at the 0.05 level.*

Method		**Comparison: all algorithms versus**		
		OVR	**WW**	**CS**
MC-SVM	**OVR**	-	**0.981**	**0.326**
	OVO	0.016	0.009	0.012
	DAGSVM	0.009	0.011	0.009
	WW	**0.052**	-	**0.064**
	CS	**0.875**	**0.990**	-
non-SVM	**KNN**	0.006	0.005	0.004
	NN	0.001	0.002	0.001
	PNN	0.003	0.005	0.006

Table E3. *P-values of the statistical test that compares accuracies of all algorithms with ones of the best MC-SVM methods (CS, OVR, and WW) and NN using classifications obtained by nested stratified 10-fold cross-validation design (Design I) for 4 datasets (9_Tumors, 14_Tumors, Brain_Tumor1, Brain_Tumor2) <u>with gene selection</u>. Bold p-values correspond to cases when we cannot reject null hypothesis at the 0.05 level.*

Method		**Comparison: all algorithms versus**			
		OVR	**WW**	**CS**	**NN**
MC-SVM	**OVR**	-	**0.965**	**0.415**	**0.956**
	OVO	0.017	0.021	0.011	0.024
	DAGSVM	0.016	0.031	0.014	0.043
	WW	**0.071**	-	**0.052**	**0.502**
	CS	**0.751**	**0.972**	-	**0.977**
non-SVM	**KNN**	0.027	0.048	0.028	0.043
	NN	**0.083**	**0.612**	**0.072**	-
	PNN	0.018	0.028	0.020	0.027

3. Using inverse power-law curves to explain observed classification accuracy

We were interested to what extent the number of samples, categories, and variables (prior to gene selection) can explain classification accuracy observed in the datasets studied. We explored the relations between classification accuracy and the following possible performance predictors: Θ_a = number of samples; Θ_b = number of categories; Θ_c = number of variables; Θ_d = number of samples divided by number of categories; Θ_e = number of samples divided by number of variables; and Θ_f = number of samples divided by the product of number of variables times number of categories. We fitted inverse power-law curves of the type $p = 100\% - \alpha \cdot \Theta^{-\beta}$, where p is the classification performance (accuracy), Θ is one of the performance predictors, α and β are model parameters. The choice of inverse power-law curve is motivated by prior machine learning research in learning curves [Cortes1993]. The curve was fitted using the nonlinear Levenberg-Marquardt least squares iterative method available in the Matlab Optimization Toolbox [Venkataraman2002].

The inverse power-law curve for explaining performance of the best MC-SVM methods OVR, WW, and CS as a function of Θ_f (number of samples divided by the product of number of variables prior to gene selection[1] times number of categories) fitted best according to Euclidian norm metric (**Figure E1**). The resulting curve is $p = 100\% - 0.001021 \cdot \Theta_f^{-1.431305}$. According to this formula, whenever $\Theta_f > 1.6 \cdot 10^{-3}$, MC-SVM techniques CS, WW, and OVR produce multicategory cancer diagnoses with accuracies > 90%. We note that this heuristic rule did not hold in our experiments neither when gene selection was employed[2] nor when RCI performance metric was used instead of accuracy.

The analysis presented above in an initial step towards this research. It is important to note that curve fitting procedure used in this study is very simplistic since it does not incorporate predictors describing degree of biological difficulty and assumes that datasets and learning tasks used in this study are representative. More complex approaches to modeling performance of the classifier as a function of dataset characteristics may also applicable for this domain (e.g., [Mukherjee2003]).

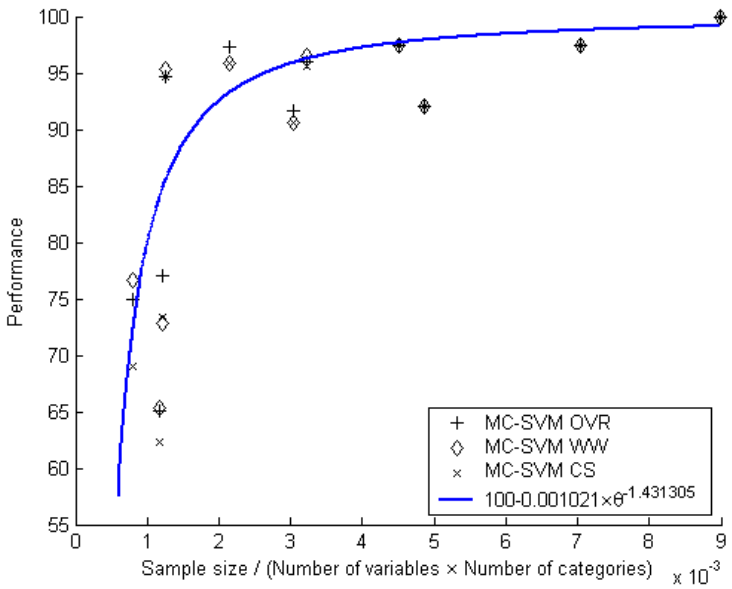

Figure E1. *Number of samples divided by the product of the number of categories times the number of variables (horizontal axis) explains observed classification accuracy of the best MC-SVM classifiers, OVR, WW, and CS (vertical axis).*

[1] The original *SRBCT* data from cDNA array with 6567 genes is not publicly available. We used a public version of this dataset with 2308 genes with red intensity greater than 20 across all samples [Khan2001]. While fitting inverse power-law curves, we treat this version of *SRBCT* dataset as if no gene selection has been performed.

[2] Given an arbitrary gene expression dataset with more samples than categories, it is always possible to select 1 best gene according to some gene scoring criterion. Hence, Θ_f will be > 1. Based on our heuristic rule, predictive accuracy should be greater than 99.99%, which obviously may not be the case.

4. Ensemble classification

Table E4. *Performance results (**accuracies**) of ensemble classification applied to outputs of learners that were used without gene selection. Ensembles were constructed both based on outputs of only MC-SVM classifiers and based on outputs of all (MC-SVM and non-SVM) classifiers. These results were obtained using nested stratified 10-fold cross-validation design (Design I). MC-SVM ensemble methods indicated with "*" were applied using extended ranges for optimization of SVM parameters: costs = {1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 10, 100, 1000, 10000} and degrees = {1,2,3,4}. MC-SVM ensemble methods indicated with "+" were used with RBF kernel and the following ranges for optimization of SVM parameters: cost = {0.0001, 0.01, 1, 100} and values of σ (RBF kernel parameter) = {0.0001, 0.001, 0.01, 1}.*

			Multicategory classification					
Input data of ensemble learner	**Method**		**9_Tumors**	**11_Tumors**	**14_Tumors**	**Brain_Tumor1**	**Brain_Tumor2**	**Leukemia1**
Outputs of MC-SVM methods	**DTs**		56.43%	93.62%	70.04%	**91.67%**	**79.50%**	**97.50%**
	MC-SVM	**OVR**	19.10%	34.02%	21.89%	78.36%	56.17%	88.93%
		OVO	32.38%	84.05%	40.04%	90.56%	72.50%	96.07%
		DAGSVM	34.05%	84.61%	39.70%	90.56%	72.50%	96.07%
		OVR*	17.19%	30.49%	23.20%	81.47%	63.83%	88.93%
		OVO*	35.48%	86.87%	50.76%	90.56%	72.50%	96.07%
		DAGSVM*	41.67%	87.37%	48.81%	90.56%	70.50%	96.07%
		OVR+	45.33%	90.27%	51.08%	90.56%	68.83%	**97.50%**
		OVO+	59.00%	92.04%	57.14%	90.56%	70.33%	96.07%
		DAGSVM+	59.00%	92.04%	54.90%	90.56%	68.33%	96.07%
	Majority voting		63.90%	94.68%	70.57%	90.56%	75.33%	**97.50%**
Outputs of all methods	**Majority voting**		61.90%	93.04%	67.06%	**91.67%**	77.83%	**97.50%**
Best results of non-ensemble classifiers			**65.33%**	**95.30%**	**76.60%**	**91.67%**	77.83%	**97.50%**

			Multicategory classification			***Binary classifcation***	
Input data of ensemble learner	**Method**		**Leukemia2**	**Lung_Cancer**	**SRBCT**	**Prostate_Tumor**	**DLBCL**
Outputs of MC-SVM methods	**DTs**		**97.32%**	96.05%	**100.00%**	**92.00%**	**97.50%**
	MC-SVM	**OVR**	76.43%	84.76%	86.63%	**92.00%**	**97.50%**
		OVO	94.46%	95.55%	**100.00%**	**92.00%**	**97.50%**
		DAGSVM	94.46%	95.55%	**100.00%**	**92.00%**	**97.50%**
		OVR*	82.14%	85.21%	86.63%	**92.00%**	**97.50%**
		OVO*	94.46%	96.05%	**100.00%**	**92.00%**	**97.50%**
		DAGSVM*	94.46%	96.05%	**100.00%**	**92.00%**	**97.50%**
		OVR+	**97.32%**	96.05%	**100.00%**	**92.00%**	**97.50%**
		OVO+	94.46%	96.05%	**100.00%**	**92.00%**	**97.50%**
		DAGSVM+	94.46%	96.05%	**100.00%**	**92.00%**	**97.50%**
	Majority voting		95.89%	96.05%	**100.00%**	**92.00%**	**97.50%**
Outputs of all methods	**Majority voting**		95.89%	95.09%	**100.00%**	**92.00%**	**97.50%**
Best results of non-ensemble classifiers			**97.32%**	**96.55%**	**100.00%**	**92.00%**	**97.50%**

Table E5. *Performance results (**accuracies**) of ensemble classification applied to outputs of learners that were used with gene selection on three datasets: 9_Tumors*, *Brain_Tumor1*, and *Brain_Tumor2. Ensembles were constructed both based on outputs of only MC-SVM classifiers and based on outputs of all (MC-SVM and non-SVM) classifiers. These results were obtained using nested stratified 10-fold cross-validation design (Design I).*

Input data of ensemble learner	Method	9_Tumors	Brain_Tumor1	Brain_Tumor2
Outputs of MC-SVM methods	DTs	66.19%	88.31%	79.17%
	Majority voting	71.52%	90.42%	84.00%
Outputs of all methods	DTs	59.52%	88.44%	73.50%
	Majority voting	68.43%	90.42%	82.00%
Best results of non-ensemble classifiers		*74.86%*	*92.67%*	*85.67%*
FS method and number of features		*BW, 1000*	*KW, 500*	*KW, 500*

5. Comparison with previously published results

Table E6. *Comparison of classification results obtained in the present study with previously published studies on the same datasets. If multiple studies were present for a dataset, we selected one that employed the most similar learning task and followed the most similar experimental design compared to our study.*

	Our study			Published studies			
Dataset	**Accuracy without gene selection**		**Accuracy with gene selection**	**Major differences in dataset preparation between our study and published studies**	**Experimental design**	**Methods and Results**	**Reference**
	Design I	**Design II**	**Design I**				
9_Tumors	**65.33%**	**63.33%**	**74.86%**	This dataset was used for a different task - prediction of chemosensitivity			[Staunton2001]
11_Tumors	**95.30%**	**95.40%**	**Not analyzed**	The dataset used by [Su2001] contains 1 more sample compared to our study. That sample was not included in the publicly available version of the dataset.	The study performed LOOCV on 100 samples (training set) and then tested the classifier on 75 samples (testing set).	**Methods:** Gene selection procedure involved the following three major steps: (1) minimal thresholding of gene expression data; (2) selection of genes with small p-values according to Wilcoxon test; (3) further gene selection by use of SVMs. The study applied a variant of MC-SVM OVR method for classification. **Results:** The study achieved **97%** accuracy for LOOCV on the training set (100 samples) and **95%** accuracy on the testing set (75 samples).	[Su2001]
14_Tumors	**76.60%**	**75.32%**	**76.60%**	Compared to our study, the dataset used in [Ramaswamy2001] did not contain 90 samples from normal tissues and spanned only over 14 diagnostic categories (not 26 as used in our study). Furthermore, [Ramaswamy2001] excluded 20 "*poorly differentiated samples*" from the main testing set, although their technical quality was "indistinguishable from the other samples in the study". Unlike [Ramaswamy2001], we treated all testing samples equally and included these 20 samples in the testing set.	The study performed LOOCV on 144 samples (training set) and then tested the classifier on two testing sets - 54 samples (main testing set) and 20 samples ("*poorly differentiated*" testing set).	**Methods:** Genes were ranked and selected according to their contribution to the solution of classification problem by SVMs. The study employed MC-SVM methods (OVO and OVR) and variants of KNN and WV algorithms for classification. **Results:** Best classification results were obtained without gene selection and using MC-SVM OVR classifier: **78%** accuracy for LOOCV on the training set (144 samples), **78%** accuracy on the main testing set (54 samples), and **30%** accuracy on the "*poorly differentiated*" testing set (20 samples). Classification results with gene selection were inferior compared to results obtained by utilizing all genes.	[Ramaswamy2001]
Brain_Tumor1	**91.67%**	**91.11%**	**92.67%**	Unlike [Pomeroy2002], where researchers experimented only with a 42 sample-dataset, our study used a version of this dataset with 90 samples.	The study performed LOOCV on 42 samples.	**Methods:** Genes were selected with S2N metric. KNN algorithm was used for classification. **Results:** The study obtained **83.33%** accuracy.	[Pomeroy2002]
Brain_Tumor2	**77.83%**	**78.00%**	**85.67%**	Unlike our study which solved this classification problem with 4 specific diagnostic categories, [Nutt2003] solved this problem with 2 more general categories.	The study performed LOOCV on 21 samples (training set) and then tested the classifier on 29 samples (testing set).	**Methods:** Gene selection procedure involved the following two steps: (1) minimal thresholding of gene expression data; (2) selection of genes with S2N metric. KNN algorithms was used for classification. **Results:** The study achieved **86%** accuracy for LOOCV on the training set (21 samples) and **59%** accuracy on the testing set (29 samples).	[Nutt2003]

This table is continued on the next page.

This table begins on the previous page.

	Our study			Published studies			
Dataset	**Accuracy without gene selection**		**Accuracy with gene selection**	**Major differences in dataset preparation between our study and published studies**	**Experimental design**	**Methods and Results**	**Reference**
	Design I	**Design II**	**Design I**				
DLBCL	**97.50%**	**97.40%**	**Not analyzed**	No differences	The study performed LOOCV on 77 samples.	**Methods:** Genes were selected with S2N metric. Classification was performed with WV algorithm. **Results:** The study obtained **92%** accuracy.	[Shipp2002]
Leukemia1	**97.50%**	**97.22%**		No differences	The study performed LOOCV on 34 out of 72 samples.	**Methods:** Gene selection procedure involved the following two steps: (1) minimal thresholding of gene expression data; (2) gene selection with BW metric. [Lee2003] used MC-SVM MSVM algorithm for classification. **Results:** The study obtained **97%** accuracy.	[Lee2003]
Leukemia2	**97.32%**	**98.61%**		The dataset used by [Armstrong2002] contains 5 less samples compared to our study (and publicly available version of the data).	The study performed LOOCV on 57 samples (training set) and then tested the classifier on 10 samples (testing set).	**Methods:** Gene selection was performed with S2N metric. KNN algorithm was used for classification. **Results:** The study achieved **95%** accuracy for LOOCV on the training set (57 samples) and **90%** accuracy on the testing set (10 samples).	[Armstrong2002]
Lung_Cancer	**96.55%**	**96.55%**		[Aliferis2003a] solved the following three binary classification tasks: *Task 1* - normal versus cancerous (203 samples); *Task 2* - adeno versus squamous (160 samples); and *Task 3* - metastatic adeno versus non-metastatic adeno (139 samples).	The study used nested stratified N-fold cross-validation design (N = 5, 5, and 7 for tasks 1, 2, and 3, respectively).	**Methods:** Gene selection was performed by SVM-based recursive feature elimination and univariate attribute filtering (UAF). Classification algorithms SVM, Neural Networks, and KNN were used. **Results:** The study achieved **99.64%**, **99.07%**, and **96.83%** area under ROC curve (AUC) for experiments without gene selection for tasks 1, 2, and 3, respectively. When gene selection was performed, the study achieved **99.80%**, **99.63%**, and **97.62%** AUC for tasks 1, 2, and 3, respectively.	[Aliferis2003a]
Prostate_Tumor	**92.00%**	**92.16%**		No differences	The study performed LOOCV on 102 samples.	**Methods:** Gene selection was performed with S2N metric. KNN algorithm was used for classification. **Results:** The study achieved **94%** accuracy.	[Singh2002]
SRBCT	**100.00%**	**100.00%**		The dataset used by [Khan2001] contains 5 more non-SRBCT samples compared to our study.	The study performed LOOCV on 63 samples (training set) and then tested the classifier on 25 samples, including 5 non-SRBCT samples (testing set).	**Methods:** Minimal thresholding was applied to gene expression data. PCA was used to reduce dimensionality to 10 first principal components. Gene selection was performed by measuring sensitivity of the outputs with respect to inputs. Neural Networks were used for classification. **Results:** The study achieved **100%** accuracy for LOOCV on the training set (63 samples) and **100%** accuracy on the testing set (25 samples).	[Khan2001]

6. Classification results with Decision Trees and Weighted Voting

Table E7. *Performance results (**accuracies**) without gene selection obtained using a nested stratified 10-fold cross-validation design (Design I) for DT, WV OVR, and WV OVO classifiers. These results are further improved by gene selection (see **Table E8**). The last column in the bottom table reports average performance computed over datasets.*

		Multicategory classification					
Method		**9_Tumors**	**11_Tumors**	**14_Tumors**	**Brain_Tumor1**	**Brain_Tumor2**	**Leukemia1**
non-SVM	**DT**	28.29%	68.82%	53.23%	70.42%	63.17%	83.39%
	WV OVR	30.14%	30.52%	25.69%	22.14%	47.00%	18.04%
	WV OVO	25.29%	34.22%	28.74%	26.86%	36.83%	63.57%

		Multicategory classification			*Binary classifcation*		**Averages**
Method		**Leukemia2**	**Lung_Cancer**	**SRBCT**	**Prostate_Tumor**	**DLBCL**	
non-SVM	**DT**	85.00%	86.14%	82.74%	89.00%	88.21%	72.58%
	WV OVR	41.79%	5.45%	23.65%	59.73%	75.36%	34.50%
	WV OVO	34.64%	40.24%	53.83%	59.73%	75.36%	43.57%

Table E8. *Performance results (**accuracies**) with gene selection obtained using a nested stratified 10-fold cross-validation design (Design I) for DT, WV OVR, and WV OVO classifiers and two datasets, Brain_Tumor1 and Brain_Tumor2.*

Method		**Brain_Tumor1**	**Brain_Tumor2**
non-SVM	**DT**	75.89%	72.67%
	WV OVR	34.44%	47.00%
	WV OVO	47.31%	36.83%

References

[Aliferis2003a] Aliferis C.F., I. Tsamardinos, P. Massion, A. Statnikov, N. Fananapazir, D. Hardin. "Machine Learning Models For Classification Of Lung Cancer and Selection of Genomic Markers Using Array Gene Expression Data" In Proceedings of the 16th International Florida Artificial Intelligence Research Society (FLAIRS) Conference, 2003.

[Armstrong2002] Armstrong, S., et al. "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia", Nature Genetics, volume 30, January 2002.

[Cortes1993] Cortes, C., Lawrence D. Jackel, Sara A. Solla, Vladimir Vapnik, John S. Denker: Learning Curves: Asymptotic Values and Rate of Convergence. NIPS 1993: 327-334.

[Khan2001] Khan, J., et al. "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks", Nature Medicine, volume 7, Number 6, June 2001.

[Lee2003] Lee, Yoonkyung and Cheol-Koo Lee. "Classification of multiple cancer types by multicategory support vector machines using gene expression data", Bioinformatics 2003 19: 1132-1139.

[Mukherjee2003] Mukherjee, S., Pablo Tamayo, Simon Rogers, Ryan Rifkin, Anna Engle, Colin Campbell, Todd R Golub, Jill P Mesirov, "Estimating dataset size requirements for classifying DNA microarray data", Journal of Computational Biology, 10(2):119-142, 2003.

[Nutt2003] Nutt, C., et al. "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification", Cancer Res. 2003 Apr 1;63(7):1602-7.

[Pomeroy2002] Pomeroy, L., et al. "Prediction of central nervous system embryonal tumour outcome based on gene expression", Nature, vol 415, 24 January 2002.

[Ramaswamy2001] Ramaswamy, S., et al. "Multiclass cancer diagnosis using tumor gene expression signatures", Proc Natl Acad Sci U S A Dec 11, 2001.

[Singh2002] Singh, D., et al. "Gene expression correlates of clinical prostate cancer behavior", Cancer Cell : March 2002, Vol. 1

[Shipp2002] Shipp, M., et al. "Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning", Nature Medicine, Volume 8, Number 1, January 2002.

[Staunton2001] Staunton, J., et al. "Chemosensitivity prediction by transcriptional profiling", PNAS, September 11, 2001, vol. 98, no. 19, 10787-10792.

[Su2001] Su, A.I., et al. "Molecular classification of human carcinomas by use of gene expression signatures", Cancer Res. 2001 Oct 15;61(20):7388-93.

[Venkataraman2002] Venkataraman, P. Applied Optimization with MATLAB Programming. John Wiley & Sons, Inc., 2002.