Abstract: The paper presents a novel learning algorithm for the class of L2 Support Vector Machines classifiers dubbed Direct L2 SVM. The proposed algorithm avoids solving the quadratic programming problem and yet, it produces both the same exact results as the classic quadratic programming based solution in a significantly shorter CPU time. The connections between various L2 SVM algorithms will be highlighted and some geometric properties of the Direct L2 SVM will be pointed at. All the other known L2 based SVMs can be looked at as the special cases of a Direct L2 SVM. The developed Direct L2 SVM algorithm is posed as the Non-Negative Least Squares problem which solves the comprehensive L2 SVM exactly and, in a striking difference to both the Least Squares SVM and proximal SVM, is able to produce the sparse solutions.

Keywords: Direct L2 SVM, DL2 SVM, Non-Negative Least Squares, NNLS, Enclosing Ball, Interpolating Ball.
While the change seems to be minor the consequences are significant. First and foremost, the L2 formulation may be transformed into an efficient linear system of equations problem and due to the avoiding solving the Quadratic Programming (QP) problem L2 SVMs learn faster and produce similar accuracies as the L1 SVM. The most related work to the ideas presented here are the SVM based approaches and models derived as the least squares (LS) SVM [5], proximal SVM [6] and various 'geometric' approaches related either to L1 SVMs [7–8], or to L2 SVMs [9–14]. Very well documented comparisons between performances of the L1 and L2 SVMs can be found in [4]. For the sake of comprehensiveness and in order to have a common model from which all the L2 SVM approaches above can be derived, we present the most general L2 SVM learning task as follows

$$\underset{w, b, \rho, \zeta}{\text{arg min}} \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{i=1}^{m} \sigma_i^2 + \frac{k}{2} b^2 - k_{\rho} \rho,$$

subject to,

$$y_i (\langle \phi(x_i), w \rangle + b) \geq \rho - \zeta_i, \quad i = 1, \ldots, m.$$  

One can easily notice the addition of bias term $b$ and parameter $\rho$ in the cost function of a comprehensive L2 SVM problem setting as well as having $\rho$ on the right hand side of the constraints (6). All the other L2 SVMs (mentioned above and well accepted in practice) models can be looked at as the special cases of the learning task given in (5) and (6).

More specifically,

- the LS SVM minimizes (5) with both $k_b = 0$ and $k_{\rho} = 0$, while the linear inequality constraints (6) are replaced with equality constraints. Also, instead using the variable parameter $\rho$ subject to the optimization on the right hand side of (6), LS SVM works with a fixed $\rho = 1$. Such an approach leads to solving a system of linear equations of the size $(m+1, m+1)$ and produces non-sparse, i.e., dense, solutions. In other words, it produces a model in which all the data are support vectors. (Below, we will call such a solution affine solution, see Fig. 1).

- the proximal SVM poses the learning problem similar to LS SVM with only difference being the addition of the bias term $b^2$ to the cost function (5) with $k_b = 1$ while, same as in LS SVM, $k_{\rho} = 0$. It also uses $\rho = 1$ on the right hand side of (6) and it replaces the inequality constraints in (6) by the equality ones too. Thus, same as above, one has to solve a system of linear equations which again leads to the dense solutions.

- several 'geometric' approaches presented in [10–14] dubbed as core, ball, sphere and minimal norm (MN) SVMs have identical posing of the L2 SVM problem as given above. They are minimizing (5) with $k_b = 1$ and $k_{\rho} = 1$ and they are using the same set of inequality constraints as given in (6). All these approaches are equivalent to the geometric problems of finding minimal enclosing ball in the feature space defined by $\phi(x_i)$ which will be defined below or, for L1 SVMs, of finding the closest point of the convex hull to the origin [8–13]. Because of their iterative nature in obtaining the solution (meaning, at no point they need to have a Hessian matrix for all the training data calculated), they are well suited for solving large classification problems. As of now, it seems that the MN SVM discovered and presented in [12–13] has an edge in terms of both a speed and the best scalability.

In what follows, we are going to present the direct, non-iterative, algorithm for finding exact solution to general L2 SVM problem as posed in (5) and (6).

## 2 The Comprehensive L2 SVM Learning Problem Solution

The first step in solving the comprehensive L2 SVM learning problem given in (5) and (6) will be the standard step of forming the primal Lagrangian as follows

$$L_w(w, b, \rho, \zeta, \alpha) = \frac{1}{2} \|w\|^2 + \frac{k}{2} b^2 - k_{\rho} \rho + C \sum_{i=1}^{m} \sigma_i^2 - \sum_{i=1}^{n} \alpha_i \left( y_i (\langle \phi(x_i), w \rangle + b) - \rho - \zeta_i \right), \quad \alpha_i \geq 0, \quad i = 1, \ldots, m.$$  

Next, the derivatives of $L_w$ in respect to $w$, $b$, $\rho$, $\zeta$ and $\alpha_i$ are equaled to 0, which after a simple rearrangements leads to,

$$w = \sum_{i=1}^{m} \alpha_i y_i \phi(x_i),$$

$$b = \frac{1}{k_b} \sum_{i=1}^{m} \alpha_i y_i,$$

$$\sum_{i=1}^{m} \alpha_i = k_{\rho},$$

$$\zeta_i = \frac{\alpha_i}{C},$$

$$y_i \left( \sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b \right) + \zeta_i = \rho.$$
The crucial step in deriving the direct model for solving the comprehensive L2 SVM problem is plugging in equations (8)–(9), and (11) into the equation (12) and dividing the resulting expression by a scalar variable $\rho$ as proposed in [14]. This introduces the new dual variables $\beta_i$ defined as

$$\beta_i = \alpha_i / \rho.$$ (13)

These steps transform the L2 SVM QP learning problem given by (5)–(6) into the $(m, m)$ matrix equation

$$K \beta = 1,$$ (14)

subject to

$$\beta_i \geq 0, \; i = 1, \ldots, m,$$ (15)

which is the classic non-negative least squares (NNLS) problem. The positive definite matrix $K_\beta$ is defined as shown below

$$K_\beta = \left[ K + \frac{1}{k_0} \mathbf{1}_{m \times m} + \text{diag}_{m \times m}(\frac{1}{C}) \right] \cdot \mathbf{y} \mathbf{y}^T,$$ (16)

and the entry $k(x_i, x_j)$ of the kernel matrix $K$ corresponds to the scalar product of the original data images $\phi$ in the feature space defined as

$$k(x_i, x_j) = \left[ \phi(x_i)^T \phi(x_j) \right]_{i=1, \ldots, m, j=1, \ldots, m}.$$ (17)

Subscript $f$ in $K_\beta$ denotes the fact that the matrix $K_\beta$ is related to the feature space in which $\hat{k}(x_i, x_j) = \hat{\phi}(x_i)^T \hat{\phi}(x_j)$ and, that it differs from a standard kernel matrix $K$ related to the space defined by $\phi(x)$. As already mentioned above, the direct algorithm for solving (14) and (15) replaces the QP problem defined by (5) and (6) by the NNLS solution of a linear system of equations given by (14). Once the non-negative $\beta_i$ values have been calculated, the value of a parameter $\rho$ can be readily found by combining (10) and (13) as

$$\rho = \frac{k}{\sum_{i=1}^{m} \beta_i^2}.$$ (18)

Finally, the dual variables $\alpha_i$ of a comprehensive L2 SVM model are computed as

$$\alpha_i = \beta_i \rho.$$ (19)

In conclusion, instead of solving the QP problem (5)–(6) one solves a NNLS problem defined by (14)–(15). Sure, let's remind that the NNLS problem posed above by (14)–(15) can also be posed as the QP problem

$$\min \frac{1}{2} \beta^T K_\beta \beta - 1^T \beta,$$ (20)

subject to

$$\beta_i \geq 0, \; i = 1, \ldots, m,$$ (21)

but solving the DL2 SVMs task by the QP algorithm might be too slow and just prohibitive for very large datasets. Having $\alpha_i$ calculated by (19) and $b$ by the use of (9), given an input sample $x$, the DL2 SVM model's output $o(x)$ is computed by using kernel $k$ (and not $\hat{k}$) as follows,

$$o(x) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x) + b = \sum_{n=1}^{s\text{Support}} \alpha_i y_i k(x_i, x) + b.$$ (22)

There are several remarks which should be done here. First, the NNLS algorithm complexity scales with the number of the non-negative $\beta_i$ (i.e., $\alpha_i$) values and not with the number of data. In the terminology of SVMs, the complexity of the DL2 SVM (NNLS based) algorithm, scales with the number of support vectors. This is a very good news when facing ultra-large datasets. Hence, when a percentage of support vectors is small, a significant speed up can be expected for large datasets. Next, NNLS solvers find the exact solutions and they are not iterative in the nature. However, they are solving the system of equations in a repetitive manner where the previous solution can be used in updating the new one.

2.1 Geometric Insights and Some Comments and Comparisons with Other L2 SVM Models

Recently several publications (including the major ones related to this work referred above [10–14]) have been pointing at the similarity between solving some geometrical problems and L2 SVM tasks. These investigations have been driven by the desire to develop learning algorithms which can handle ultra-large datasets. They have succeeded in the sense that the geometric approaches have shown better scalability than all the other SVMs notably LS, proximal and the classic L1 SVMs. In the essence, all the mentioned geometric approaches related to L2 SVMs are pointing to the fact that finding dual variables equals to finding enclosing/interpolating sphere in kernel induced feature space around the image points $\hat{\phi}(x_i)$. Similar to [15], the problem is visualized below in Fig. 1 and all the objects shown will be used in showing the similarities and differences between various L2 SVM problem settings introduced above.
After the mapping into a feature space all the points are mapped onto the hypersphere centered at the origin \( \mathbf{O} \) of the feature space with the radius \( R = \sqrt{k(x, x)} \). This sphere is not completely shown in Fig. 1 and only a part of the sphere with a radius \( R \) is shown by a dotted circular arc. In the case of Gaussian kernel and for the most commonly used coefficients \( k_b = 1 \) and \( k_\rho = 1 \) the data will be mapped onto the hypersphere having radius

\[
R = \sqrt{2 + 1/C}
\]

in all classification problems. Hence, the radius \( R \) of the hypersphere in the feature space doesn't depend upon any particular dataset. Also, one can readily see that the solutions for \( c_{CH} \) and \( c_{AFF} \) are the closest points in either the convex hull (\( c_{CH} \)) or in the affine space (\( c_{AFF} \)) of the image points \( \hat{\phi}(\mathbf{x}_i) \) to the origin \( \mathbf{O} \).

Below, some basic findings about the equality and/or similarity of various approaches (i.e., learning algorithms) for solving L2 SVM problem will be pointed at and commented.

**Theorem 1**: The learning algorithms presented in [10–14] and dubbed core, ball, sphere and minimal norm (MN) SVMs have the same solutions as the DL2 SVM.

**Proof**: The cost function (5) and constraints (6) in all models are equal. All algorithms in [10–14] are iterative with the proof of a convergence to the unique optimal solution. The matrix \( \mathbf{K}_f \) of the DL2 SVM learning problem derived from (5)–(6) is positive definite, i.e., singular, which guarantees finding the same unique solution by NNLS too.

**Comment**: The detailed and extensive simulation comparisons of all the models given in [13] for various large real benchmarking datasets within the experimental framework of a double crossvalidation confirm the Theorem 1.

**Theorem 2**: LS SVM is the affine solution of the L2 SVM learning task, which finds the center \( c_{AFF} \) of the interpolating hypersphere for points \( \hat{\phi}_i \).

**Proof**: LS SVM solves a system of linear equations with a positive definite matrix exactly which means that the solution \( c_{AFF} \) is a linear combination of points \( \hat{\phi}_i \) or that, according to [10–14], the hyper-sphere centered at \( c_{AFF} \) interpolates all the points \( \hat{\phi}_i \) (as depicted in Fig. 1, right).

**Corollary 1**: LS SVM can't obtain sparse solution due to the fact that it replaces the inequality constraints (6) by the equality ones which enables/allows the dual variables \( \alpha_i \) of the LS SVM problem be negative. In other words, the linear combination of feature vectors \( \hat{\phi}_i \) can get outside the convex hull. In addition, there is no bias \( b \) term in the cost function of the LS SVM and therefore, \( b \) can't be eliminated from the final solution in the learning phase of LS SVM either. Hence, LS SVM system of equality equations must be solved by a regular inversion always resulting in a full dense (here called affine) solution for the sphere centered at \( c_{AFF} \) which interpolates points \( \hat{\phi}_i \) in the feature space. LS SVM arrives at dense solution even in the cases when points \( \hat{\phi}_i \) don't enclose the affine center (situation shown on the left in Fig. 1). It is different for DL2 SVM where, the use of the bias term \( b \) in the cost function and keeping (6) as a set.
of inequality equations enables both an elimination of $b$ during the learning phase and a formulation of the novel DL2 SVM problem which for the calculation of dual variables uses NNLS and (depending upon the values of hyperparameters) is able to produce sparse solution.

**Corollary 2**: For the very small values of $k_0$ (say, whenever $k_0 < 1e^{-5}$) the dual variables $\beta$ (in DL2 SVM) and $\alpha$ (in L2 and LS SVM) are numerically equal as long as the points $\phi_i$ enclose the center of an affine space $c_{aff}$. This is the case when the convex hull solution $c_{aff}$ coincides with the affine solution $c_{aff}$. Stating it differently, they are equal as long as all the data are support vectors. The only difference is that, in the LS SVM model, $\alpha_i$ corresponding to negative class have a negative sign. Remind that the decision function value $\alpha(x)$ for DL2 and L2 SVMS is obtained by $\alpha_i$ multiplied by $y_i$ and in this way this sign difference will be compensated. Note also that, the 'true' dual variables $\alpha_i$ of the DL2 SVMs problem are $\beta_i$ scaled by $\rho$ defined in (18). In other words, as long as all the data are support vectors (and, when the values of $k_0 \leq 1e^{-5}$) the connections between dual variables in L2, LS and DL2 SVMs are related as,

$$\alpha_{(l2, ls)} = \frac{\alpha_{(gsvm)}}{\rho}$$

When using a common value $k_0 = 1$, the $\beta_i$ values obtained by the DL2 SVM are slightly different in respect to the solutions for $\alpha_i$ obtained in LS and L2 SVM. This is a consequence of the fact that in the cost function (5) the norm of both the weight vector $w$ and bias term $b$ are minimized, while the L2 and LS SVM cost function (3) is minimizing $||w||^2$ only.

**Remark 1**: For small values of both Gaussian variance (i.e., its shape parameter $\sigma$) and penalty parameter $C$ original points are mapped onto the affine space enclosing the $c_{aff}$ (as depicted on the right graph in Fig. 1) and solutions for $\beta_i$ of DL2, $\alpha_i$ for LS and standard $\alpha_i$ solutions for L2 SVMS are numerically equal (except for the negative signs in LS SVM as commented in the previous paragraph). As both the shape parameter $\sigma$ and penalty $C$ increase the points of positive class (shown as red circles in Fig.1) and negative class (blue point) converge to the positive and negative 'pole' of the affine space and cease to enclose the center of the affine space which is moving closer toward origin $O$ but it is never reaching it. This leads to the sparse solution for both DL2 and L2 SVMs (or, to the $enclosing$ ball centered at $c_{aff}$) and the following still holds $\alpha_{(l2)} = \alpha_{(ls)}/\rho$ for these two models. (However, even for larger $\sigma$ values the LS SVM still finds the affine center $c_{aff}$ as the non-sparse solution of LS SVM system of equations).

### 2.2 Few More DL2 SVM Models

The DL2 SVM setting given by (5) and (6) is just one possible posing of the problem. Another one is obtained by excluding the parameter $\rho$ from the optimization part. This means that the last term in (5), $-k_0 \rho$, doesn't exist and that the inequality constraints are given as $y_i (\phi(x)^T w + b) \geq 1 - \zeta_i, \ for \ i = 1, 2, \ldots, m$. Following the derivations above, it can be readily shown that the model of DL2 SVM without using $\rho$ is almost equal to the DL2 SVM (14)–(15) and it is

$$K_\alpha \alpha = 1,$$  \hspace{1cm} (25)

subject to

$$\alpha_i \geq 0, \ for \ i = 1, 2, \ldots, m.$$  \hspace{1cm} (26)

Next, it is very well known that for the positive definite kernels one doesn’t have to use a bias term $b$ and the DL2 SVM without using both bias $b$ and parameter $\rho$ is shown below,

$$K_{\alpha} \alpha = 1,$$  \hspace{1cm} (27)

subject to

$$\alpha_i \geq 0, \ for \ i = 1, 2, \ldots, m,$$  \hspace{1cm} (28)

and $K_{\alpha} = [K + diag(1/C)] * YY^T$.

If $\rho$ was used, there would be $\beta$ (as defined in (13)) instead of $\alpha$ in (27) but the matrix $K_{\alpha}$ would not change.

### 2.3 The Very First Experiments, Comparison Results and Comments

The solutions of all the 'geometric' SVM models presented in [10–14] and the DL2 SVM proposed here are equal. See extensive experimental results for various big datasets in [11–13]. As for the CPU time needed to get them, the first simulation runs within the Matlab computing environments are strongly supporting the claims made regarding the CPU time needed to arrive at the solution of DL2 SVM in the Section 2.1 above. The CPU time for solving binary classification problems for datasets ranging from 10 to 1,000 samples by using NNLS for solving (14)–(15) and QUADPROG for solving (20)–(21) are shown in Fig.2. Curves show that the bigger the data sample, the bigger speed up obtained by using NNLS is. The graph is just a hint about a possible speed up because it is well known that Matlab's QP routine is not known for a speed. Right now, we are developing a new DL2 SVM code which will be an extension of an efficient program called *gsvm* implementing MN SVM algorithm developed within [12–13]. Once this is done an extensive CPU times comparisons of DL2 SVM and the other algorithms on big real datasets will be performed.
3 Conclusions

A novel SVM algorithm which replaces solving a QP problem by NNLS solutions of a system of linear equations dubbed a Direct L2 SVM is proposed. This leads to a significant speed up of SVMs training phase within the Matlab environment. The paper is a contribution to the theory and few ideas are presented for possible future theoretical and/or experimental investigations. The DL2 SVM problem setting (cost function and constraints) is same as the ones for the core, ball, sphere and MN SVMs. Consequently, the DL2 SVM arrives at the same solution as these geometric approaches. The geometric insights into the (theoretically) infinite feature space may help in both a better understanding of the connections between various L2 SVMs models and in finding faster learning algorithms in the future. Some relations between various L2 SVM algorithms are pointed at. The major advantage of using NNLS for finding the dual variables of an L2 SVM for large data sets originates from its possible parallelization and the use of Cholesky factorization and/or nonnegative conjugate gradient in solving (14)–(15) which is the subject of present research.

References