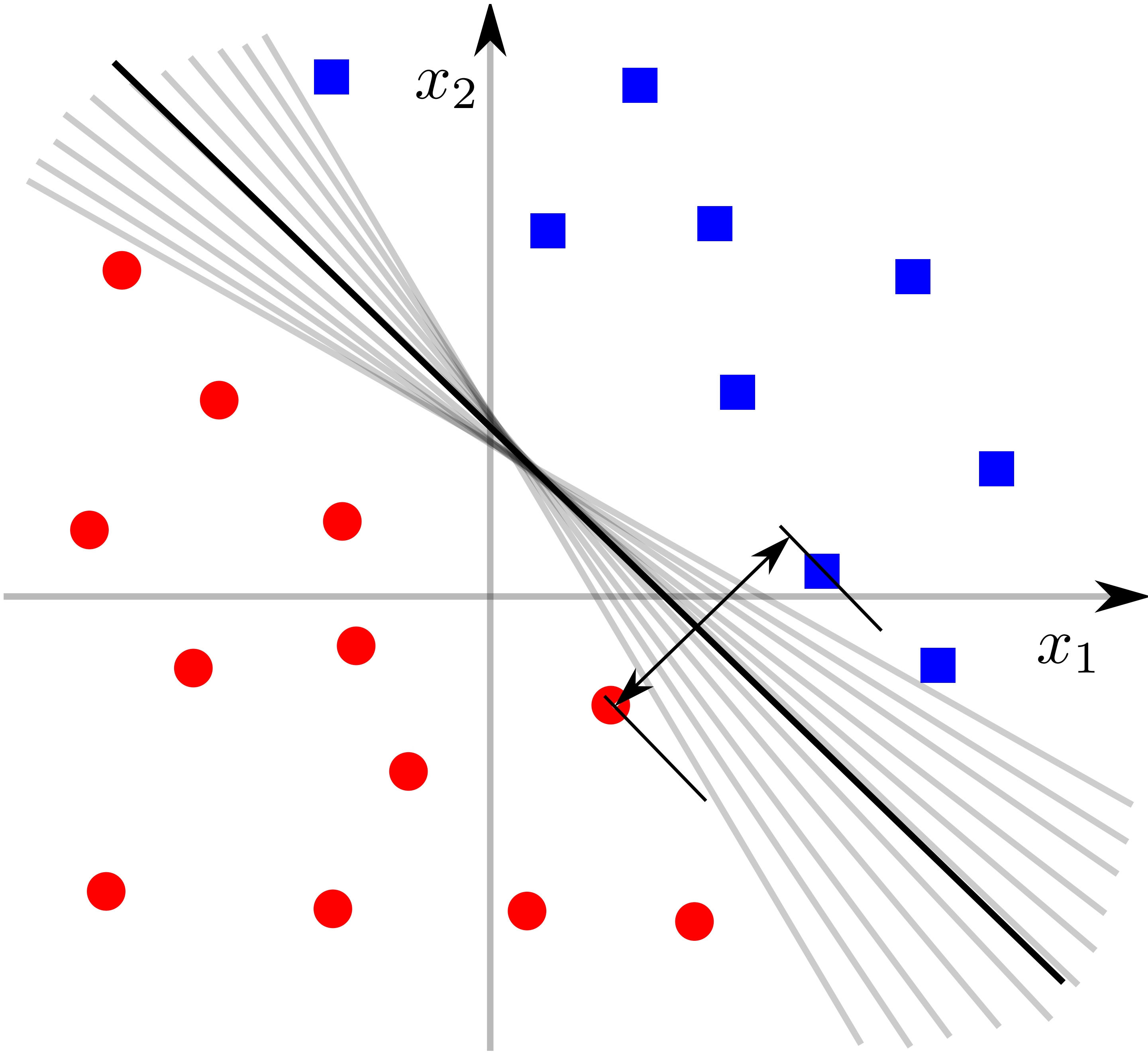
Support vector machines

In a nutshell

- Linear classifiers selecting hyperplane maximizing separation margin between classes (*large margin classifiers*)
- Solution only depends on a small subset of training examples (*support vectors*)
- Sound generalization theory (bounds or error based on margin)
- Can be easily extended to nonlinear separation (*kernel machines*)

Maximum margin classifier



Maximum margin classifier

Classifier margin

- Given a training set $\mathcal{D}$, a classifier *confidence margin* is:

$$\rho = \min_{(\boldsymbol{x},y)\in\mathcal{D}} y f(\boldsymbol{x})$$

- It is the minimal confidence margin (for predicting the true label) among training examples

- A classifier *geometric margin* is:

$$\frac{\rho}{||\boldsymbol{w}||} = \min_{(\boldsymbol{x},y)\in\mathcal{D}} \frac{yf(\boldsymbol{x})}{||\boldsymbol{w}||}$$

Maximum margin classifier

Canonical hyperplane

- There is an infinite number of equivalent formulation for the same hyperplane:

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{x} + w_0 &= 0 \\ \alpha(\boldsymbol{w}^T\boldsymbol{x} + w_0) &= 0 \quad \forall\alpha \neq 0 \end{aligned}$$

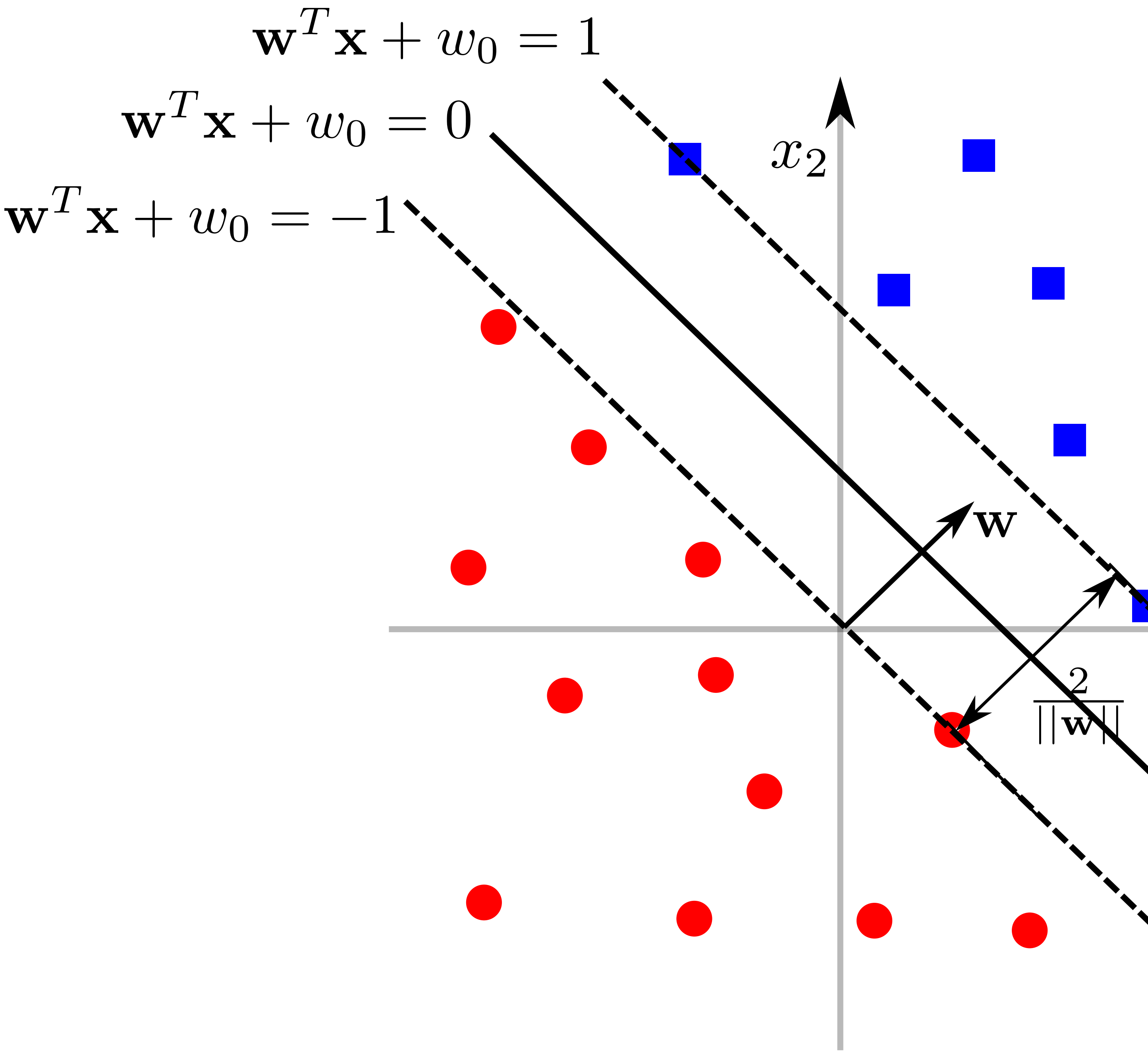
- The *canonical hyperplane* is the hyperplane having confidence margin equal to 1:

$$\rho = \min_{(\boldsymbol{x},y)\in\mathcal{D}} yf(\boldsymbol{x}) = 1$$

- Its geometric margin is:

$$\frac{\rho}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||}$$

Maximum margin classifier



Hard margin SVM

Theorem 1 (Margin Error Bound)**.** *Consider the set of decision functions* $f(\mathbf{x}) = \text{sign}\mathbf{w}^T\mathbf{x}$ *with* $||\mathbf{w}|| \leq \Lambda$ *and* $||\mathbf{x}|| \leq R$*, for some* $R, \Lambda > 0$*. Moreover, let* $\rho > 0$ *and* ν *denote the fraction of training examples with margin smaller than* $\rho/||\mathbf{w}||$*, referred to as the* margin error.

For all distributions P *generating the data, with probability at least* $1 - \delta$ *over the drawing of the* m *training patterns, and for any* $\rho > 0$ *and* $\delta \in (0, 1)$*, the probability that a test pattern drawn from P will be misclassified is* *bound from above by*

$$\nu + \sqrt{\frac{c}{m}\left(\frac{R^2\Lambda^2}{\rho^2}\ln^2 m + \ln(1/\delta)\right)}.$$

Here, c is a universal constant.

Hard margin SVM

Margin Error Bound: interpretation

$$\nu + \sqrt{\frac{c}{m}\left(\frac{R^2\Lambda^2}{\rho^2}\ln^2 m + \ln(1/\delta)\right)}.$$

The probability of test error depends on (among other components):

- number of margin errors ν (examples with margin smaller than $\rho/||\boldsymbol{w}||$)
- number of training examples (error depends on $\sqrt{\frac{\ln^2 m}{m}}$)
- size of the margin (error depends on $1/\rho^2$)

Note
If ρ is fixed to 1 (canonical hyperplane), maximizing margin corresponds to minimizing $||\boldsymbol{w}||$

Hard margin SVM

Learning problem

$$\begin{aligned} \min_{\boldsymbol{w},w_0} \quad & \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{subject to:} \quad & \\ & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 \\ & \forall(\boldsymbol{x}_i, y_i) \in \mathcal{D} \end{aligned}$$

Note

- constraints guarantee that all points are correctly classified (plus canonical form)
- minimization corresponds to maximizing the (squared) margin
- quadratic optimization problem (objective is quadratic, points satisfying constraints form a convex set)

Hard margin SVM

Learning problem

$$\begin{aligned} \min_{\boldsymbol{w},w_0} \quad & \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{subject to:} \quad & \\ & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 \\ & \forall(\boldsymbol{x}_i, y_i) \in \mathcal{D} \end{aligned}$$

Note

- constraints guarantee that all points are correctly classified (plus canonical form)
- minimization corresponds to maximizing the (squared) margin
- quadratic optimization problem (objective is quadratic, points satisfying constraints form a convex set)

Digression: constrained optimization

Karush-Kuhn-Tucker (KKT) approach

- A constrained optimization problem can be addressed by converting it into an *unconstrained* problem with the same solution
- Let's have a constrained optimization problem as:

$$\begin{aligned} \min_z \quad & f(z) \\ \text{subject to:} \quad & \\ & g_i(z) \geq 0 \; \forall i \end{aligned}$$

- Let's introduce a non-negative variable $\alpha_i \geq 0$ (called Lagrange multiplier) for each constraint and rewrite the optimization problem as (Lagrangian):

$$\min_z \max_{\boldsymbol{\alpha} \geq 0} f(z) - \sum_i \alpha_i g_i(z)$$

Digression: constrained optimization

Karush-Kuhn-Tucker (KKT) approach

$$\min_z \max_{\boldsymbol{\alpha} \geq 0} f(z) - \sum_i \alpha_i g_i(z)$$

The optimal solutions z^* for this problem are the same as the optimal solutions for the original (constrained) problem:

- If for a given z' at least one constraint is *not* satisfied, i.e. $g_i(z') < 0$ for some i, maximizing over α_i leads to an infinite value (not a minimum, unless there is no non-infinite minimum)
- If all constraints are satisfied (i.e. $g_i(z') \geq 0$ for all i), maximization over the $\boldsymbol{\alpha}$ will set all elements of the summation to zero, so that z' is a solution of $\min_z f(z)$.

Hard margin SVM
Karush-Kuhn-Tucker (KKT) approach

$$\begin{aligned} \min_{\boldsymbol{w}, w_0} \quad & \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{subject to:} \quad & \\ & y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1 \\ & \forall (\boldsymbol{x}_i, y_i) \in \mathcal{D} \end{aligned}$$

- The constraints can be included in the minimization using Lagrange multipliers $\alpha_i \geq 0$ ($m = |\mathcal{D}|$):

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + \ w_0 \ ) - 1)$$

- The Lagrangian is minimized wrt $\mathbf{w}, w_0$ and maximized wrt α_i (solution is a *saddle point*)

Hard margin SVM

Dual formulation

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + \ w_0 \ ) - 1)$$

- Vanishing derivatives wrt primal variables we get:

$$\frac{\partial}{\partial w_0} L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$
$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

Hard margin SVM

Dual formulation

- Substituting in the Lagrangian we get:

$$\begin{aligned}
&\frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + \ w_0 \ ) - 1) = \\
&\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \\
&\underbrace{\sum_{i=1}^{m} \alpha_i y_i}_{=0} w_0 + \sum_{i=1}^{m} \alpha_i = \\
&\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j = L(\boldsymbol{\alpha})
\end{aligned}$$

- which is to be maximized wrt the dual variables $\boldsymbol{\alpha}$

Hard margin SVM

Dual formulation

$$\begin{aligned}
\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\
\text{subject to} \quad & \alpha_i \geq 0 \quad i = 1, \dots, m \\
& \sum_{i=1}^{m} \alpha_i y_i = 0
\end{aligned}$$

- The resulting maximization problem including the constraints
- Still a quadratic optimization problem

Hard margin SVM

Note

- The dual formulation has simpler contraints (box), easier to solve
- The primal formulation has $d+1$ variables (number of features +1):

$$\min_{\boldsymbol{w},w_0}\frac{1}{2}||\boldsymbol{w}||^2$$

- The dual formulation has m variables (number of training examples):

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$$

- One can choose the primal formulation if it has much less variables (problem dependent)

Hard margin SVM

Decision function

- Substituting $\mathbf{w}=\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i$ in the decision function we get:

$$f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+w_0=\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i^T\mathbf{x}+w_0$$

- The decision function is linear combination of dot products between training points and the test point
- dot product is kind of *similarity* between points
- Weights of the combination are $\alpha_i y_i$: large α_i implies large contribution towards class y_i (times the similarity)

Hard margin SVM

Karush-Khun-Tucker conditions (KKT)

$$L(\mathbf{w},w_0,\boldsymbol{\alpha})=\frac{1}{2}||\mathbf{w}||^2-\sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i+\ w_0\ )-1)$$

- At the saddle point it holds that for all i:

$$\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i+\ w_0\ )-1)=0$$

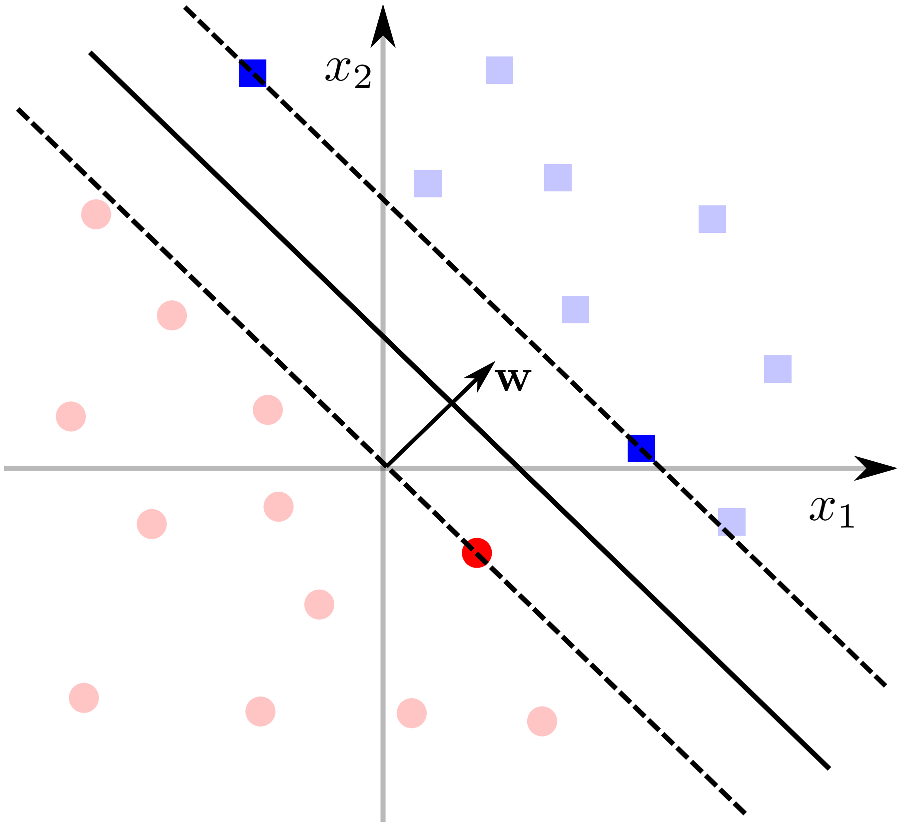
- Thus, either the example does not contribute to the final $f(\boldsymbol{x})$:

$$\alpha_i=0$$

- or the example stays on the minimal confidence hyperplane from the decision one:

$$y_i(\mathbf{w}^T\mathbf{x}_i+\ w_0\ )=1$$

Hard margin SVM



Support vectors

- points staying on the minimal confidence hyperplanes are called *support vectors*
- All other points do not contribute to the final decision function (i.e. they could be removed from the training set)
- SVM are *sparse* i.e. they typically have few support vectors

Hard margin SVM

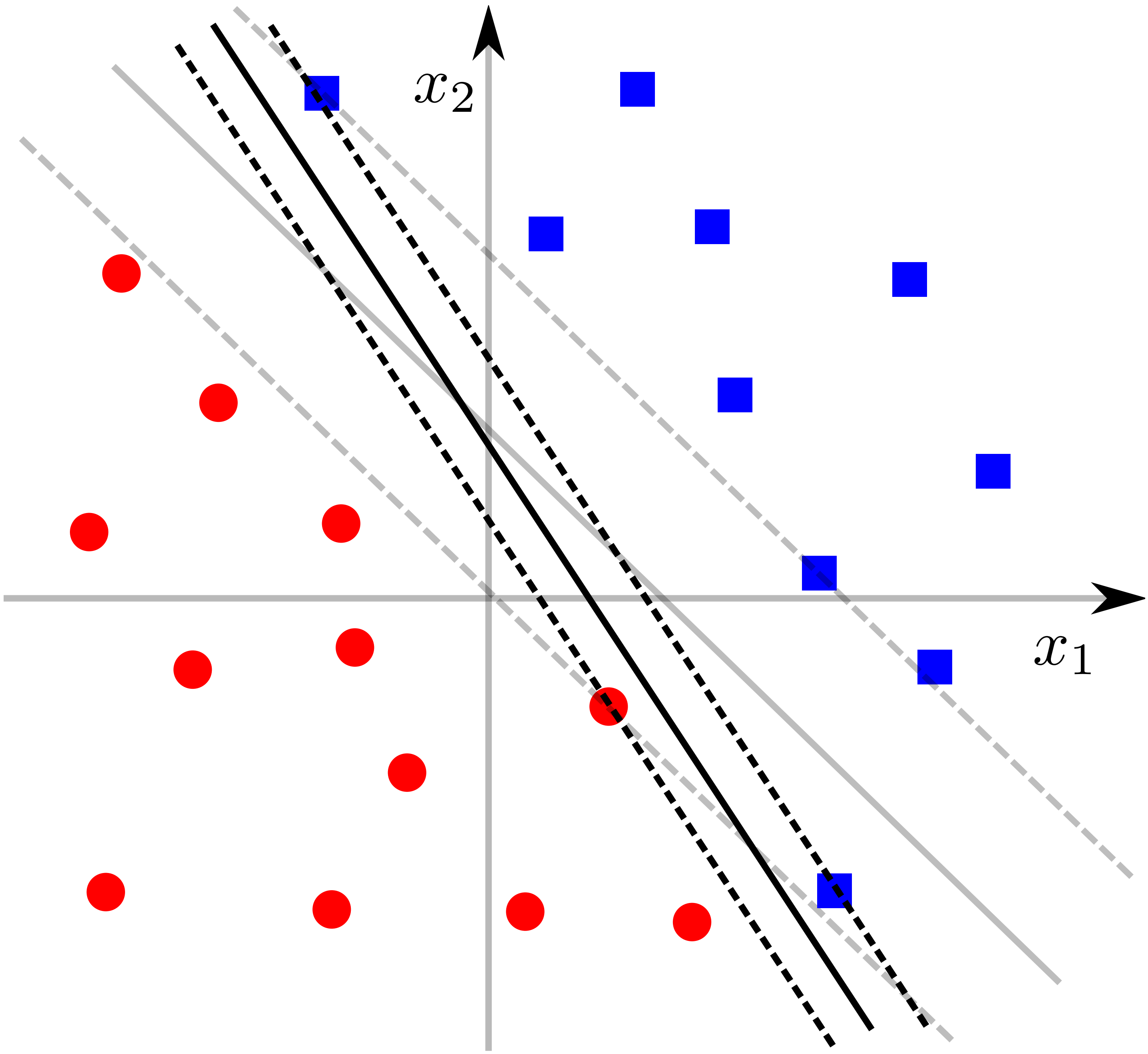
Decision function bias

- The bias w_0 can be computed from the KKT conditions
- Given an arbitrary support vector $\mathbf{x}_i$ (with $\alpha_i > 0$) the KKT conditions imply:

$$y_i(\mathbf{w}^T\mathbf{x}_i + \ w_0\ ) = 1$$
$$y_i\mathbf{w}^T\mathbf{x}_i + y_i w_0 = 1$$
$$w_0 = \frac{1 - y_i\mathbf{w}^T\mathbf{x}_i}{y_i}$$

- For robustness, the bias is usually averaged over all support vectors

Soft margin SVM



Soft margin SVM

Slack variables

$$\min_{\mathbf{w}\in\mathcal{X}, w_0\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^m} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^m \xi_i$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \xi_i \quad i = 1, \ldots, m$$
$$\xi_i \geq 0 \quad i = 1, \ldots, m$$

- A slack variable ξ_i represents the penalty for example x_i not satisfying the margin constraint
- The sum of the slacks is minimized together to the inverse margin
- The regularization parameter $C \geq 0$ trades-off data fitting and size of the margin

Soft margin SVM

Regularization theory

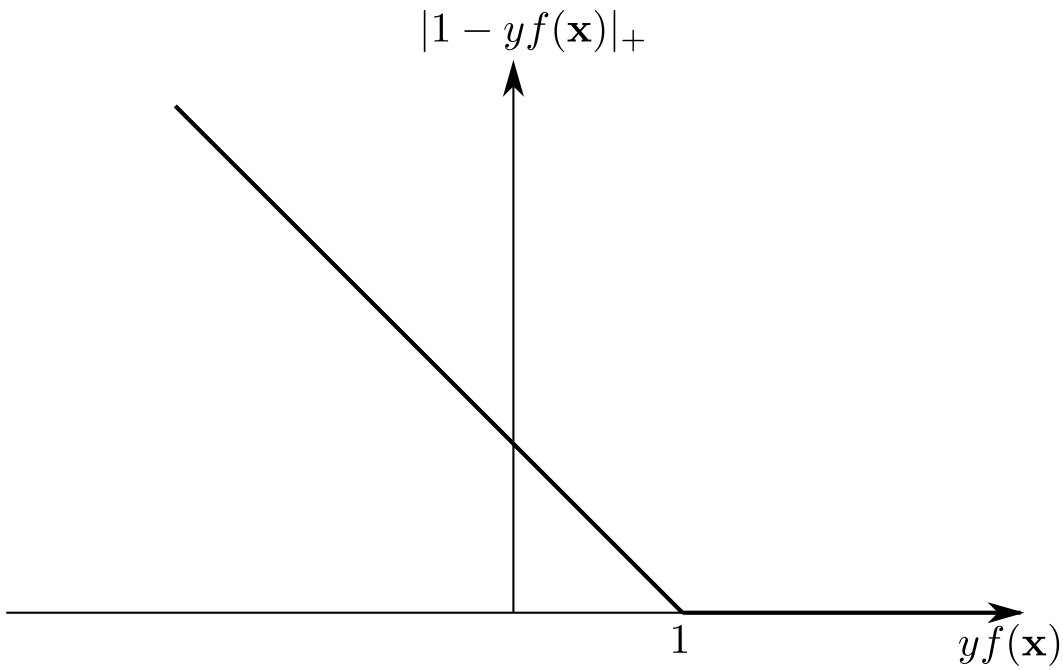
$$\min_{\mathbf{w}\in\mathcal{X}, w_0\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^m} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^m \ell(y_i, f(\mathbf{x}_i))$$

- Regularized loss minimization problem
- The loss term accounts for error minimization
- The margin maximization term accounts for regularization i.e. solutions with larger margin are preferred

Note

- Regularization is a standard approach to prevent overfitting
- It corresponds to a prior for *simpler* (more regular, smoother) solutions

Soft margin SVM



Hinge loss

$$\ell(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+ = |1 - y_i(\mathbf{w}^T \mathbf{x}_i + w_0)|_+$$

- $|z|_+ = z$ if $z > 0$ and 0 otherwise (positive part)
- it corresponds to the slack variable ξ_i (violation of margin costraint)
- all examples not violating margin costraint have zero loss (sparse set of support vectors)

Soft margin SVM
Lagrangian

$$L = C \sum_{i=1}^{m} \xi_i + \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{m} \alpha_i (y_i(\boldsymbol{w}^T \mathbf{x}_i + \ w_0 \ ) - 1 + \xi_i) - \sum_{i=1}^{m} \beta_i \xi_i$$

- where $\alpha_i \geq 0$ and $\beta_i \geq 0$
- Vanishing derivatives wrt primal variables we get:

$$\frac{\partial}{\partial w_0} L = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\frac{\partial}{\partial \mathbf{w}} L = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial \xi_i} L = 0 \quad \Rightarrow \quad C - \alpha_i - \beta_i = 0$$

Soft margin SVM
Dual formulation

- Substituting in the Lagrangian we get

$$
\begin{aligned}
& C\sum_{i=1}^m \xi_i + \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^m \alpha_i(y_i(\boldsymbol{w}^T\mathbf{x}_i + \ w_0\ ) - 1 + \xi_i) - \sum_{i=1}^m \beta_i\xi_i = \\
& \sum_{i=1}^m \xi_i \underbrace{(C - \alpha_i - \beta_i)}_{=0} + \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \\
& \underbrace{\sum_{i=1}^m \alpha_i y_i}_{=0} w_0 + \sum_{i=1}^m \alpha_i = \\
& \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j = L(\boldsymbol{\alpha})
\end{aligned}
$$

Soft margin SVM

Dual formulation

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\
\text{subject to} \quad & 0 \leq \alpha_i \leq C \quad i = 1,\dots,m \\
& \sum_{i=1}^m \alpha_i y_i = 0
\end{aligned}
$$

- The box constraint for α_i comes from $C - \alpha_i - \beta_i = 0$ (and the fact that both $\alpha_i \geq 0$ and $\beta_i \geq 0$)

Soft margin SVM

Karush-Khun-Tucker conditions (KKT)

$$
L = C\sum_{i=1}^m \xi_i + \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^m \alpha_i(y_i(\boldsymbol{w}^T\mathbf{x}_i + \ w_0\ ) - 1 + \xi_i) - \sum_{i=1}^m \beta_i\xi_i
$$

- At the saddle point it holds that for all i:

$$
\begin{aligned}
& \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + \ w_0\ ) - 1 + \xi_i) = 0 \\
& \beta_i\xi_i = 0
\end{aligned}
$$

- Thus, support vectors ($\alpha_i > 0$) are examples for which $(y_i(\mathbf{w}^T\mathbf{x}_i + \ w_0\ ) \leq 1$

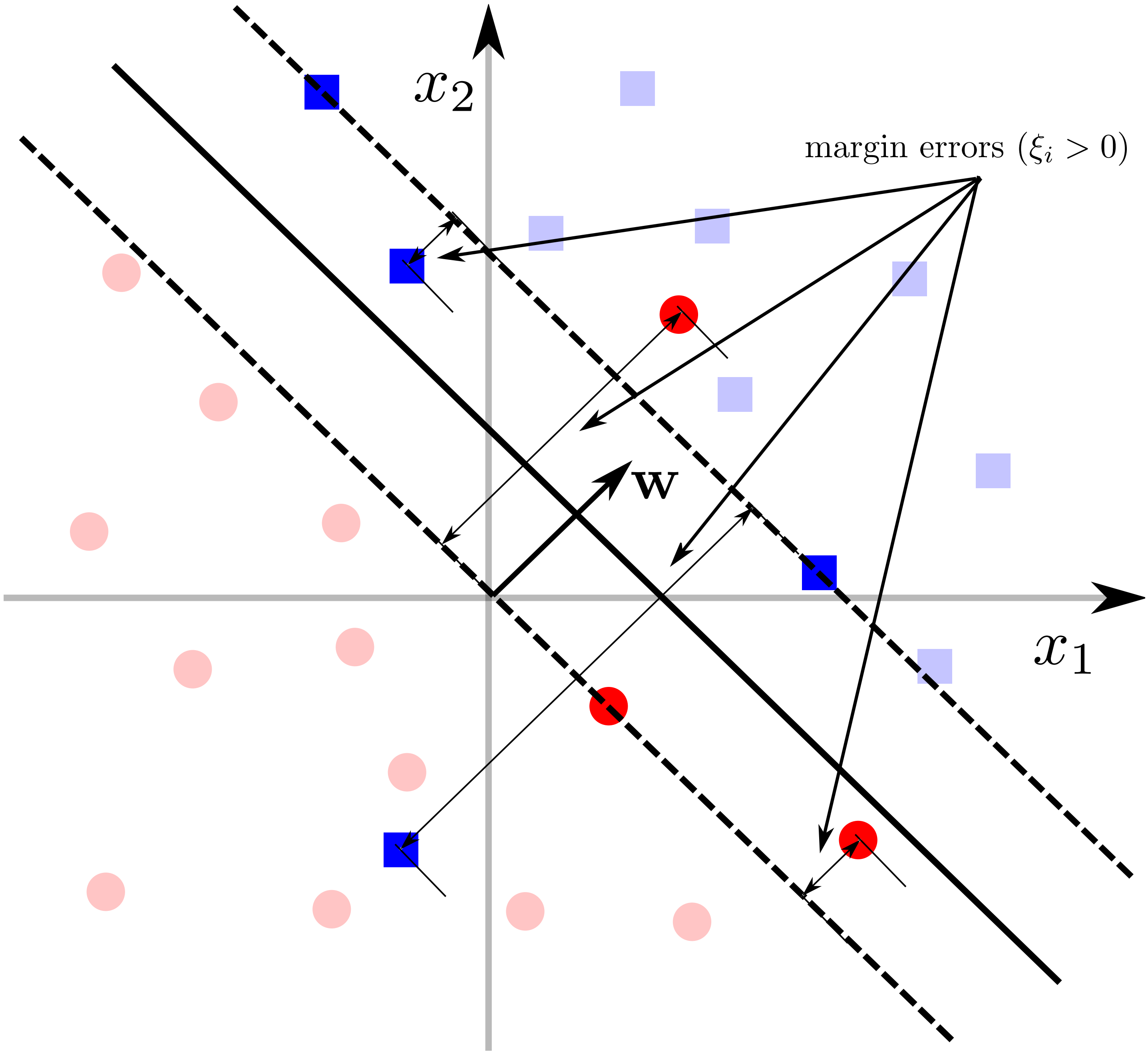
Soft margin SVM

Support Vectors

$$\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + \ w_0 \ ) - 1 + \xi_i) = 0$$
$$\beta_i\xi_i = 0$$

- If $\alpha_i < C$, $C - \alpha_i - \beta_i = 0$ and $\beta_i\xi_i = 0$ imply that $\xi_i = 0$
  - These are called *unbound* SV (($y_i(\mathbf{w}^T\mathbf{x}_i + \ w_0 \ ) = 1$, they stay on the confidence one hyperplane
- If $\alpha_i = C$ (*bound* SV) then ξ_i can be greater the zero, in which case the SV are margin errors

Support vectors



Large-scale SVM learning

Stochastic gradient descent

$$\min_{\mathbf{w} \in \mathcal{X}} \frac{\lambda}{2} ||\mathbf{w}||^2 + \frac{1}{m} \sum_{i=1}^{m} |1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle|_+$$

- Objective for a single example $(\mathbf{x}_i, y_i)$:

$$E(\mathbf{w}; (\mathbf{x}_i, y_i)) = \frac{\lambda}{2} ||\mathbf{w}||^2 + |1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle|_+$$

- Subgradient:

$$\nabla_{\mathbf{w}} E(\mathbf{w}; (\mathbf{x}_i, y_i)) = \lambda \mathbf{w} - \mathbb{1}[y_i \langle \mathbf{w}, \mathbf{x}_i \rangle < 1] y_i \mathbf{x}_i$$

Large-scale SVM learning

Note

- Indicator function

$$\mathbb{1}[y_i \langle \mathbf{w}, \mathbf{x}_i \rangle < 1] = \begin{cases} 1 & \text{if } y_i \langle \mathbf{w}, \mathbf{x}_i \rangle < 1 \\ 0 & \text{otherwise} \end{cases}$$

- The subgradient of a function f at a point $\mathbf{x}_0$ is any vector $\boldsymbol{v}$ such that for any $\mathbf{x}$:

$$f(\mathbf{x}) - f(\mathbf{x}_0) \geq \boldsymbol{v}^T (\mathbf{x} - \mathbf{x}_0)$$

Large-scale SVM learning

Pseudocode (pegasus)

1. Initialize $\mathbf{w}_1 = 0$
2. for $t = 1$ to T:
   (a) Randomly choose $(\mathbf{x}_{i_t}, y_{i_t})$ from $\mathcal{D}$
   (b) Set $\eta_t = \frac{1}{\lambda t}$
   (c) Update $\mathbf{w}$:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} E(\mathbf{w}; (\mathbf{x}_{i_t}, y_{i_t}))$$

3. Return $\mathbf{w}_{T+1}$

Note

The choice of the learning rate allows to bound the runtime for an ϵ-accurate solution to $\mathcal{O}(d/\lambda\epsilon)$ with d maximum number of non-zero features in an example.

References

Biblio

- C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Mining and Knowledge Discovery, 2(2), 121-167, 1998.
- S. Shalev-Shwartz et al., *Pegasos: primal estimated sub-gradient solver for SVM*, Mathematical Programming, 127(1), 3-30, 2011.

Software

- svm module in `scikit-learn` http://scikit-learn.org/stable/index.html
- `libsvm` http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- `svmlight` http://svmlight.joachims.org/

APPENDIX

Appendix
Additional reference material

Large-scale SVM learning

Dual version

- It is easy to show that:

$$\mathbf{w}_{t+1} = \frac{1}{\lambda t} \sum_{i=1}^{t} \mathbb{1}[y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle < 1] y_{i_t} \mathbf{x}_{i_t}$$

- We can represent $\mathbf{w}_{t+1}$ implicitly by storing in vector $\boldsymbol{\alpha}_{t+1}$ the number of times each example was selected and had a non-zero loss, i.e.:

$$\alpha_{t+1}[j] = |\{t' \leq t : i_{t'} = j \wedge y_j \langle \mathbf{w}_{t'}, \mathbf{x}_j \rangle < 1\}|$$

Large-scale SVM learning

Pseudocode (pegasus dual)

1. Initialize $\boldsymbol{\alpha}_1 = 0$
2. for $t = 1$ to T:
   (a) Randomly choose $(\mathbf{x}_{i_t}, y_{i_t})$ from $\mathcal{D}$
   (b) Set $\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t$
   (c) If $y_{i_t} \frac{1}{\lambda t} \sum_{j=1}^{t} \alpha_t[j] y_j \langle \mathbf{x}_j, \mathbf{x}_{i_t} \rangle < 1$
      i. $\alpha_{t+1}[i_t] = \alpha_{t+1}[i_t] + 1$
3. Return $\boldsymbol{\alpha}_{T+1}$

Note
This will be useful when combined with kernels.