examples, and train on the remaining ones, then the probability of error on the left out example gives us a fair indication of the true test error. Of course, doing this for a single training example leads to an error of either zero or one, so it does not yet give an estimate of the test error. The leave-one-out method *repeats* this procedure for each individual training example in turn, and averages the resulting errors.

Let us return to the present case. If we leave out a pattern $\mathbf{x}_{i^*}$, and construct the solution from the remaining patterns, the following outcomes are possible (cf. (7.11)):

1. $y_{i^*}\left(\langle \mathbf{x}_{i^*}, \mathbf{w}\rangle + b\right) > 1$. In this case, the pattern is classified correctly and does not lie on the margin. These are patterns that would not have become SVs anyway.

2. $y_{i^*}\left(\langle \mathbf{x}_{i^*}, \mathbf{w}\rangle + b\right) = 1$. In other words, $\mathbf{x}_{i^*}$ exactly meets the constraint (7.11). In this case, the solution $\mathbf{w}$ does not change, even though the coefficients α_i would change: Namely, if $\mathbf{x}_{i^*}$ might have become a Support Vector (i.e., $\alpha_{i^*} > 0$) had it been kept in the training set. In that case, the fact that the solution is the same, no matter whether $\mathbf{x}_{i^*}$ is in the training set or not, means that $\mathbf{x}_{i^*}$ can be written as $\sum_{\mathrm{SVs}} \beta_i y_i \mathbf{x}_i$ with, $\beta_i \geq 0$. Note that condition 2 is *not* equivalent to saying that $\mathbf{x}_{i^*}$ may be written as some linear combination of the remaining Support Vectors: Since the sign of the coefficients in the linear combination is determined by the class of the respective pattern, not any linear combination will do. Strictly speaking, $\mathbf{x}_{i^*}$ must lie in the cone spanned by the $y_i\mathbf{x}_i$, where the $\mathbf{x}_i$ are all Support Vectors.[6] For more detail, see [565] and Section 12.2.

3. $0 < y_{i^*}\left(\langle \mathbf{x}_{i^*}, \mathbf{w}\rangle + b\right) < 1$. In this case, $\mathbf{x}_{i^*}$ lies within the margin, but still on the correct side of the decision boundary. Thus, the solution looks different from the one obtained with $\mathbf{x}_{i^*}$ in the training set (in that case, $\mathbf{x}_{i^*}$ would satisfy (7.11) after training); classification is nevertheless correct.

4. $y_{i^*}\left(\langle \mathbf{x}_{i^*}, \mathbf{w}\rangle + b\right) < 0$. This means that $\mathbf{x}_{i^*}$ is classified incorrectly.

Note that cases 3 and 4 necessarily correspond to examples which would have become SVs if kept in the training set; case 2 potentially includes such situations. Only case 4, however, leads to an error in the leave-one-out procedure. Consequently, we have the following result on the generalization error of optimal margin classifiers [570]:[7]

Leave-One-Out Bound

Proposition 7.4 *The expectation of the number of Support Vectors obtained during training on a training set of size m, divided by m, is an* upper bound *on the expected probability of test error of the SVM trained on training sets of size $m-1$.*[8]

6. Possible non-uniqueness of the solution's expansion in terms of SVs is related to zero Eigenvalues of $(y_i y_j k(x_i, x_j))_{ij}$, cf. Proposition 2.16. Note, however, the above caveat on the distinction between linear combinations, and linear combinations with coefficients of fixed sign.
7. It also holds for the generalized versions of optimal margin classifiers described in the following sections.
8. Note that the leave-one-out procedure performed with m training examples thus yields