computation. For this purpose we have to express $f$ as a kernel expansion

$$f(x) = \sum_i \alpha_i k(x_i, x) \tag{10.126}$$

where the $x_i$ are (previously seen) training patterns. Then (10.126) becomes

$$\alpha_t \longleftarrow (1 - \lambda\Lambda)\alpha_t - \Lambda c'(x_t, y_t, f(x_t)) \qquad \text{for } i = t \tag{10.127}$$

$$= -\Lambda c'(x_t, y_t, f(x_t)) \qquad \text{for } \alpha_t = 0 \tag{10.128}$$

$$\alpha_i \longleftarrow (1 - \lambda\Lambda)\alpha_i \qquad \text{for } i \neq t. \tag{10.129}$$

Eq. (10.127) means that, at each iteration, the kernel expansion may grow by one term. Further, the cost of training at each step is not larger than the prediction cost. Once we have computed $f(x_t)$, $\alpha_t$ is obtained by the value of the derivative of $c$ at $(x_t, y_t, f(x_t))$.

Instead of updating all coefficients $\alpha_i$ we may simply cache the power series $1, (1 - \lambda\Lambda), (1 - \lambda\Lambda)^2, (1 - \lambda\Lambda)^3, \ldots$ and pick suitable terms as needed. This is particularly useful if the derivatives of the loss function $c$ only assume discrete values, say $\{-1, 0, 1\}$ as is the case when using the soft-margin type loss functions.

**Truncation** The problem with (10.127) and (10.129) is that, without any further measures, the number of basis functions $n$ will grow without bound. This is not desirable since $n$ determines the amount of computation needed for prediction. The regularization term helps us here. At each iteration the coefficients $\alpha_i$ with $i \neq t$ are shrunk by $(1 - \lambda\Lambda)$. Thus, after $\tau$ iterations, the coefficient $\alpha_i$ will be reduced to $(1 - \lambda\Lambda)^\tau \alpha_i$.

**Proposition 10.8 (Truncation Error)** *For a loss function $c(x, y, f(x))$ with its first derivative bounded by $C$ and a kernel $k$ with bounded norm $\|k(x, \cdot)\| \leq X$, the truncation error in $f$ incurred by dropping terms $\alpha_i$ from the kernel expansion of $f$ after $\tau$ update steps is bounded by $\Lambda(1 - \lambda\Lambda)^\tau CX$. In addition, the total truncation error due to dropping all terms which are at least $\tau$ steps old is bounded by*

$$\|f - f_{\text{trunc}}\|_{\mathcal{H}} \leq \sum_{i=1}^{t-\tau} \Lambda(1 - \lambda\Lambda)^{t-i} CX < \lambda^{-1}(1 - \lambda\Lambda)^\tau CX \tag{10.130}$$

Here $f_{\text{trunc}} = \sum_{i=t-\tau+1}^{t} \alpha_i k(x_i, \cdot)$. Obviously the approximation quality increases exponentially with the number of terms retained.

The regularization parameter $\lambda$ can thus be used to control the storage requirements for the expansion. Moreover, it naturally allows for distributions $\mathrm{P}(x, y)$ that change over time in which case it is desirable to *forget* instances $(x_i, y_i)$ that are much older than the average time scale of the distribution change [298].

We now proceed to applications of (10.127) and (10.129) in specific learning situations. We utilize the standard addition of the constant offset $b$ to the function expansion, $g(x) = f(x) + b$ where $f \in \mathcal{H}$ and $b \in \mathbb{R}$. Hence we also update $b$ into $b - \Lambda \partial_b R_{\text{stoch}}[g]$.

**Classification** We begin with the soft margin loss (3.3), given by $c(x, y, g(x)) =$