Furthermore, we define the probability of an example $k$ respectively. For a fixed $L = \{1, \ldots, L\}$ so that each latent variable is associated with a different label set.

Under the boosting framework, we define $H^o_k(x, l, l_s)$ and $H^a_k(x, l, l_s)$ as linear combinations of weak learners:

$$H^o_k(x, l, l_s) = \sum_{m=1}^{M} \rho^o_{k,m} h^o_{k,m}(x, l, l_s)$$

$$H^a_k(x, l, l_s) = \sum_{m=1}^{M} \rho^a_{k,m} h^a_{k,m}(x, l, l_s)$$

Furthermore, we define the probability of an example $x$ being class $k$ as:

$$\hat{p}_k(x) = \frac{\exp(F_k(x, L))}{\sum_l \sum_{l_s} \exp(F_l(x, L))}$$

and we can define the loss function for LatentBoost as the negative log-likelihood of the training data:

$$\text{loss}_{LB} = \sum_{n=1}^{N} \Psi(y^{(n)} - F_k(x^{(n)}, L))) = - \sum_{n=1}^{N} \sum_{k=1}^{K} y^{(n)} \log \hat{p}_k(x^{(n)})$$

To optimize the loss function, we learn the weak learners (both unary and pairwise) and their associated weights in an iterative fashion. At the $m$-th iteration, we compute the gradient of the loss function $\text{loss}_{LB}$ with respect to the current unary potential functions, $\rho^o_{k,m}(x^{(n)}, l)$. We then pick the weak learner $h^o_{k,m}$ that is the most parallel in the $N$-dimensional data space with the negative gradient $[-\rho^o_{k,m}(x^{(n)}, l)]_t^N$ by a least-squares minimization problem. Updating the pairwise functions is done in a similar fashion. The weights $\rho^a_{k,m}$ and $\rho^o_{k,m}$ can be simply computed by a line search algorithm. Putting everything together, we have the LatentBoost algorithm illustrated in Algorithm 1.

---

**Algorithm 1 LatentBoost**

1. $F_{x,0} = 0, k = 1, \ldots, K$
2. for $m = 1$ to $M$
3. Compute $Pr(l|x^{(n)})$, $Pr(l^{(n)} = 1, l|x^{(n)})$, $Pr(l, l^{(n)} = 1|x^{(n)})$ and $Pr(l^{(n)} = 1, l, l^{(n)} = 1|x^{(n)})$ for $\forall t, t' \in \mathcal{Y}$, $\forall (t, s) \in \mathcal{E}$
4. for $k = 1$ to $K$
5. //update unary potentials
6. for $i \in \mathcal{Y}$
7. $h_{k,m} = \arg\min \sum_{n=1}^{N} \Psi_{n}^{-1} \left[ -\rho^o_{k,m}(x^{(n)}, l) \right]_t^N$
8. $\rho^o_{k,m} = \arg\min \sum_{n=1}^{N} \Psi_{n}^{-1} \left[ y^{(n)} - F_{k,m-1}(x^{(n)}, l) + \rho^o_{k,m}(x^{(n)}, l) \right]_t^N$
9. $F_{k,m}(x, l) = F_{k,m-1}(x, l) + \rho^o_{k,m} h_{k,m}^o(x, l)$

10. end for
11. //update pairwise potentials
12. for $(t, s) \in \mathcal{E}$
13. $h_{k,m} = \arg\min \sum_{n=1}^{N} \sum_{l_s} \left[ -\rho^a_{k,m}(x^{(n)}, l, l_s) \right]_t^N$
14. $\rho^a_{k,m} = \arg\min \sum_{n=1}^{N} \sum_{l_s} \left[ y^{(n)} - F_{k,m-1}(x^{(n)}, l) + \rho^a_{k,m} h_{k,m}^a(x^{(n)}, l, l_s) \right]_t^N$
15. $F_{k,m}(x, l, l_s) = F_{k,m-1}(x, l, l_s) + \rho^a_{k,m} h_{k,m}^a(x, l, l_s)$

16. end for
17. end for
18. end for

---

To test LatentBoost on the task of human action recognition. Our method for human action recognition operates on a figure-centric representation of the human figure extracted from an input video. We use the optical flow features in [2] on unary potentials and colour histogram features on pairwise potentials. We show that LatentBoost outperforms GradientBoost on two publicly available datasets: Weizmann human action dataset [1] and TRECVID surveillance event detection [5].

---


