

---

# Generalized Sparselets Computational Cost Analysis (Supplementary Material)

---

Ross Girshick  
Hyun Oh Song  
Trevor Darrell

University of California, Berkeley, Berkeley, CA 94720

RBG@EECS.BERKELEY.EDU  
SONG@EECS.BERKELEY.EDU  
TREVOR@EECS.BERKELEY.EDU

Consider the linear discriminant function

$$f_{\mathbf{w}}(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \mathbf{w}^\top \Phi(x, y), \quad (1)$$

where  $\mathbf{w}$  is a parameter vector in  $\mathbb{R}^n$ ,  $x$  comes from an input space  $\mathcal{X}$ , and  $y$  is in a label space  $\mathcal{Y}$ .

For clarity, we will assume that  $n = pm$  for some integer  $p$ , where  $m$  is the length of each sparselet  $\mathbf{s}_i$  in the sparselet dictionary  $\mathbf{S}$ . This assumption can be removed with simple modifications to the discussion that follows. We partition  $\mathbf{w}$  into a set of blocks  $\mathbf{b}_i$  in  $\mathbb{R}^m$  such that  $\mathbf{w} = (\mathbf{b}_1^\top, \dots, \mathbf{b}_p^\top)^\top$ .

Let  $\mathcal{A}$  be an algorithm such that  $\mathcal{A}(\mathbf{w}, x)$  computes  $f_{\mathbf{w}}(x)$  — *i.e.*, it solves the argmax in Eq. 1. We are going to build a bipartite graph  $\mathcal{G} = (\mathcal{B} \cup \mathcal{C}, \mathcal{E})$  that represents certain computations performed by  $\mathcal{A}$ . The graph depends on  $\mathcal{A}$ 's inputs  $\mathbf{w}$  and  $x$ , but to lighten notation we will omit this dependence.

Each node in  $\mathcal{G}$  corresponds to a vector in  $\mathbb{R}^m$ . With a slight abuse of notation we will label each node with the vector that it is in correspondence with. Similarly, we will label the edges with a pair of vectors (*i.e.*, nodes), each in  $\mathbb{R}^m$ . We define the first set of disconnected nodes in  $\mathcal{G}$  to be the set of all blocks in  $\mathbf{w}$ :  $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_p\}$ . We will define the second set of disconnected nodes,  $\mathcal{C}$ , next.

Any algorithm that computes Eq. 1 will perform some number of computations of the form  $\mathbf{b}^\top \mathbf{c}$ , for a block  $\mathbf{b} \in \mathcal{B}$  and some vector  $\mathbf{c} \in \mathbb{R}^m$ . The vectors  $\mathbf{c}$  appearing in these computations are most likely subvectors of  $\Phi(x, y)$  arising from various values of  $y$ . The graph  $\mathcal{G}$  is going to represent all unique computations of this form. Conceptually, we can construct  $\mathcal{C}$  by running the algorithm  $\mathcal{A}$  and adding each *unique* vector  $\mathbf{c}$  that appears in a computation of the form  $\mathbf{b}^\top \mathbf{c}$  to  $\mathcal{C}$ . The edge set  $\mathcal{E}$  connects a node  $\mathbf{b} \in \mathcal{B}$  to a node in  $\mathbf{c} \in \mathcal{C}$  if and only if  $\mathcal{A}$  performs the computation  $\mathbf{b}^\top \mathbf{c}$ . For a specific algorithm  $\mathcal{A}$ , we can construct  $\mathcal{G}$  analytically. An example graph for a multiclass classification

problem is given in Fig. 1.

Graph  $\mathcal{G}$ 's edges encode exactly all of the computations of the form  $\mathbf{b}^\top \mathbf{c}$  and therefore we can use it to analyze the computational costs of  $\mathcal{A}$  with and without generalized sparselets.

Obviously, not all of the computation performed by  $\mathcal{A}$  are of the form captured by the graph. For example, when generalized distance transforms are used by  $\mathcal{A}$  to solve in the computation of Eq. 1 for deformable part models, the cost of computing the distance transforms is outside of the scope of  $\mathcal{G}$  (and outside the application of sparselets). We let the quantity  $T(\mathbf{w}, x)$  account for all computational costs not represented in  $\mathcal{G}$ .

We are now ready to write the number of operations performed by  $\mathcal{A}(\mathbf{w}, x)$ . First, without sparselets we have

$$T_{\text{Original}}(\mathbf{w}, x) = T(\mathbf{w}, x) + m \sum_{\mathbf{c} \in \mathcal{C}} \deg(\mathbf{c}), \quad (2)$$

where  $\deg(\mathbf{v})$  is the degree of a node  $\mathbf{v}$  in  $\mathcal{G}$ . The second term in Eq. 2 accounts for the  $m$  additions and multiplications that are performed when computing  $\mathbf{b}^\top \mathbf{c}$  for a pair of nodes  $(\mathbf{b}, \mathbf{c}) \in \mathcal{E}$ .

When sparselets are applied, the cost becomes

$$T_{\text{Sparselets}}(\mathbf{w}, x) = T(\mathbf{w}, x) + dm|\mathcal{C}| + \lambda_0 \sum_{\mathbf{c} \in \mathcal{C}} \deg(\mathbf{c}), \quad (3)$$

The second term in Eq. 3 accounts for the cost of pre-computing the sparselet responses,  $\mathbf{r} = \mathbf{S}^\top \mathbf{c}$  (cost  $dm$ ), for each node in  $\mathcal{C}$ . The third term accounts for the sparse dot product  $\alpha(\mathbf{b})^\top \mathbf{r}$  (cost  $\lambda_0$ ) computed for each pair  $(\mathbf{b}, \mathbf{c}) \in \mathcal{E}$ , where  $\alpha(\mathbf{b})$  is the sparselet activation vector for  $\mathbf{b}$ .

The speedup is the ratio  $T_{\text{Original}}/T_{\text{Sparselets}}$ .

$$\frac{T(\mathbf{w}, x) + m \sum_{i=1}^{|\mathcal{C}|} \deg(\mathbf{c}_i)}{T(\mathbf{w}, x) + dm|\mathcal{C}| + \lambda_0 \sum_{i=1}^{|\mathcal{C}|} \deg(\mathbf{c}_i)} \quad (4)$$

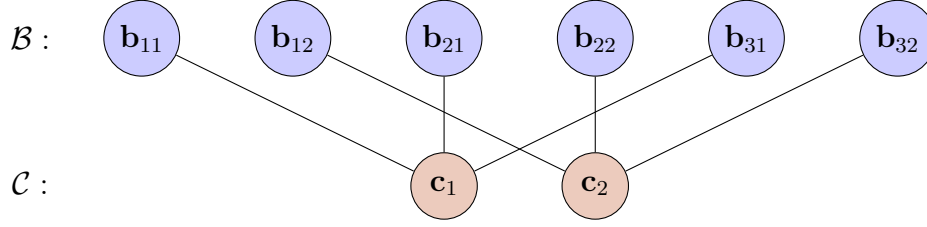


Figure 1. Computation graph for a multiclass problem with  $K = 3$ . Let the sparselet size be  $m$  and the number of blocks be  $p = 2$ . We define  $\mathbf{w} = (\mathbf{w}_1^\top, \mathbf{w}_2^\top, \mathbf{w}_3^\top)^\top$  in  $\mathbb{R}^{Kpm}$ . Each per-class classifier  $\mathbf{w}_k$  in  $\mathbb{R}^{pm}$  is partitioned into  $p$  blocks such that  $\mathbf{w}_k = (\mathbf{b}_{k1}^\top, \mathbf{b}_{k2}^\top)^\top$ . An input vector  $\mathbf{x}$  in  $\mathbb{R}^{pm}$  is partitioned into subvectors such that  $\mathbf{x} = (\mathbf{c}_1^\top, \mathbf{c}_2^\top)^\top$ . The feature map  $\Phi(\mathbf{x}, k)$  in  $\mathbb{R}^{Kpm}$  is defined as:  $\Phi(\mathbf{x}, 1) = (\mathbf{x}^\top, 0, \dots, 0)^\top$ ;  $\Phi(\mathbf{x}, 2) = (0, \dots, 0, \mathbf{x}^\top, 0, \dots, 0)^\top$ ;  $\Phi(\mathbf{x}, 3) = (0, \dots, 0, \mathbf{x}^\top)^\top$ . The edges in the graph encode the dot products computed while solving  $\operatorname{argmax}_{k \in \{1,2,3\}} \mathbf{w}^\top \Phi(\mathbf{x}, k)$ .

In all of the examples we consider in this paper, the degree of each node in  $\mathcal{C}$  is a single constant:  $\deg(\mathbf{c}) = Q$  for all  $\mathbf{c} \in \mathcal{C}$ . In this case, the speedup simplifies to the following.

$$\frac{T(\mathbf{w}, x) + Q|\mathcal{C}|m}{T(\mathbf{w}, x) + dm|\mathcal{C}| + Q|\mathcal{C}|\lambda_0} \quad (5)$$

If we narrow our scope to only consider the speedup restricted to the operations of  $\mathcal{A}$  affected by sparselets, we can ignore the  $T(\mathbf{w}, x)$  terms and note that the  $|\mathcal{C}|$  factors cancel.

$$\frac{Qm}{dm + Q\lambda_0} \quad (6)$$

This narrowing is justified in the multiclass classification case (with  $K$  classes) where the cost  $T(\mathbf{w}, x)$  amounts to computing the maximum value of  $K$  numbers, which is negligible compared to the other terms. The computation graph for a simple multiclass example with  $K = 3$  is given in Fig. 1.