

How do I implement the following efficiently?

$[X \in \mathbb{R}^{N \times p}, y \in \mathbb{R}^N, w \in \mathbb{R}^p, p \gg N]$

$f(X, y, w) = \sum_{k: |\beta_k| > 0} (x_j, x_k) w_k = (\tilde{X}[:,j], \tilde{X}) (\tilde{w})$

```
for i in n_iterations
    for j in p
        f(X, y, w)
    end
end
```

background:

- w starts out dense but gets sparser as $i \rightarrow n_iterations$
- if $w[j] = 0$ it will stay zero
- $(\tilde{X}[:,j], \tilde{X})$ is recalculated for each iteration of the outer loop, but it can't be cached from the beginning since $p \times p$ is too large to fit in memory.
- implementation will be in Cython
- `cblas_ddot` could be used for the scalar product
- [current implementation](#) (indexing of cached values is buggy)