Sparse approximation

We will consider, from several different perspectives, the problem of finding a *sparse representation* of a signal. For simplicity, the majority of our discussion will take place in the context of vectors $f \in \mathbb{R}^n$. Everything that is said here automatically extends to continuous-times signals that lie in a fixed finite-dimensional subspace in a straightforward way.

We start with a problem that has an easy solution. Say that Ψ is an orthobasis for $\mathbb{R}^n$, which in this case we can take as an $n \times n$ matrix whose columns are $\psi_1, \psi_2, \ldots, \psi_n$, and we wish to find the best approximation of a a given $f \in \mathbb{R}^n$ using a fixed number of elements from Ψ. If we give ourselves a budget of S-terms, the best S-term approximation of f in the basis Ψ is defined as the solution to the following optimization program:

$$\min_{\beta \in \mathbb{R}^n} \|f - \Psi\beta\|_2^2 \quad \text{subject to} \quad \#\{\gamma : \beta[\gamma] \neq 0\} \leq S. \qquad (1)$$

It is customary to use the notation $\|\beta\|_0$ for the number of non-zero terms in β, although it should be stressed that $\|\cdot\|_0$ does not obey the properties of a norm. Since Ψ is an orthogonal transform, Parsaval tells us that

$$\|f - \Psi\beta\|_2^2 = \|\Psi^*(f - \Psi\beta)\|_2^2 = \|\alpha - \beta\|_2^2$$

where $\alpha = \Psi^* f$ are the Ψ-transform coefficients of f. So we can rewrite the optimization program above as

$$\min_{\beta \in \mathbb{R}^n} \|\alpha - \beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_0 \leq S.$$

It is easy to argue that the solution to this problem is to have β take the S-largest terms from α and set the rest to zero. Thus the algorithm for solving (1) is

1. Compute $\alpha = \Psi^* f$.

2. Find the locations of the S-largest terms in α; call this set Γ.
3. Set
$$\tilde{\beta}_S[\gamma] = \begin{cases} \alpha[\gamma] & \gamma \in \Gamma \\ 0 & \gamma \notin \Gamma \end{cases}$$
4. Compute $\tilde{f}_S = \Psi\tilde{\beta}_S$.

When the basis is overcomplete, choosing the best S-term approximation is nowhere near as straightforward. To separate this problem from our previous discussion (and to be consistent with the existing literature), we will change notation slightly. We have a fixed collection of functions $\psi_1, \psi_2, \ldots, \psi_p$, which we call a *dictionary*, from which we would like to choose some subset to approximate a given signal $f \in \mathbb{R}^n$. That is, we wold like to find a subset $\Gamma \subset \{1, 2, \ldots, p\}$ so that

$$f \approx \sum_{\gamma \in \Gamma} \alpha_\gamma \psi_\gamma$$

for some coefficient set $\{\alpha_\gamma,\ \gamma \in \Gamma\}$. The idea is that $p >> n$, and different signals of interest will use different subsets Γ. Given a signal, we want a systematic way of choosing a representation which uses as few terms as necessary.

Unless otherwise stated, we will use the convention that the columns of Ψ are normalized, $\|\psi_k\|_2 = 1$. We will find it useful to organize the $\{\psi_k\}$ as columns in an $n \times p$ matrix Ψ:

$$\Psi = \begin{bmatrix} | & | & & | \\ \psi_1 & \psi_2 & \cdots & \psi_p \\ | & | & & | \end{bmatrix}.$$

Now the task is to find a sparse vector α such that

$$f \approx \Psi\alpha.$$

We can visualize this task as selecting an appropriate subset of the columns from Ψ to build up f — each subset of columns spans a different subspace of $\mathbb{R}^n$, so in some sense we choosing from a family of subspaces the one which is closest to containing f.

In general, finding the best representation in an overcomplete is a problem with combinatorial complexity (NP hard). But it can still be attacked in a principled manner. In the next two lectures, we will talk about two very different frameworks for approaching this problem:

Greedy algorithms Matching pursuit (MP) and the closely related Orthogonal Matching Pursuit (OMP) operate by iterative choosing columns of the matrix. At each iteration, the column that reduces the approximation error the most is chosen.

Convex programming Relaxes the combinatorial problem into a closely related convex program, and minimizes a global cost function. The particular program, based on ℓ_1 minimization, we will look at has been given the name Basis Pursuit in the literature.

I will stress that there are other sparse approximation algorithms that do not quite fit into these two categories (approximate message passing, iterative thresholding, etc.), but learning about these two types will give you a good start on this subject.

Soft thresholding and ℓ_1 minimization

Since we will be looking a lot at using ℓ_1 minimization for reconstruction sparse signals, it is worth it to point out an interesting connection between ℓ_1 minimization and soft-thresholding.

Consider the following optimization program,

$$\min_\beta \; \|\beta\|_{\ell_1} + \frac{\lambda}{2}\|\Psi\beta - f\|_2^2,$$

where Ψ is an orthobasis. Just as in the ℓ_0 case above, this program is equivalent to

$$\min_\beta \; \|\beta\|_{\ell_1} + \frac{\lambda}{2}\|\beta - \alpha\|_2^2,$$

where $\alpha = \Psi^* f$. We can now break the functional above apart term-by-term:

$$\min_\beta \; \sum_{i=1}^{n} |\beta(i)| + \frac{\lambda}{2}(\beta(i) - \alpha(i))^2.$$

Since each term in the sum above is positive, we can optimize term-by-term. It is not hard to show that the solution is

$$\hat{\beta}(i) = \max(\alpha_i - \lambda\,\mathrm{sgn}(\alpha_i), 0),$$

that is, we acquire the solution vector $\hat{\beta}$ by **soft thresholding** the entries in the transform coefficient vector α.

Matching Pursuit

Matching pursuit (MP) is a relatively straightforward algorithm for selecting a (hopefully small) subset of a dictionary with which to represent a given signal.

MP is very easy to interpret: it iteratively selects columns from Ψ to use in the representation while keeping track of the current approximation and the residual (the "left overs").

It is a **greedy algorithm** in that at each iteration, it chooses the column that is most closely correlated with the residual regardless of history or future implication of this choice.

MP comes with very few performance guarantees, and in fact it is relatively straightforward to construct examples of simple problems where it fails spectacularly.

Nevertheless, understanding the algorithm can be instructive, and many of the qualitative principles it is based on will show up later.

Matching Pursuit Algorithm:

Given a signal $f \in \mathbb{R}^n$, an $n \times p$ dictionary Ψ with normalized columns $\|\psi_\gamma\|_2 = 1$, and a convergence criteria $\epsilon > 0$,

1. Set $f_0 = 0,\ R_0 = f,\ k = 0$
   (f_0 is the initial approximation, and R_0 is the initial residual)

2. Select the index γ_k such that $|\langle R_k, \psi_{\gamma_k}\rangle|$:

$$\gamma_k = \arg\max_{1\le\ell\le p} |\langle R_k, \psi_\ell\rangle|$$

3. Set

$$\begin{aligned} f_{k+1} &= f_k + \langle R_k, \psi_{\gamma_k}\rangle\psi_{\gamma_k} \\ R_{k+1} &= R_k - \langle R_k, \psi_{\gamma_k}\rangle\psi_{\gamma_k} \end{aligned}$$

   (Add in the new contribution to the approximation, subtract it from the residual)

4. Repeat 2–3 until $\|f - f_k\|_2 \le \epsilon$ or we have the desired number of terms

Note that at each step

$$R_{k+1} \perp \psi_{\gamma_k}$$

since

$$\begin{aligned} \langle R_{k+1}, \psi_{\gamma_k}\rangle &= \langle R_k, \psi_{\gamma_k}\rangle - \langle R_k, \psi_{\gamma_k}\rangle\langle\psi_{\gamma_k}, \psi_{\gamma_k}\rangle \\ &= 0 \qquad (\text{since } \langle\psi_{\gamma_k}, \psi_{\gamma_k}\rangle = 1) \end{aligned}$$

Thus we can expand the energy in the signal as

$$\begin{aligned}
\|f\|_2^2 &= |\langle f, \psi_{\gamma_0}\rangle|^2 + \|R_1\|_2^2 \\
&= |\langle f, \psi_{\gamma_0}\rangle|^2 + |\langle R_2, \psi_{\gamma_1}\rangle|^2 + \|R_2\|_2^2 \\
&\vdots \\
&= \sum_{k'=0}^{k-1} |\langle R_{k'}, \psi_{\gamma_{k'}}\rangle|^2 + \|R_k\|_2^2
\end{aligned}$$

So at each step, we are selecting the atom ψ_{γ_k} which makes the residual error as small as possible.

Important note: The *same column* can be selected on multiple iterations — and in fact in general there are multiple columns which get selected an *infinite* number of times.

[Example in MATLAB]

Even though MP takes an infinite number of iterations in general, it converges exponentially:
Theorem: There exists a $\lambda > 0$ such that for all $k \geq 0$

$$\|R_k\|_2 \ \leq \ 2^{-\lambda k}\|f\|_2$$

(For proof of this see, for example, Chapter 9 in Mallat, *A Wavelet Tour of Signal Processing*.)

Orthogonal Matching Pursuit

OMP is a variant of MP: it selects atoms (columns) iteratively from Ψ, but it uses as the current approximate an *orthogonal projection* onto the space spanned by the atoms selected so far.

The main consequence of this modification is the OMP is guaranteed to converge in n (or fewer) iterations. We can interpret the algorithm as building up a subspace in which the approximation lives — each iteration adds a dimension to this subspace, and so at most n iterations are possible.

Each iteration of OMP is (much) more expensive than MP, since the columns have to be orthogonalized at every step.

Orthogonal Matching Pursuit Algorithm:
Given a signal $f \in \mathbb{R}^n$ and an $n \times p$ dictionary Ψ,

1. Set $f_0 = 0$, $R_0 = f$
2. Select γ_k as in MP:
$$\gamma_k \;=\; \arg\max_{1\leq\ell\leq p} \;|\langle R_k, \psi_\ell\rangle|$$
3. Set
$$u'_k = \psi_{\gamma_k} - \sum_{\ell=0}^{k-1} \langle \psi_{\gamma_k}, u_\ell\rangle\, u_\ell, \qquad u_k = \frac{u'_k}{\|u'_k\|_2}$$
$$f_{k+1} = f_k + \langle R_k, u_k\rangle u_k$$
$$R_{k+1} = R_k - \langle R_k, u_k\rangle u_k$$

Note that at each k
$$f = P_{U_k} f + R_k$$
where $U_k = \operatorname{span}\{u_1, \ldots, u_k\} = \operatorname{span}\{\psi_{\gamma_1}, \ldots, \psi_{\gamma_k}\}$.

Basis Pursuit

Basis Pursuit (BP) is an entirely different approach to finding a sparse representation. Rather than building up the signal f by adding contributions from iteratively selected columns, it sets up the problem as an **optimization program**.

The problem setup is the same as the above, we have a signal $f \in \mathbb{R}^n$ that we want to write as a linear combination of (a hopefully small number of) atoms from a $n \times p$ dictionary Ψ. That is, we would like to choose an $\alpha \in \mathbb{R}^p$ such that

$$f = \Psi\beta.$$

When $p > n$, Ψ is rectangular and there will in general be many such β. Out of all the valid decompositions

$$\{\beta \ : \ \Psi\beta = f\} \subset \mathbb{R}^p,$$

we want the one that uses the smallest number of terms. Another way to say that is we want to solve

$$\min \ \#\{\gamma \ : \ \beta(\gamma) \neq 0\} \quad \text{subject to} \quad \Psi\beta = f.$$

The number of nonzero terms in a vector is often called its ℓ_0 norm, though it should be clear that this is not at all a valid norm. Rewriting the program above, we want to solve

$$\min \ \|\beta\|_{\ell_0} \quad \text{subject to} \quad \Psi\beta = f.$$

The program above is easy enough to state, but it is basically impossible to solve — it is a combinatorial optimization program.

Basis Pursuit (BP) simply takes the program above and replaces the ℓ_0 norm with the ℓ_1 norm:

$$\min \ \|\beta\|_{\ell_1} \quad \text{subject to} \quad \Psi\beta = f.$$

This is, in a formal way, the natural convex relaxation of the sparest decomposition problem. Since the program is convex, it is tractable.

BP as a linear program

In fact, Basis Pursuit can be recast as a linear program using a simple trick. We are interested in

$$\min_{\beta} \ \sum_{\gamma=1}^{p} |\beta(\gamma)| \quad \text{subject to} \quad \Psi\beta = f,$$

The (equality) constraints above are linear, but the functional is nonlinear. However, the functional is *piecewise linear*, which means that it can be transformed into a linear functional by adding a few more constraints.

Introducing the auxiliary variable $u \in \mathbb{R}^p$, and equivalent program to BP is

$$\min_{u,p} \ \sum_{\gamma=1}^{p} u(\gamma) \quad \text{subject to} \quad \Psi\beta = f, \quad -u(\gamma) \leq \beta(\gamma) \leq u(\gamma) \ \forall\gamma$$

This is a linear program, and it has the same solution as BP. Thus

$$\text{Basis Pursuit} = \text{Linear Programming}$$

Efficiency of Basis Pursuit

In fact, we can show that given an $n \times p$ matrix Ψ and a vector $f \in \mathbb{R}^n$, the solution to

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_1 \quad \text{subject to} \quad \Psi\beta = f$$

never has more than n non-zero terms. So in a very concrete way, BP is selecting a *basis* (set of n linearly independent vectors) from the dictionary Ψ.

We prove this as follows. Let α be a feasible vector ($\Psi\alpha = f$) supported on $\Gamma \subset \{1, 2, \ldots, p\}$ — that is, Γ contains the indices of the non-zero locations in α. We will use s as the size of Γ:

$$s = |\Gamma| = \#\text{non-zero terms in } x.$$

We will show that if $s > n$, then there necessarily exists a vector with $s - 1$ non-zeros terms that is also feasible.

Let Ψ_Γ be the $n \times s$ matrix containing the s columns of Ψ indexed by Γ. Let $z \in \mathbb{R}^s$ be a vector containing the signs of α on Γ

$$z = \{\text{sgn}(\alpha(\gamma)),\ \gamma \in \Gamma\}$$

and let $z' \in \mathbb{R}^p$ be the extension of z

$$z'(\gamma) = \begin{cases} \text{sgn}(\alpha(\gamma)) & \gamma \in \Gamma \\ 0 & \gamma \notin \Gamma \end{cases}$$

Suppose $s > n$. Then Ψ_Γ has more columns than rows, and hence has a non-trivial null space. Let $h \in \text{Null}(\Psi_\Gamma)$ be any vector in this null space, and let h' be the extended version of h:

$$h'(\gamma) = \begin{cases} h(\gamma) & \gamma \in \Gamma \\ 0 & \gamma \notin \Gamma \end{cases}.$$

It should be clear that $h' \in \text{Null}(\Psi)$. We may assume without loss of generality that

$$\langle h, z\rangle \leq 0,$$

since otherwise we could just use $-h$ (since of course if $h \in \text{Null}(\Psi_\Gamma)$, then $-h \in \text{Null}(\Psi_\Gamma)$). For ϵ small enough, we will have on Γ:

$$\text{sgn}(\alpha + \epsilon h) = \text{sgn}(\alpha) = z,$$

and so

$$\begin{aligned}
\|\alpha + \epsilon h'\|_1 &= \sum_{\gamma\in\Gamma} \text{sgn}(\alpha(\gamma) + \epsilon h(\gamma))(\alpha(\gamma) + h(\gamma)) \\
&= \sum_{\gamma\in\Gamma} z(\gamma)\alpha(\gamma) + \epsilon \sum_{\gamma\in\Gamma} z(\gamma)h(\gamma) \\
&= \|\alpha\|_1 + \epsilon\langle z, h\rangle \\
&\leq \|\alpha\|_1
\end{aligned}$$

since $\langle z, h\rangle \leq 0$.

We can move in the direction h lowering the ℓ_1 norm until

$$\text{sgn}(\alpha(\gamma) + \epsilon h(\gamma)) \neq \text{sgn}(\alpha(\gamma)).$$

This happens exactly when one of the elements of $\alpha + \epsilon h$ is driven exactly to zero; call the corresponding step size ϵ_0. By construction

$$\alpha_1 = \alpha + \epsilon_0 h',$$

is feasible, has smaller ℓ_1 norm than α, and has one fewer non-zero element than α.

The entire argument above relied on finding a non-zero vector in the null space of Ψ_Γ. We are guaranteed that such a vector exists if $s > n$, so the argument above implies that the solution (or at least one of the solutions if the solutions if there is more than one) to basis pursuit has at most n non-zero terms.