Identifying Localized Signals On Graphs

1 — Frequency Approach

One possible way to approach is this problem is to decompose the signal s into a linear combination of diffusion wavelets at various scales. Then the more components the signal has in larger scales, the more spread the signal is.

In particular, we define a family of diffusion wavelets like so: suppose we have a diffusion operator M on our graph, with *columns* giving probability distributions from points. For example, if we have spatial data with affinity matrix A and degree matrix D, $M = D^{-1}A$. We define:

$$P = \frac{1}{2}(I + M) \text{ a lazy random walk operator}$$

Typical wavelets use $\Psi_j = P^{2^{j-1}} - P^{2^j}$. For our purposes, we will use $\Psi_j = P^{2_j}$ with $\Psi_0 = I$, as I have found that this construct works better. We will also see that these are, spatially speaking, a more intuitive choice. So technically we are not working with wavelets, but let's be abusive and call them wavelets anyway. For each set matrix Ψ_j, especially for large j, we find that the columns are often redundant. We can trim them down using a matrix $\tilde{\Psi_j}$, whose columns (subset of Ψ_j's) can span the original Ψ_j with error ϵ:

$$\frac{1}{\|\Psi_j\|}\|^2 \|\Psi_j - \tilde{\Psi_j}(\tilde{\Psi_j}^T \tilde{\Psi_j})^{-1} \tilde{\Psi_j^T} \Psi_j\|^2 \leq \epsilon \tag{1}$$

Think of $\tilde{\Psi_j}$ as a selection of wavelets at scale j that span the graph. In general, $|\tilde{\Psi_j}|$ decreases in j and increases in ϵ. Here's an algorithm for obtaining such a $\tilde{\Psi_j}$. For example,

Algorithm 1 Obtaining $\tilde{\Psi}_j$

```
**Input:** A set of diffusion wavelets $\Psi_j$ and a tolerance $\epsilon \geq 0$
**Ouput:** $\tilde{\Psi}_j$, whose columns are a subset of $\Psi_j$ such that they nearly span $\Psi_j$
Let $\tau \leftarrow \epsilon \|\Psi_j\|^2$
$Q, R, p \leftarrow Pivotal - QR(\Psi_j)$
$R_{norm} \leftarrow \|\Psi_j\|^2$
**while** for $i$ in $1...n$ **do**
    $R_{norm} \leftarrow R_{norm} - \sum_{j=i}^{n} R_{i,j}^2$
    **if** $R_{norm} \leq \tau$ **then**
        return $\Psi_j(p(1), p(2)...p(i))$
    **end if**
**end while**
```

This algorithm is proven to be correct in the Appendix. If n is very large, we could also input a randomly sampled set of columns into the algorithm. Think of Ψ_j as being some spanning set of wavelets. Of course, for larger and larger scales, we might imagine that we can take relatively few of these, as they can cancel a great deal of low frequency signals. Here is such an example:

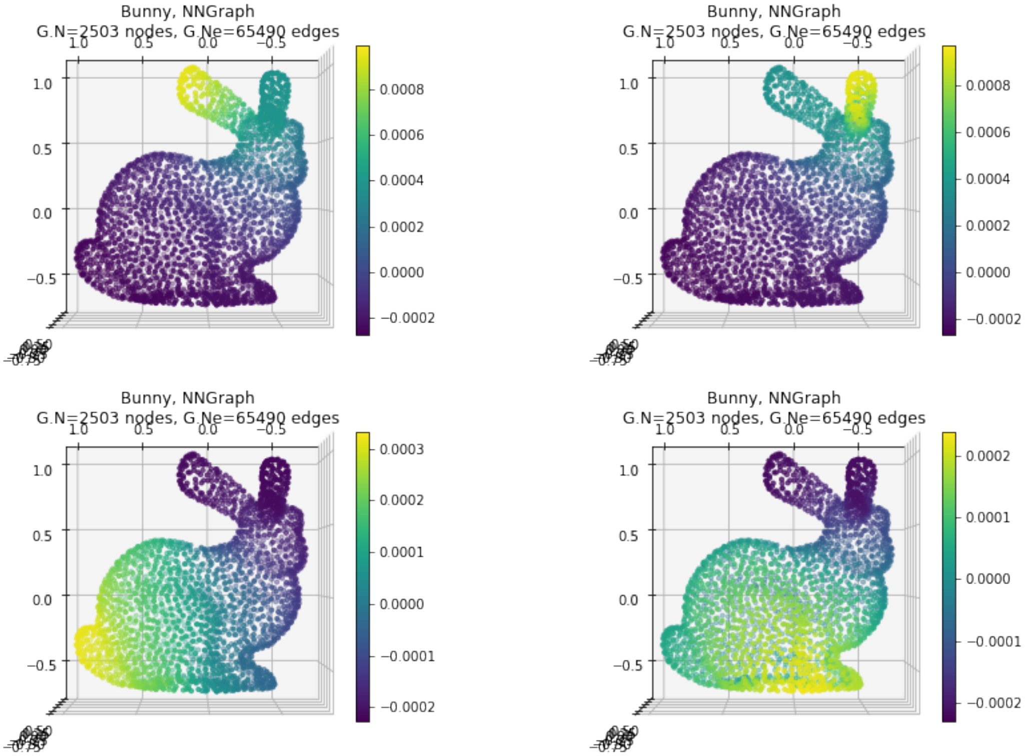

Figure 1: Example of $\tilde{\Psi}_j$. Notice that the original matrix Ψ_j is 2503×2503. In contrast, $\tilde{\Psi}_j$ is 2503×4. The four "spanning" columns as outputted by Algorithm 1 using tolerance $\epsilon = 10^{-3}$ are shown above.

Detecting Local Signals

With this out of the way, we can present a possible algorithm for detecting "localized" signals. We first create a matrix $\Phi = \begin{bmatrix}\Psi_0 & \Psi_1..\Psi_J\end{bmatrix}$. That is, it is the horizontal concatenation of all of our wavelets. We then define a weight vector $\omega = \begin{bmatrix}1...1 & 2...2 & .... & 2^J...2^J\end{bmatrix}$. For a more efficient implementation, we could also set Φ using the $\tilde{\Phi_j}$'s instead; this would also help avoid overfitting. Regardless, normalize the columns of Φ for consistency. Now, given a signal s (assume it is normalized), we can estimate a sparse linear combination x of columns of Φ that approximate s. For our purposes, let's employ the matching pursuit algorithm. Recall that this algorithm attempts to solve, given some N,

$$\min_x \|\Phi x - s\|^2 \text{ subject to } \|x\|_0 \leq N$$

So suppose we have an algorithm $MP(\Phi, s, N)$ that outputs the desired x. We can then set a measure of locality ℓ of s like so:

$$\ell(s;x) = \omega \cdot |x| = \sum_i |x_i|\omega_i = \sum_j 2^j \sum_{k:k \text{ corresponds to scale j}} |x_k|$$

Which motivates the following algorithm for finding localized signals:

Algorithm 2 Calculating Smoothness

```
Input: A diffusion operator $M$ and signal $s$ defined on a set $\mathcal{X}$. SPARSE, a boolean variable encoding whether to use the full $\Psi_j$ or $\tilde{\Psi_j}$, an $\epsilon$ to dictate $\tilde{\Psi_j}$. OMP a routine for matching pursuit.
Output: $\ell(s)$, a measure of localization of $s$ on the set $\mathcal{X}$.

$P \leftarrow \frac{1}{2}(I + M)$
$J \leftarrow \log(|\mathcal{X}|)$ (unless otherwise specified)
$\Psi_0 \leftarrow I$
for $j \in [J]$ do
    $\Psi_j \leftarrow P^{2^j}$
    Normalize the columns of $\Psi_j$
    if SPARSE then
        Calculate $\tilde{\Psi_j}$ given $\Psi_j, \epsilon$ per algorithm 1
    end if
end for
$\omega \leftarrow [1, 1...2, 2...2^J, 2^J]$ with as many $2^j$'s as columns in $\Psi_j$ or $\tilde{\Psi_j}$
$\Phi \leftarrow \begin{bmatrix}\Psi_0 & \Psi_1.. & \Psi_J\end{bmatrix}$ or $\Phi = \begin{bmatrix}\tilde{\Psi_0} & ... & \tilde{\Psi_J}\end{bmatrix}$ as appropriate.
Normalize columns of $\Phi$
$x \leftarrow OMP(\Phi, \frac{s}{\hat{s}})$ (we normalize $s$ in case we'd like to compare different signals)
$\ell(s) \leftarrow \langle w, |x| \rangle$, where $|x|$ is the elementwise absolute value of $x$
return $\ell(s)$.
```

An alternative algorithm, for the reduced set, is to use $\tilde{\Psi_j}$ to calculae $\Psi_{\tilde{j+1}}$. This would go like so:

Algorithm 3 Calculating Smoothness (Alternative)

```
**Input:** A diffusion operator $M$ and signal $s$ defined on a set $\mathcal{X}$. SPARSE, a boolean variable encoding whether to use the full $\Psi_j$ or $\tilde{\Psi}_j$, an $\epsilon$ to dictate $\tilde{\Psi}_j$. OMP a routine for matching pursuit.
**Output**: $\ell(s)$, a measure of localization of $s$ on the set $\mathcal{X}$.

$P \leftarrow \frac{1}{2}(I + M)$
$J \leftarrow \log(|\mathcal{X}|)$ (unless otherwise specified)
$\Psi_0 \leftarrow I$
for $j \in [J]$ do
    $\Psi_j \leftarrow P^{2^{j-1}} \tilde{\Psi}_{j-1}$
    Normalize the columns of $\Psi_j$
    Set $\tilde{\Psi}_j$ per our algorithm.
end for
$\omega \leftarrow [1, 1...2, 2...2^J, 2^J]$ with as many $2^j$'s as columns in $\tilde{\Psi}_j$
$\Phi \leftarrow \begin{bmatrix} \tilde{\Psi}_0 & ... & \tilde{\Psi}_J \end{bmatrix}$ as appropriate.
Normalize columns of $\Phi$
$x \leftarrow OMP(\Phi, \frac{s}{\hat{s}})$ (we normalize $s$ in case we'd like to compare different signals)
$\ell(s) \leftarrow \langle w, |x| \rangle$, where $|x|$ is the elementwise absolute value of $x$
return $\ell(s)$.
```

A benefit to this version is that Ψ_j has as many rows as $\tilde{\Psi}_{j-1}$, so our work keeps decreasing, which becomes highly beneficial at larger and larger scales. Note it is more efficient to caalculate P via repeated squaring. This way, we calculate $P^{2^j} \leftarrow (P^{2^{j-1}})^2$ in time $\mathcal{O}(n^2)$. So for J scales, runtime is at worst $\mathcal{O}(n^2 \log(n))$. Compare this to an eigendecomposition, which could take time up to $\mathcal{O}(n^3)$. So for large n and relatively few scales, this is more effective.

Example on the Bunny Graph

As a start, let's use the redundant ϕ, which contains all wavelets at all scales. As a sanity check, we can consider graph signals of the form $s = M^t \delta_i$ and verify that, as we increase t, ℓ increases.

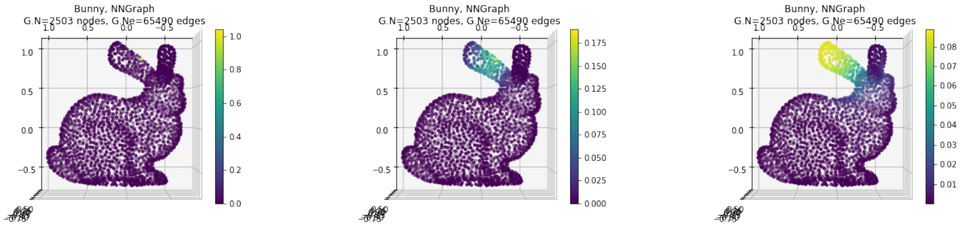

Figure 2: Here we have visualized signals of the form $M^t \delta_i$ with $t = 0, 2^3, 2^6$ and $i = 1400$. We see that as we increase t, the signal should intuitively be less "local." Accordingly, the respective measures of locality are approximately $1, 344$, and 413

We could also repeat this using our QR-reduced dictionaries. In this case, the signals are the same, but the estimates of locality are $24, 357$, and 492. So wee see that in the simplest case, our algorithm does what it's supposed to.

Moving Signals Across the Sphere

Another thing we might imagine doing is fixing one signal on a sphere and moving another to see how the relative localization changes. This is visualized in the below diagram:

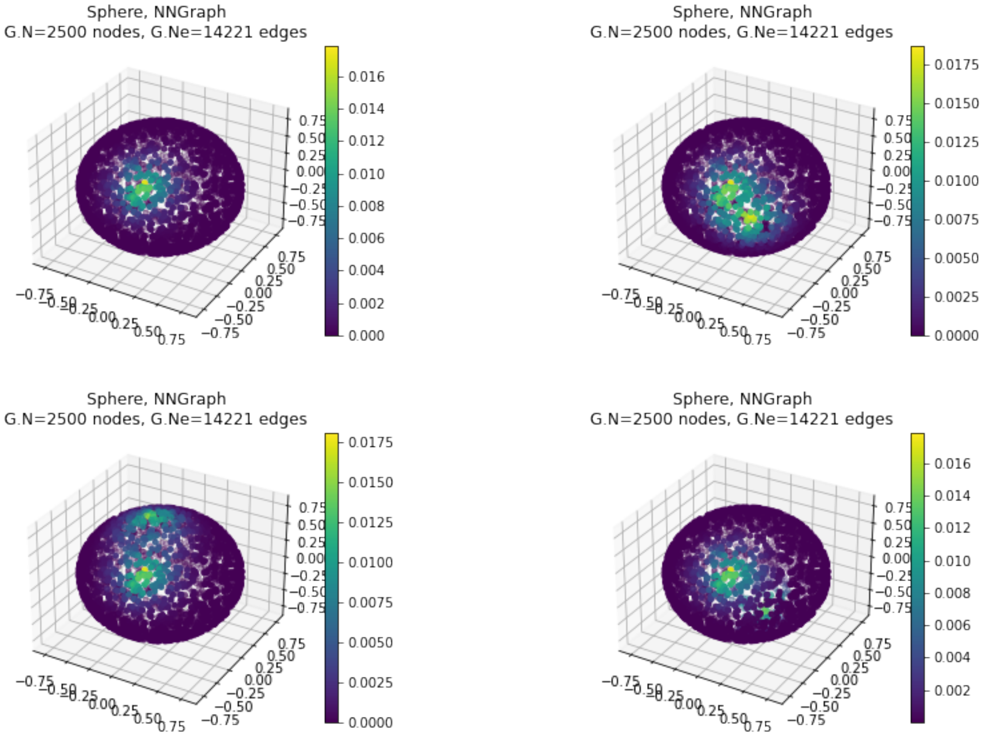

Figure 3: We can see that as we continue this process, we would hope that the signal is identified as less and less localized. But once the signal becomes more and more bimodal, the level of localization stabilizes. What we see is that the corresponding levels of localizations $32, 57, 46$ and 44. Here we use the sparser wavelets with $\epsilon = 0.001$, $N = 50$

This example demonstrates that, at the bare minimum, the most centered signal indeed receives the lowest score. Then, the large jump in value from the first to the second signals that "wider" signals are weighed more heavily than bimodal ones. Whether this is desirable is questionable; maybe it is, maybe it isn't. If it is not, it may be worth revisiting the weight vector ω.

Non-Frequency Approach

Another possible approach is to avoid scanning the signal using frequencies altogether and use a quantity which captures the "spread" of a signal. Suppose we have our set $\mathcal{X}$ and, rather than a diffusion operator, some distance $d(\cdot,\cdot)$ defined on $\mathcal{X} \times \mathcal{X}$. In this case, if we have a nonnegative signal $s \in \mathbb{R}^n_+$, we can normalize s by taking $\hat{s} = \frac{s}{\sum_{x \in X} s(x)}$, and view $\hat{s}$ as a probability distribution. Now we can examine the quantity:

$$\mathcal{S}(s) = \sum_{x,y} \hat{s}(x)\hat{s}(y)d(x,y)$$

Notice that in the simplest case, if s is binary on $\mathcal{X}$ (ie $s \in \{0,1\}^{|\mathcal{X}|}$), we couls say $s = 1$ over a set $A \subset X$ and observe

$$\mathcal{S}(s) = \frac{1}{|A|^2} \sum_{x,y \in A^2} d(x,y)$$

Which can be viewed as average distance between vertices where s is "on." So then $\mathcal{S}(s)$ for general s serves as a continuous extension of this. Similarly, if we think of $\hat{s}$ as a distribution, then, the joint distribution of (x,y) is simply $\hat{s}(x)\hat{s}(y)$. So then we have the interpretation,

$$\mathcal{S}(s) = \sum_{x,y \in \mathcal{X} \times \mathcal{X}} \mathbb{P}\{x,y\}d(x,y) = \mathbb{E}_{x,y \sim \hat{s}}[d(x,y)]$$

Note that if $\mathcal{X}$ is finite, then we can store $d(\cdot,\cdot)$ in a matrix D. So then,

$$\mathcal{S}(s) = \frac{s^T D s}{s^T \mathbf{1}\mathbf{1}^T s}$$

So in practice, we could choose $d(\cdot,\cdot)$ in any number of ways. If $\mathcal{X} \in \mathbb{R}^n$, we could simply use the Euclidean distance between data points and store them in a matrix D. If X is not spatial, d could be the length of the shortest path. We could also let $d(x,y)$ be, for example, the diffusion distance between x and y, given by $\|P(x,\cdot) - P(y,\cdot)\|^2_{L^2(X,dP/d\pi)}$ where $P : \mathcal{X} \times \mathcal{X}$ is a probability kernel with stable distribution π. As shown by Coifman & Lafon, this would be equivalent to $\|\Psi(x,\cdot) - \Psi(y,\cdot)\|^2$ where Ψ denotes the matrix whose columns are right eigenvectors of P. Another possible choice could be the distance used by PHATE, given by $\sqrt{\|\log P(x,\cdot) - \log P(y,\cdot)\|}$.

Regardless of the matrix D, we know that the level curves of $\mathcal{S}$ are given by ellipsoids on a d dimensional simplex. Thus, we can compare our signal s to other values in the level set:

$$\{x : \sum_i x(i) = 1, x^T D x = \mathcal{S}(s)\}$$

Statistical Inference

Without loss of generality, assume s is a probabiliity distribution. We would like to construct a prior distribution for s. One reasonable hypothesis, call it H_0 for localization may be that there exists a center $x^\star \in X$ such that $s(x) = k(x, x^\star) + \mathcal{N}(0, \sigma)$ where k is some kernel function which measures the affinity between x and $x^\star$. Thus, we think of s as decreasing in distance from $x^\star$ plus Gaussian noise. We can define a likelihood function:

$$\begin{aligned}\ell(s; x^\star, k, \sigma) &= \prod_{x \in \mathcal{X}} \mathbb{P}\{k(x, x^\star) + \mathcal{N}(0, \sigma) = s(x)\} \\ &= \prod_{x \in \mathcal{X}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(k(x, x^\star) - s(x))^2}{2\sigma^2})\end{aligned}$$

Thus, we can define log-likelihood:

$$\log(\ell(s; x^\star, k, \sigma)) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|k(x^\star, \cdot) - s(\cdot)\|^2$$

Where $k(x^\star, \cdot)$ is the vector consisting of distances from $x^\star$ to each x. As a simplifying assumption, say $k(x, y; h) = \exp(-\frac{\|x-y\|^2}{2h^2})$, that is a Gaussian kernel with bandwidth h. We can use some gradient descent algorithm to find the maximimum likelihood estimates of h, σ, and $x^\star$. In this way, we obtain some distribution π from which we think s arose. We can then estimate the distribution of $\mathcal{S}(f)$ for $f \sim \pi$. If the value of $\mathcal{S}(s)$ is extremal in this distribution, this could give evidence against this model of generation. Note that the corresponding p-value would *not* be a measure of locality, but rather a measure of well a centered model can appproximate s.

Pivotal Statistics

We might be interested in developing a statistic which does not explicitly depend on $k, x^\star$, or σ which is pivotal under this mode of generation.

Appendix

Algorithm Correctness

Proposition: $\tilde{\Psi_j}$ as outputted by Algorithm 1 satisfies equation (1). *Proof:* This is equivalent to proving:

$$\|\Psi_j - \tilde{\Psi_j}(\tilde{\Psi_j}^T\tilde{\Psi_j})^{-1}\tilde{\Psi_j^T}\Psi_j\|^2 \leq \epsilon\|\Psi_j\|^2 = Q \cdot R(p(1)...p(i))$$

Recall that our algorithm first takes a pivotal QR decomposition of Ψ_j. Our algorithm first finds i such that $\|\Psi_j\| - \sum_{k=0}^{i}\sum_{j=k}^{n} R_{k,j}^2 \leq \epsilon\|\Psi_j\|$. Or equivalently, so that $\|\sum_{k=0}^{i}\sum_{j=k}^{n} R_{k,j}^2\| \geq (1-\epsilon)\|\Psi_j\|$. First, observe that $\|\Psi_j\|^2 = \|QR\|^2 = \|R\|^2$ because Q is orthonormal.

Observe that for all $k \leq i$ that $R_{k,k} > 0$ per how the pivotal QR decomposition algorithm works. If any columns are linearly dependent, those will be attributed to indices higher than i. I also claim that. Thus, $C(\tilde{\Psi_j}) = C(q_1...q_i)$, where $q_1...q_i$ are the first i columns of Q. Thus, if we assemble these columns into a matrix Q_i, we find that the projection matrix onto $C(Q_i)$ is the same as the projection onto $\tilde{\Psi_j}$. Thus,

$$\|\Psi_j - \tilde{\Psi_j}(\tilde{\Psi_j}^T\tilde{\Psi_j})^{-1}\tilde{\Psi_j^T}\Psi_j\|^2 = \|\Psi_j - Q_iQ_i^T\Psi_j\|^2$$

Suppose the columns of Ψ_j are $\psi_j^1...\psi_j^n$. We can also compute the above norm by column:

$$= \sum_k \|(I - Q_iQ_i^T)\psi_j^k\|^2$$

But since Q is orthonormal and ψ_j is in the span of Q, we can represent $(I - Q_iQ_i^T)$ by the matrix $\hat{Q}_i\hat{Q}_i^T$, where $\hat{Q}_i$ contains columns $i+1...n$ of Q. So:

$$= \sum_k \|\hat{Q}_i\hat{Q}_i^T\psi_j^k\|^2$$

Note that p is a permutation, so we can rewrite the order as:

$$= \sum_k \|\hat{Q}_i\hat{Q}_i^T\psi_j^{p(k)}\|^2$$

Of course, observe that by the way the QR decomposition works, column $p(k)$ of Ψ_j is:

$$\psi^{p(k)} = \sum_{l \leq k} q_l R_{l,k}$$

Plugging this in, our original norm is: r

$$= \sum_k \|\hat{Q}_i\hat{Q}_i^T \sum_{l \leq k} q_l R_{l,k}\|^2 = \sum_k \|\sum_{l \leq k} \hat{Q}_i\hat{Q}_i^T q_l R_{l,k}\|^2$$

Since the columns of q_l are orthonormal:

$$= \sum_k \|\sum_{l \leq k} \mathbf{1}\{l > i\} q_l R_{l,k}\|^2$$

Applying Parseval's identity:

$$= \sum_k \sum_{l \le k} \mathbf{1}\{l > i\} R_{l,k}^2$$

Which is precisely equal to the *sums of squares* of rows $i+1...n$ of R. Thus,

$$\|\Psi_j - \tilde{\Psi}_j(\tilde{\Psi}_j^T\tilde{\Psi}_j)^{-1}\tilde{\Psi_j^T}\Psi_j\|^2 = \sum_{k>i}\sum_{l \ge k} R_{k,l}^2 = \|R\|^2 - \sum_{k \le i}\sum_{l \ge k} R_{k,l}^2$$

By design,

$$= \|\Psi_j\|^2 - \sum_{k \le i}\sum_{l \ge k} R_{k,l}^2 \le \epsilon\|\Psi_j\|^2$$

Thus, the claim must be true as claimed □.

Maximum Likelihood Estimates (WORK IN PROGRESS!!!)

Recall that we have the following log likelihood:

$$\log(\ell(s; x^\star, d)) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|d(x^\star, \cdot) - s(\cdot)\|^2$$

$$= -\frac{n}{2}\log(2\pi) - n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{x \in \mathcal{X}}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))^2$$

MLE for $x^\star$. We can set the derivative equal to 0 to obtain:

$$0 = \frac{\partial}{\partial x^\star}\log(\ell(s; x^\star, d)) = -\frac{1}{2\sigma^2}\sum_{x \in \mathcal{X}}\frac{\partial}{\partial x^\star}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))^2$$

Which implies

$$\sum_{x \in \mathcal{X}}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))\frac{\partial}{\partial x^\star}\exp(-\frac{\|x - x^\star\|^2}{2h^2}) = 0$$

$$\sum_{x \in \mathcal{X}}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))\exp(-\frac{\|x - x^\star\|^2}{2h^2})\frac{\partial}{\partial x^\star}\|x - x^\star\|^2 = 0$$

$$\sum_{x \in \mathcal{X}}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))\exp(-\frac{\|x - x^\star\|^2}{2h^2})(x^\star - x) = 0$$

We might imagine a weighted average of x is the MLE for $x^\star$. In other words, let's guess $x^\star = \sum_{x \in \mathcal{X}} s(x)x$. Substituting this in:

$$\sum_{x \in \mathcal{X}}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))\exp(-\frac{\|x - x^\star\|^2}{2h^2})x^\star$$

$$-\sum_{x \in \mathcal{X}}(\exp(-\frac{\|x - x^\star\|^2}{2h^2}) - s(x))\exp(-\frac{\|x - x^\star\|^2}{2h^2})x$$

$$= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{X}} ys(y)(\exp(-\frac{\|x-x^\star\|^2}{2h^2}) - s(x)) \exp(-\frac{\|x-x^\star\|^2}{2h^2})$$

$$- \sum_{x\in\mathcal{X}} (\exp(-\frac{\|x-x^\star\|^2}{2h^2}) - s(x)) \exp(-\frac{\|x-x^\star\|^2}{2h^2})x$$

$$= \sum_{y\in\mathcal{X}} ys(y) \sum_{x\in\mathcal{X}} (\exp(-\frac{\|x-x^\star\|^2}{2h^2}) - s(x)) \exp(-\frac{\|x-x^\star\|^2}{2h^2})$$

$$- \sum_{y} s(y) \sum_{x\in\mathcal{X}} (\exp(-\frac{\|x-x^\star\|^2}{2h^2}) - s(x)) \exp(-\frac{\|x-x^\star\|^2}{2h^2})x$$

$$= \sum_{y\in\mathcal{X}} \sum_{x\in\mathcal{X}} (y-x)(\exp(-\frac{\|x-x^\star\|^2}{h^2})s(y) - \exp(-\frac{\|x-x^\star\|^2}{2h^2})s(x)s(y))$$