Contents

1 Model

$$p(\theta) = \frac{1}{2\pi}$$

$$P(r_i|\theta) = \frac{(f_i(\theta)\tau)^{r_i}}{r_i!} e^{-f_i(\theta)\tau} \tag{1}$$

$$P(r_1, \dots r_M|\theta) = \prod_{j=1}^{M} \frac{(f_j(\theta)\tau)^{r_j}}{r_j!} e^{-f_j(\theta)\tau} \tag{2}$$

where $\theta \in (0, 2\pi)$, t is the tuning curve (periodic), s is the stimulus, τ is the time for stimulus to be active. The rate function f is obtained by convoultion:

$$f(\theta) = \int t(\theta - y)s(y)dy \tag{3}$$

which specifies the firing rate (number of neurons fired in unit time).

1.1 Problem

$$\begin{gathered} \max_{t_1(\cdot),\cdots,t_M(\cdot)} I(\mathbf{r};\theta) \\ f_- \le f_i(\cdot) \le f_+ \\ \int f_i(\theta)p(\theta)d\theta = const \end{gathered} \tag{4}$$

And since f is the convolution of t and s, this is equivalent to:

$$\begin{gathered} \max_{t_1(\cdot),\cdots,t_M(\cdot)} I(\mathbf{r};\theta) \\ f_- \le t_i(\cdot) \le f_+ \\ \int t_i(\theta)p(\theta)d\theta = const \end{gathered} \tag{5}$$

1.2 Discretization and assumption

$\theta = (\theta_1, \ldots, \theta_M)$, $f = (f_1, \ldots, f_M)$. Assume the tuning curves (also the rate curves) is **rotationally invariant**, i.e. different f_i's have the same shape but difference in translations (rotations in $(0, 2\pi)$):

$$f_i(\theta_j) = f_0(\theta_j - \theta_i) = f_{j-i} \tag{6}$$

From Rotational Invariance Assumption we get:

$$\begin{aligned} I(\mathbf{r};\theta) &= D_{KL}(p(\mathbf{r},\theta)||p(\mathbf{r})p(\theta)) \\ &= \int_\theta \int_{\mathbf{r}} p(\mathbf{r}|\theta)p(\theta) \ln\left(\frac{p(\mathbf{r}|\theta)}{p(\mathbf{r})}\right) d\mathbf{r}d\theta \qquad (7) \\ &= \frac{1}{M}\sum_{i=1}^{M} D_{KL}\left(p(\mathbf{r}|\theta=\theta_i)\|p(\mathbf{r})\right) \\ &= D_{KL}\left(p(\mathbf{r}|\theta=0)\|p(\mathbf{r})\right) \\ &= -E_{\mathbf{r}|\theta=0}\left[\ln\left(\frac{p(\mathbf{r})}{p(\mathbf{r}|\theta=0)}\right)\right] \\ &= -E_{\mathbf{r}|\theta=0}\ln\left(S(\mathbf{r})\right) \qquad (8) \end{aligned}$$

where

$$P(\mathbf{r}|\theta=0) = \prod_{j=1}^{M} \frac{(f_j\tau)^{r_j}}{r_j!} e^{-(f_j\tau)}$$

$$S(\mathbf{r}) = \frac{1}{M}\sum_{k=1}^{M}\prod_{j=1}^{M}\left(\frac{(\tau f_{j-k})^{r_j}}{(\tau f_j)^{r_j}}\right) = \frac{1}{M}\sum_{k=1}^{M}\prod_{j=1}^{M} f_j{}^{r_{j+k}-r_j}$$

Therefore the discretized version of Problem:

$$\begin{gathered} \max_{t_1,\cdots,t_M}\left[-E_{\mathbf{r}|\theta=0}\ln\left(S(\mathbf{r})\right)\right] \\ f_- \le t_i \le f_+ \\ \textstyle\sum_{i=1}^{M} t_i = const \end{gathered} \tag{9}$$

Here t_i denotes the value of tuning curve t (t_0) at θ_i, not the tuning curve function as in (4), (5).

1.3 Derivatives

By writing $I(\mathbf{r};\theta) = -\sum_{\mathbf{r}} P(\mathbf{r}|\theta=0)\ln(S(\mathbf{r}))$ we get

$$\frac{\partial I}{\partial f_i} = \tau \cdot E_{\mathbf{r}|\theta=0}\left[(1 - r_i/f_i)\ln(S)\right]$$

$$\frac{\partial^2 I}{\partial f_i \partial f_j} = \tau^2 \cdot E_{\mathbf{r}|\theta=0}\left[-\left(r_i/f_i - 1\right)\left(r_j/f_j - 1\right)\ln(S) + \frac{1}{2f_i f_j}\frac{\sum_k (r_i - r_{i+k})(r_j - r_{j+k})\prod_l f_l^{r_{l+k}}}{\sum_k \prod_l f_l^{r_{k+l}}} + 1_{\{i=j\}}\cdot\left(r_i/f_i^2\right)\ln(S)\right]$$

And from the convolution relation

$$\frac{\partial I}{\partial t_i} = \sum_j \frac{\partial I}{\partial f_j}\frac{\partial f_j}{\partial t_i} = \sum_j \frac{\partial I}{\partial f_j} s_{j-i}$$

$$\frac{\partial^2 I}{\partial t_i \partial t_j} = \sum_k \sum_l \frac{\partial^2 I}{\partial f_k \partial f_l} s_{k-i} s_{l-j}$$

1.4 Monte Carlo

In computation, we approximate the above expectations by Monte Carlo:

$$\hat{I} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(S(\mathbf{r}^{(n)})\right)$$

$$\widehat{\partial_{f_i} I} = \tau \cdot \frac{1}{N}\sum_{n=1}^{N}\left[\left(1 - r_i^{(n)}/f_i\right)\ln(S(\mathbf{r}^{(n)}))\right]$$

$$\widehat{\partial_{f_i}\partial_{f_j} I} \quad = \quad \tau^2 \cdot \frac{1}{N}\sum_{t=1}^{N}\left[-\left(r_i/f_i - 1\right)\left(r_j/f_j - 1\right)\ln(S(\mathbf{r})) + \frac{1}{2f_i f_j}\sum_k (r_i - r_{i+k})(r_j - r_{j+k})Eexp(k) + 1_{\{i=j\}} \cdot \left(r_i/f_i^2\right)\ln(S(\mathbf{r}))\right]^{(n)}$$

where $\mathbf{r}^{(n)} = \left(r_1^{(n)}, \ldots, r_M^{(n)}\right)$ are sampled from independent Poisson distributions: $r_i^{(n)} \sim Poission(f_i)$.

2 Test Results

Basic Setting for most our cases (may very in different tests):

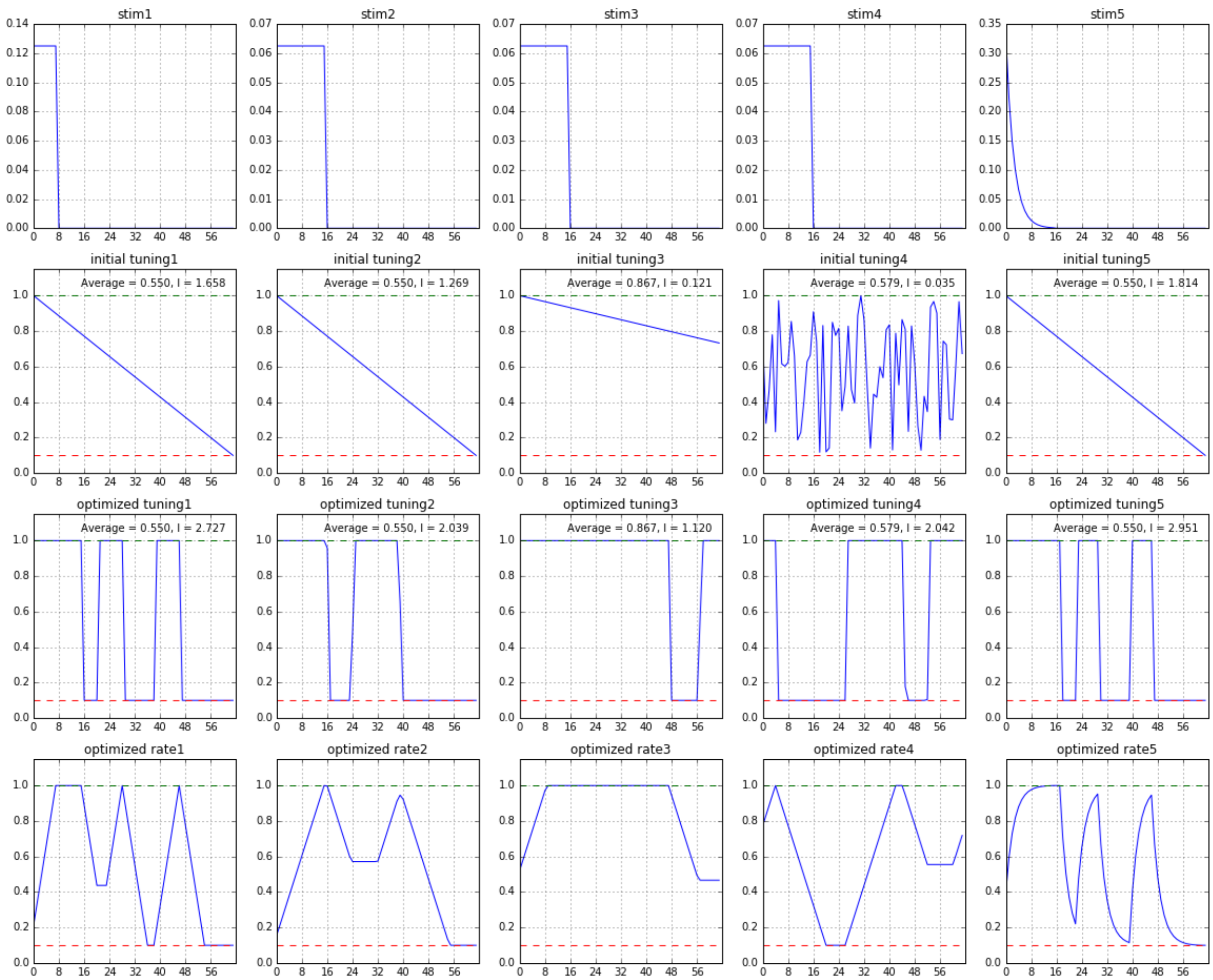
- Constrained Optimization Method: SLSQP (Sequential Least SQuares Programming). It uses the Han–Powell quasi–Newton method with a BFGS update of the B–matrix and an L1–test function in the step–length algorithm. The optimizer uses a slightly modified version of Lawson and Hanson's NNLS nonlinear least-squares solver. Reference: Scipy ,PyOpt
- Language in C++/Python, Package: scipy.optimize.minimize
- $M = numBin = 64$
- $s(x) = \frac{1}{\nu\tau}1_{[0,\nu\tau]}(x)$: rectangular stimulus function with speed ν, time period τ. Note: the integral of stim is always normalized to 1; however the rate function f is multiplied by τ , see equation 1. In some cases we set $s(x) \propto e^{-\alpha x}1_{[0,\nu\tau]}(x)$, also normalized.
- $\nu = 16$, $\tau = 1.0$
- $f_- = 0.1$, $f_+ = 1.0$, $\frac{f_+}{f_-} = 10$.
- Average of tuning curve $\frac{1}{M}\sum t_i = \frac{f_- + f_+}{2} = 0.55$
- Number of iterations in Monte Carlo: 10^5 for I, 10^6 for ∇I.
- Number of iterations in optimization around 120 (the program doesn't end itself but the shape has generally stoped to change)
- Use $\ln(\cdot)$ instead of $\log_2(\cdot)$ in all information/gradient calculations.

First we vary different parameters above and did 5 experiments:

Observe the phenomenon:

- Same parameters and constraints, different initial tuning curves may 'converge' to different maximizers, so the maximizers may be local.
- But the tuning curves all seem to reach f_+, f_- as much as possible, i.e. 'saturates' to the extremes.
- Changing stim shape from rectangle to exponential does not affect the saturation effect.
- Decreasing ν will increase the mutual information I, since we have a smaller window of convolution.

Systematically, we look at the effect of varying each test condition:



stim1
stim2
stim3
stim4
stim5
initial tuning1
Average = 0.550, I = 1.658
initial tuning2
Average = 0.550, I = 1.269
initial tuning3
Average = 0.867, I = 0.121
initial tuning4
Average = 0.579, I = 0.035
initial tuning5
Average = 0.550, I = 1.814
optimized tuning1
Average = 0.550, I = 2.727
optimized tuning2
Average = 0.550, I = 2.039
optimized tuning3
Average = 0.867, I = 1.120
optimized tuning4
Average = 0.579, I = 2.042
optimized tuning5
Average = 0.550, I = 2.951
optimized rate1
optimized rate2
optimized rate3
optimized rate4
optimized rate5

Figure 1:

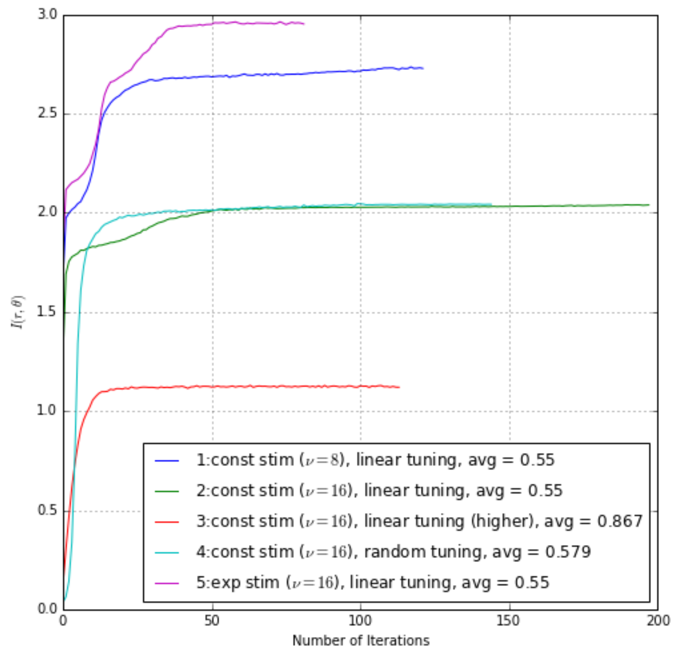


Figure 2:

2.1 Constraints in Optimization

2.1.1 f_+ and f_-

Figure 3.

Same initial tuning curve, different inequality constriants. All tends to saturates to the bounds, but surprisingly $f_+ = 2$ gives higher information (though constraint of $f_+ = 4$ are more loose, but it converges to a local maximum where I is lower.)

2.1.2 Average Firing Rate

Figure 4, Figure 5.

$\frac{f_+ + f_-}{2}$ might be the max, but not symmetric.

2.2 Stimulus Width

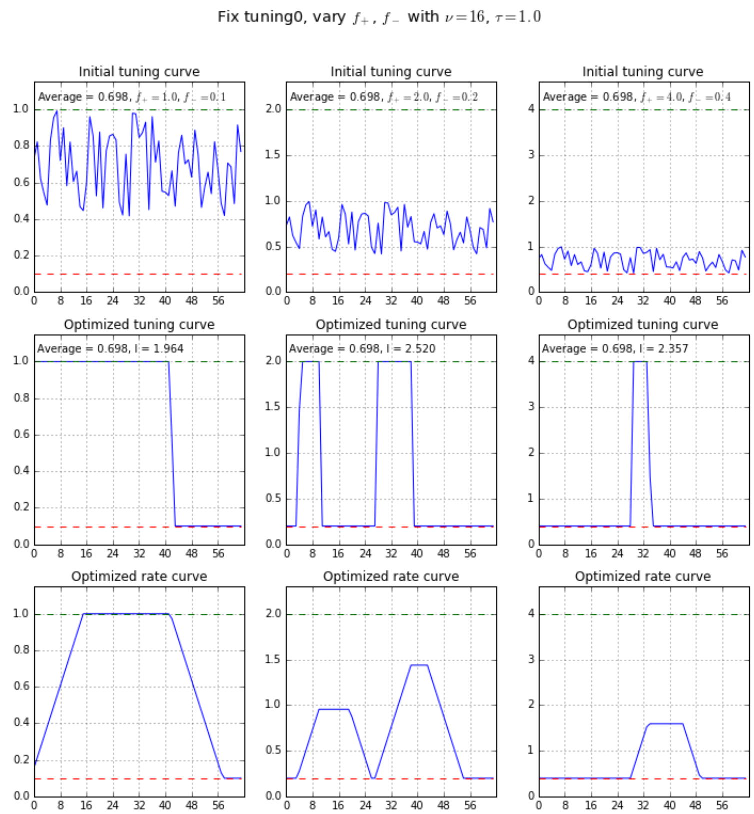
2.2.1 The effect of ν

First we just calculated the mutual information for different ν with the following tuning and stimulus in Figure 6, the mutual information shown in Figure 7.

Then we did optimization with different ν: Figure 8.

2.2.2 The effect of τ

We take $\nu = 8$, $f_- = 0.1$, $f_+ = 1.0$. Note that the stimulus $s = \frac{1}{\nu\tau}1_{[0,\nu\tau)}$ (integral normalized to 1), and the poisson distribution is related to τ: $P(\mathbf{r}|\theta = 0) = \prod_{j=1}^{M} \frac{(f_j\tau)^{r_j}}{r_j!} e^{-(f_j\tau)}$.



Fix tuning0, vary f_+, f_- with $\nu=16$, $\tau=1.0$
Initial tuning curve
Average = 0.698, $f_+=1.0$, $f_-=0.1$
Average = 0.698, $f_+=2.0$, $f_-=0.2$
Average = 0.698, $f_+=4.0$, $f_-=0.4$
Optimized tuning curve
Average = 0.698, I = 1.964
Average = 0.698, I = 2.520
Average = 0.698, I = 2.357
Optimized rate curve

Figure 3:

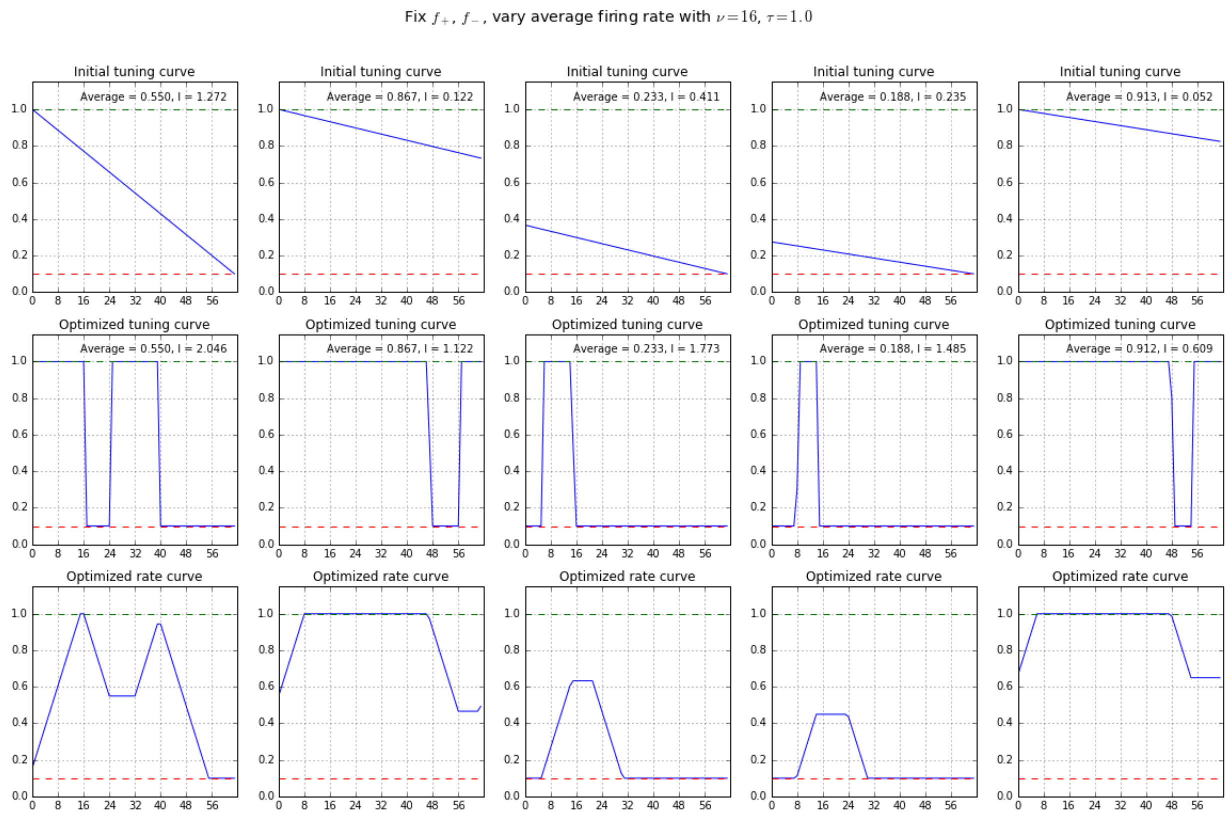

Fix f_+, f_-, vary average firing rate with $\nu = 16$, $\tau = 1.0$
Initial tuning curve
Average = 0.550, I = 1.272
Average = 0.867, I = 0.122
Average = 0.233, I = 0.411
Average = 0.188, I = 0.235
Average = 0.913, I = 0.052
Optimized tuning curve
Average = 0.550, I = 2.046
Average = 0.867, I = 1.122
Average = 0.233, I = 1.773
Average = 0.188, I = 1.485
Average = 0.912, I = 0.609
Optimized rate curve

Figure 4:

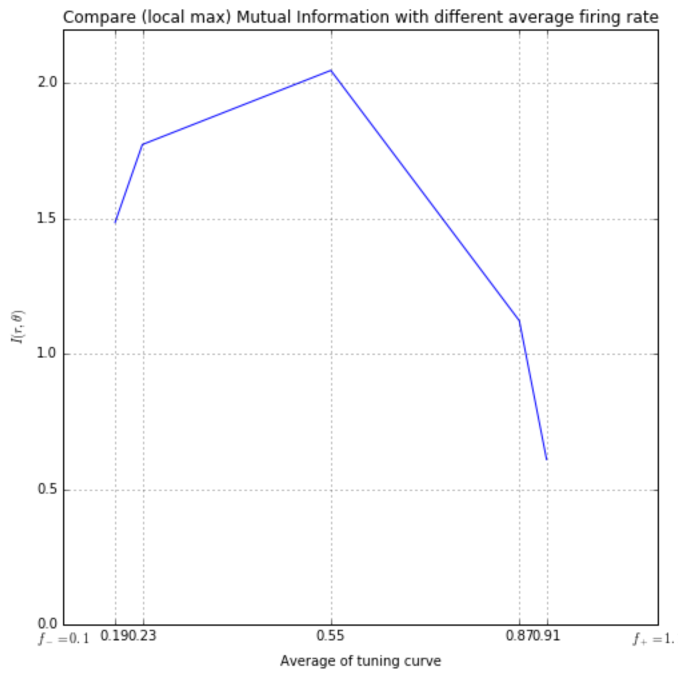


Figure 5:

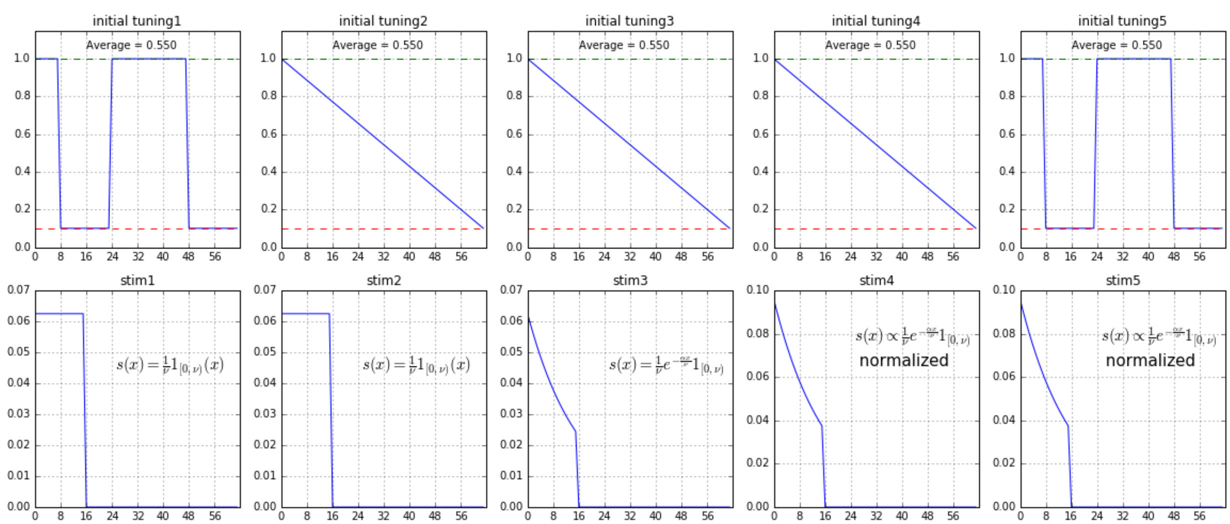


Figure 6:

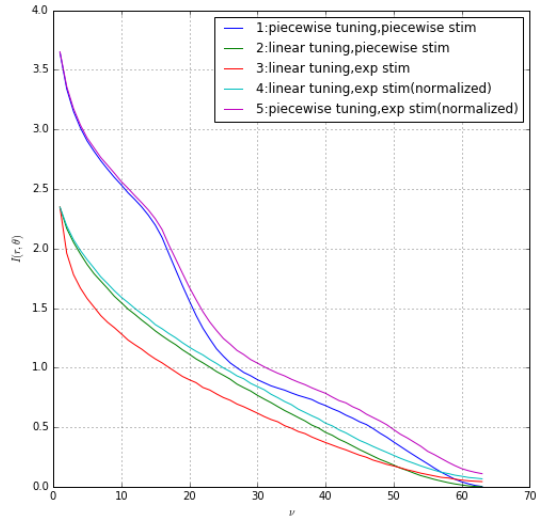


Figure 7:

What we want to investigate is the trade-off between improved evidence accumulation (larger spike count average) and worsened position certainty (larger convolution window) which is entailed by increasing τ. We expect that there is an optimal τ yielding the maximum mutual information.

Figure 9 shows max at $\tau = 0.875$, i.e. $\nu\tau = 7$.

More figures see 'data2-2' directory.

2.3 Stimulus function shape (to be examined)

2.4 Initial Conditions of Tuning curve (See Figure 1)

2.5 Random choice of Saturation

Not all curves which takes either f_+or f_- are local max. See Figure 11: randomly take different proportions of points to be f_+ (other points f_-).

3 Piecewise Constant Tuning Curves

For $M = 64$, $\nu = 16$, $\tau = 1.0$, average firing rate $= \frac{f_+ + f_-}{2}$, assume our local maximizers are of the form

$$f_+ \cdot 1_{[0,d_+)} + f_- \cdot 1_{[d_+,d_++d_-)} + f_+ \cdot 1_{[d_++d_-,\frac{M}{2}+d_-)} + f_- \cdot 1_{[\frac{M}{2}+d_-,M)}$$

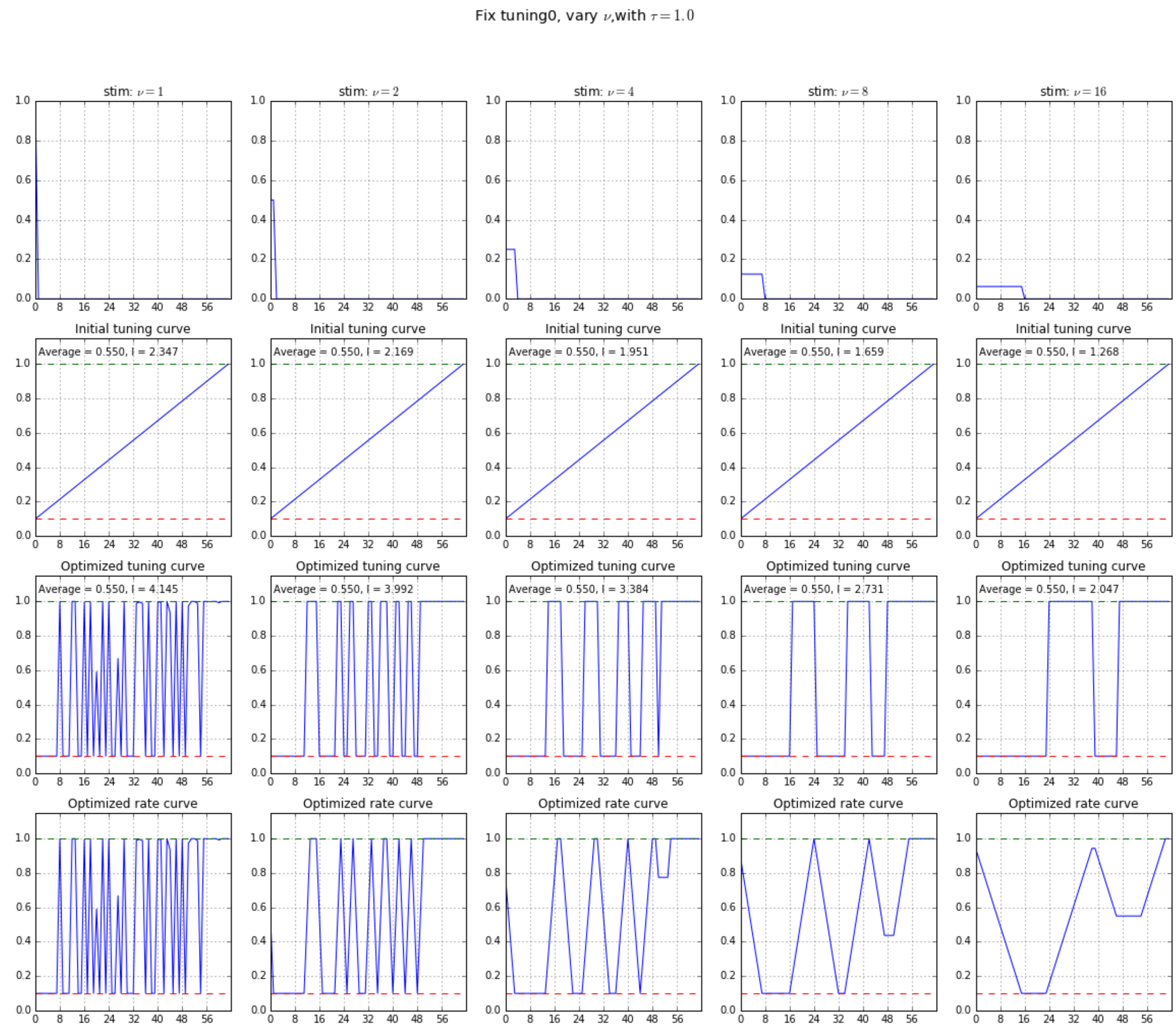
i.e. two plateaus with width sum up to 32.

1. Changing d_+, d_- we get the following diamond shape: Figure 12, optimized when $d_+ = 8$, $d_- = 16$.

2. When $f_+ = 16$, $f_- = 1.6$ we have Figure 13. By looking at rate curves in Figure 14 (of $f_+ = 1$) we can explain partially why we have larger plateaus with lower f_-, f_+: since neurons' firing rate is too low, we need to count spikes from more neurons.

3. Not necessarily the same shape when changing M: see the diamond with $M = 32$ in Figure 15.

4. Still need explanation why the gradient fails at $d_+ = d_- = 16$ (knowing that it is a sharpe local minimum because of periodicity with period $\frac{M}{2}$, losing information).



Fix tuning0, vary ν,with $\tau = 1.0$
stim: $\nu = 1$
stim: $\nu = 2$
stim: $\nu = 4$
stim: $\nu = 8$
stim: $\nu = 16$
Initial tuning curve
Average = 0.550, I = 2.347
Average = 0.550, I = 2.169
Average = 0.550, I = 1.951
Average = 0.550, I = 1.659
Average = 0.550, I = 1.268
Optimized tuning curve
Average = 0.550, I = 4.145
Average = 0.550, I = 3.992
Average = 0.550, I = 3.384
Average = 0.550, I = 2.731
Average = 0.550, I = 2.047
Optimized rate curve

Figure 8:

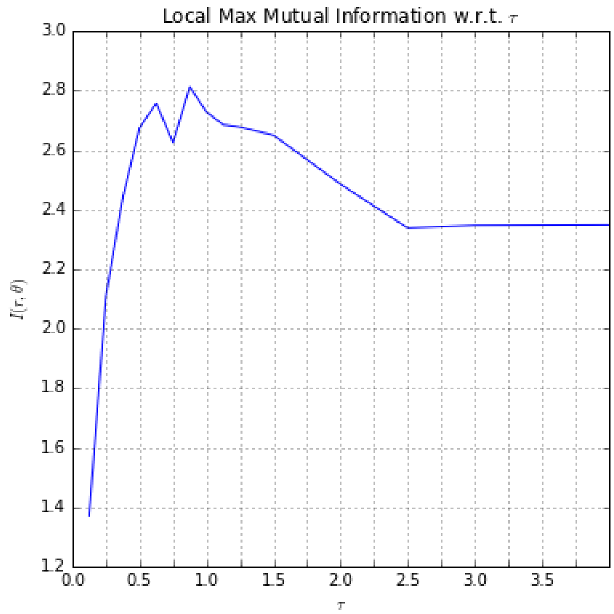


Figure 9:

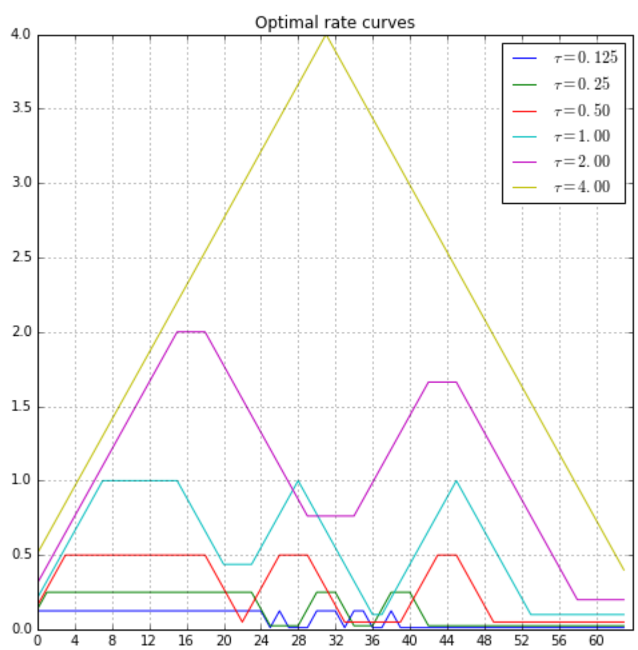


Figure 10:

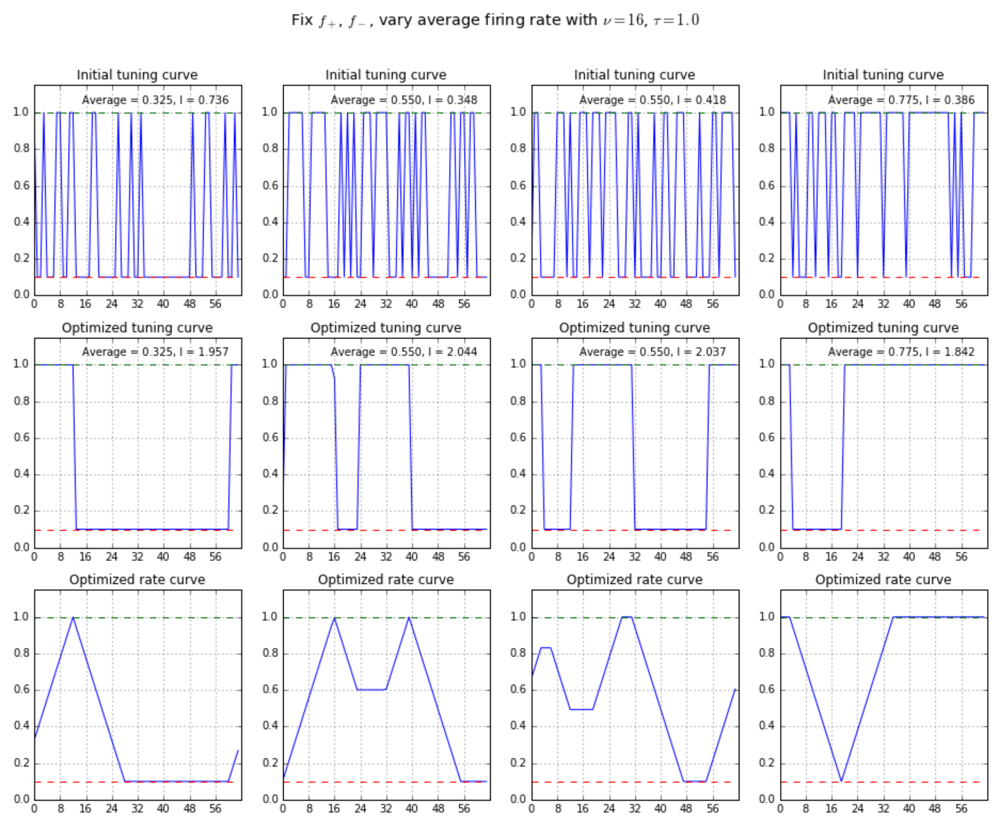

Fix f_+, f_-, vary average firing rate with $\nu = 16$, $\tau = 1.0$
Initial tuning curve
Average = 0.325, I = 0.736
Initial tuning curve
Average = 0.550, I = 0.348
Initial tuning curve
Average = 0.550, I = 0.418
Initial tuning curve
Average = 0.775, I = 0.386
Optimized tuning curve
Average = 0.325, I = 1.957
Optimized tuning curve
Average = 0.550, I = 2.044
Optimized tuning curve
Average = 0.550, I = 2.037
Optimized tuning curve
Average = 0.775, I = 1.842
Optimized rate curve
Optimized rate curve
Optimized rate curve
Optimized rate curve

Figure 11:

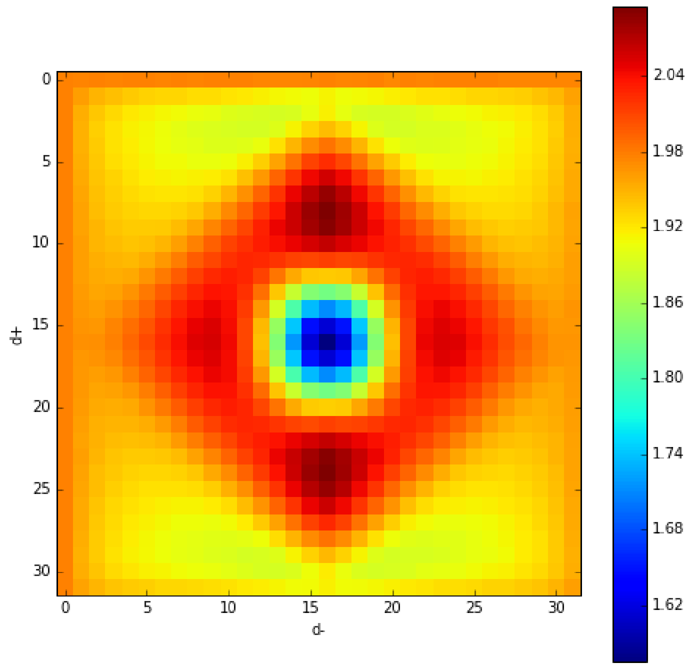


Figure 12:

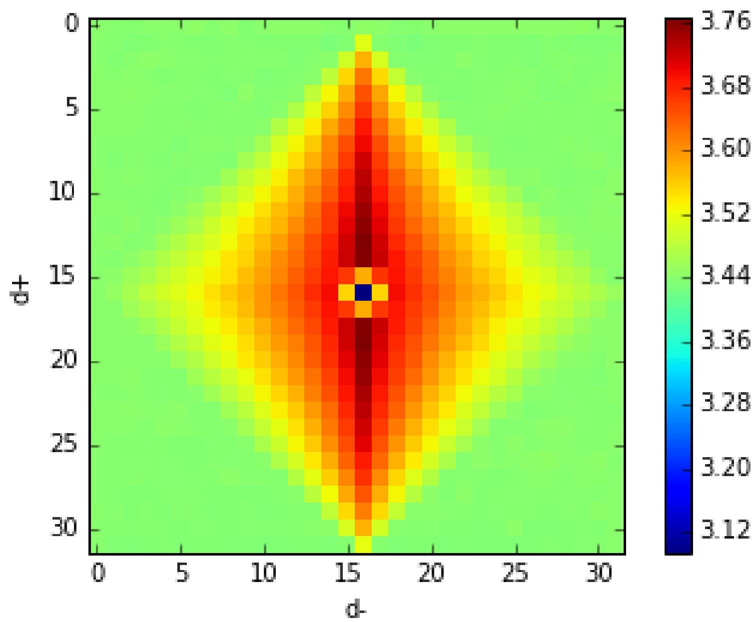


Figure 13:

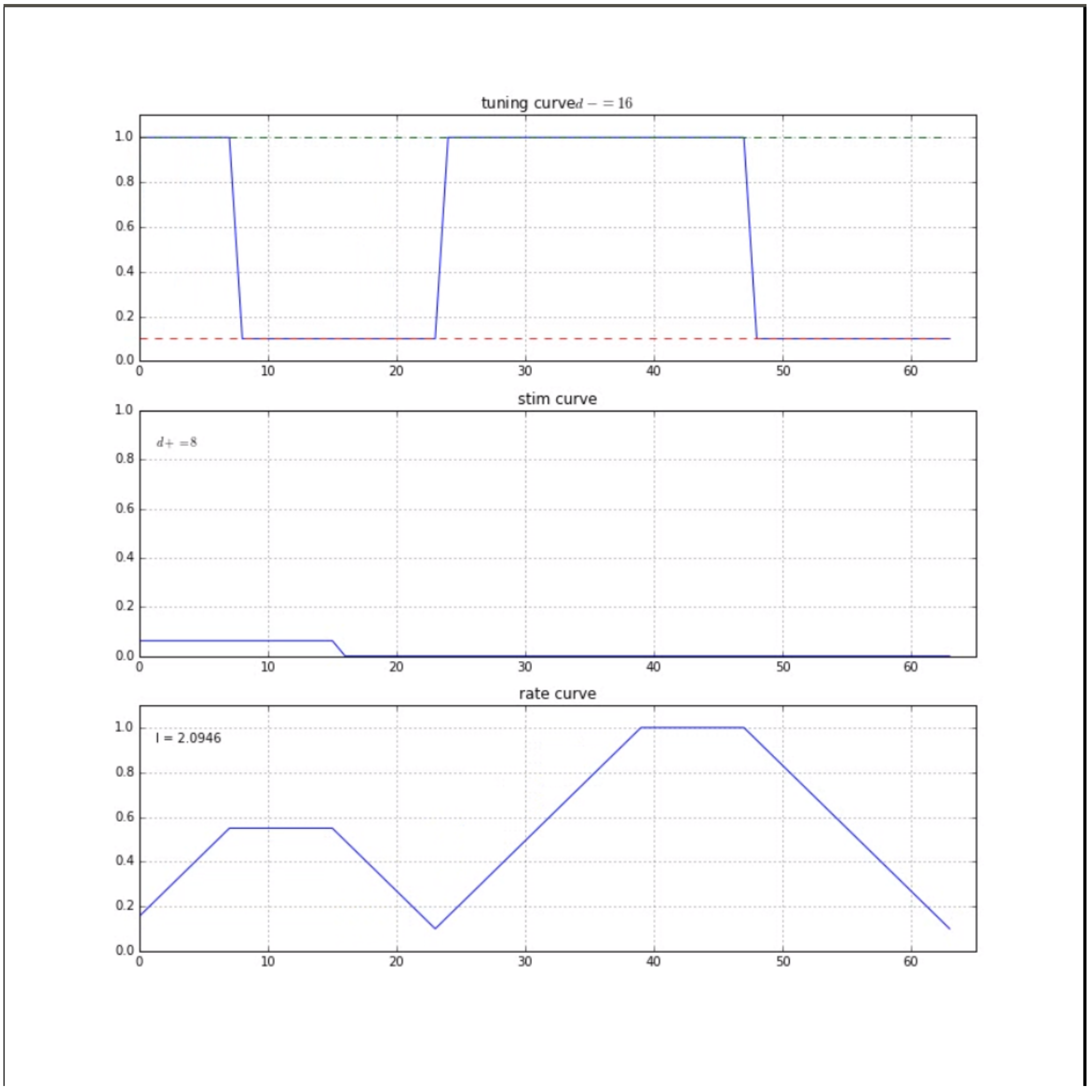


Figure 14:

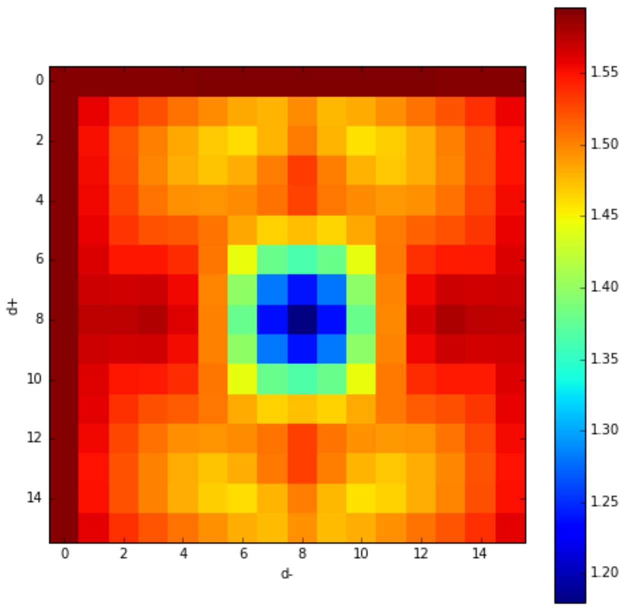


Figure 15:

4 About Karush–Kuhn–Tucker conditions

For convenience of reading we denote the tuning curve t as f in this section.

The Lagrangian

$$L(f;\alpha^+,\alpha^-,\mu) = I(f) + \alpha^+(f - f_+) + \alpha^-(f_- - f) + \mu(\frac{1}{M}\sum f - c)$$

where c is the average of initial tuning curve, $f - f_+ \leq 0$, $f_- - f \leq 0$.

The constraints of the problem (9) are linear, so the KKT conditions are necessary conditions (although may not be sufficient since $I(f)$ is not convex):

if $f^* = (f_1^*, ..., f_M^*)$ is a local maximizer, then there exists $\alpha^+ = (\alpha_1^+, ..., \alpha_M^+)$, $\alpha^- = (\alpha_1^-, ..., \alpha_M^-)$ and constant μ, such that

1. $\nabla_{f*} I(s) = \alpha^+(s) - \alpha^-(s) + \mu, \forall s$
2. $f_- \leq f^* \leq f_+$
3. $\alpha^+ \geq 0, \alpha^- \geq 0$
4. $\alpha^+(s)(f^*(s) - f_+) = 0$, $\alpha^-(s)(f_- - f^*(s)) = 0$, i.e. α^+ is supported on $\{f = f_+\}$, α^- is supported on $\{f = f_-\}$.

Hence we have:

- $\nabla_{f*} I(s) = \mu$, $\forall s$ such that $f^*(s) \neq f_+$, $f^*(s) \neq f_-$
- $\nabla_{f*} I(s) = \alpha^+(s) + \mu \geq \mu$, $\forall s$ such that $f^*(s) = f_+$
- $\nabla_{f*} I(s) = -\alpha^-(s) + \mu \leq \mu$, $\forall s$ such that $f^*(s) = f_-$

So

$$\begin{aligned} \nabla_{f*} I(s_-) &\leq \mu &\leq \nabla_{f*} I(s_+) \\ \max_{s_-} \nabla_{f*} I(s_-) &\leq \mu &\leq \min_{s_+} \nabla_{f*} I(s_+) \end{aligned} \tag{10}$$

Therefore as long as f has the property that $\max \nabla_f I(s_-) \leq \min \nabla_f I(s_+)$ (where $s_- \in \{f = f_-\}$, $s_+ \in \{f = f_+\}$), it is possible to find μ such that (10) holds, and satisfies the KKT conditions.

See the animations in 'mydata2' directory which shows the gradient and μ in experiments.