Math Modeling, Week 13

1. A kernel (or similarity function) k is said to be *positive-definite* if it satisfies the following condition. For any set of stimuli $x_1, \ldots, x_n$, we can define a matrix K by $K_{ij} = k(x_i, x_j)$. The kernel is positive-definite if K is always positive-semidefinite, meaning $c^{\mathrm{T}}Kc \geq 0$ for any vector $c \in \mathbb{R}^n$.

Prove that a kernel must be positive definite in order for it to be equivalent to the inner product for some feature representation. That is, if each stimulus x_i can be written as a vector of feature values x_{il} with $k(x_i, x_j) = \sum_l x_{il}x_{jl}$, then the matrix K must be positive-semidefinite.

Hint: If there are a finite number of features (say, m features), then we can think of $x_1, \ldots, x_n$ as defining a $n \times m$ stimulus-by-feature matrix X, with $X_{il} = x_{il}$. Notice that $K = XX^{\mathrm{T}}$, and also that $c^{\mathrm{T}}X = \sum_i c_i x_i$ is a feature vector defined by a linear combination of the x_is. Prove that $c^{\mathrm{T}}Kc$ must be nonnegative by writing it in terms of $c^{\mathrm{T}}X$. Extra points if you can extend this proof to the case of a (countably) infinite set of features.

2. Regularization refers to a variety of methods in machine learning whereby overfitting is reduced by encouraging simpler solutions. This is achieved by incorporating soft constraints into the objective function. That is, instead of optimizing the model's fit to the training data, we optimize that fit plus an additional term that penalizes complex solutions.

One of the most common instances of regularization is ridge regression, which extends ordinary least-squares (OLS) linear regression with a penalty term based on the sum of squared weights:

$$L(\beta) = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 + \lambda \sum_{j=1}^{m} \beta_j^2.$$

The first sum in L is the normal sum-squared-error term from OLS regression, where $\hat{y}_i = \sum_j x_{ij}\beta_j$ (with i indexing the n cases and j indexing the m predictors). The second sum is called an L2 penalty term (essentially because of the exponent in β^2), and its contribution is scaled by the penalty parameter λ. This penalty term encourages smaller weight vectors that are more evenly distributed across the cues.

As in OLS regression, the goal is to find the weight vector $\hat{\beta}$ that minimizes L. If we write everything in matrix notation, then the loss function becomes

$$L(\beta) = (X\beta - Y)^{\mathrm{T}}(X\beta - Y) + \lambda\beta^{\mathrm{T}}\beta$$

(where X is a $n \times m$ matrix of cases by predictors, Y is a $n \times 1$ column vector of outcomes, and β is a $m \times 1$ column vector of weights). Setting the derivative $\frac{\mathrm{d}L}{\mathrm{d}\beta}$ to zero leads to the solution

$$\hat{\beta} = (X^{\mathrm{T}}X + \lambda I)^{-1}X^{\mathrm{T}}Y.$$

The term ridge regression comes from the λI term, which adds a ridge along the diagonal of the covariance matrix. Notice that when $\lambda = 0$, $\hat{\beta}$ reduces to the OLS solution. As λ grows larger, $\hat{\beta}$ generally shrinks toward zero.

(a) Implement ridge regression on this dataset. Optimize the penalty parameter using cross-validation, by plotting the root mean squared error on the test set as a function of λ. Try it for different sizes of the training set and compare your results. See here for a guide on how to write the code.

(b) Ridge regression has a Bayesian interpretation, where the weights are generated as $\beta \sim \mathcal{N}(0, \sigma_\eta^2 I)$ and the outcomes are generated as $y_i \sim \mathcal{N}(x_i\beta, \sigma_\varepsilon^2)$. Show that posterior on β is centered on $\hat{\beta}$ as defined above. What is λ in terms of σ_η^2 and σ_ε^2?

(c) Use ridge regression to derive a regularized variant of Rescorla-Wagner, as follows. Recall that the RW learning rule is based on gradient descent on squared prediction error. Add an L2 penalty term as above, and apply gradient descent to derive a new learning rule.