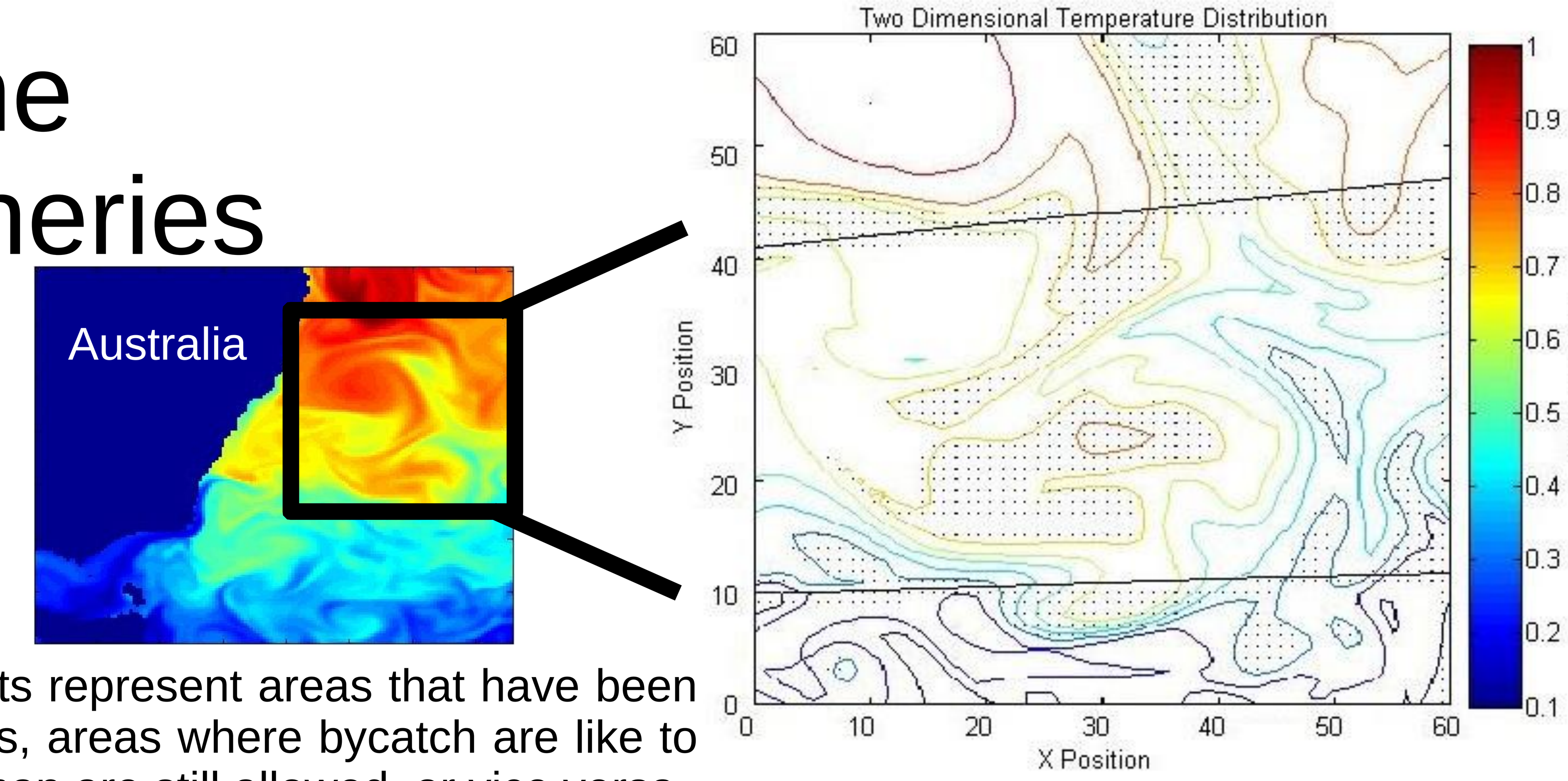
2011

Regularization Schemes for the Real Time Spatial Management of Pelagic Longline Fisheries

Kyle Egerdal
kegerdal@pugetsound.edu

Regularization Schemes for the Real Time Spatial Management of Pelagic Longline Fisheries

Kyle Egerdal, Adviser: Carl Toews
The University of Puget Sound
Summer 2011



Sample boundary allocations for the Coral and Tasman Sea, located just east of Australia. Dots represent areas that have been misclassified – that is, areas where bycatch are like to roam but that fisherman are still allowed, or vice versa.

Background

Bycatch from commercial fisheries often includes critically endangered species. Fisheries managers can ameliorate this problem by channeling fishing effort into regions where threatened species are scarce.

In the case of Bluefin tuna, local habitat preferences correlate strongly with ocean depth and temperature profiles. Coupling satellite temperature data with a regional and seasonal depth-temperature model allows managers to make near-real time spatial estimates of Bluefin prevalence. These estimates can be used to allocate fishing zones. A key challenge is to automate this process in a way that yields intelligible boundaries while balancing economic and environmental costs.

An Optimization Problem

Consider a regular square lattice, each node associated with a "tuna suitability" score between 0 and 1. The object is to draw 2 lines, Λ_1 and Λ_2, each with at most *m* segments, dividing the lattice into zones A, B, and C, where A = open fishing, B = limited fishing (permit based), and C = closed. Following [1], the (*i,j*)'th lattice point is assigned a misclassification penalty $p(s_{ij})$ that depends on the zone of the lattice point and its suitability score s_{ij}:

Suitability	Zone A	Zone B	Zone C
$0 \le s_{ij} < 1/3$	$p(s_{ij})=0$	$p(s_{ij})=U_1$	$p(s_{ij})=U_2$
$1/3 \le s_{ij} \le 2/3$	$p(s_{ij})=L_1$	$p(s_{ij})=0$	$p(s_{ij})=U_1$
$2/3 < s_{ij} \le 1$	$p(s_{ij})=L_2$	$p(s_{ij})=L_1$	$p(s_{ij})=0$

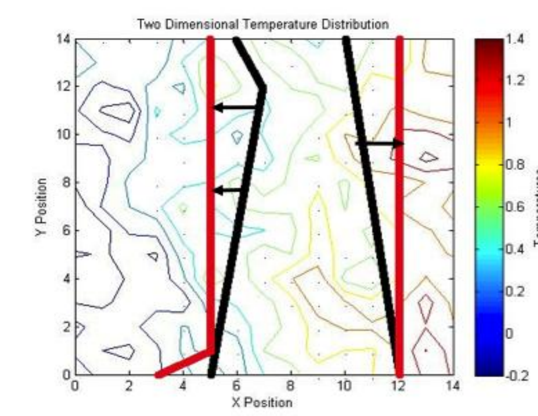
Note that higher values of U support "fish-friendly" policies, while higher values of L support "fisherman-friendly" policies. The total cost associated with boundary lines Λ_1 and Λ_2 is given by:

$$\Gamma(\Lambda_1, \Lambda_2) = \sum_{i,j=1}^{n} p(s_{ij})$$

For a given constellation of suitability scores, the optimization problem is to minimize this cost function.

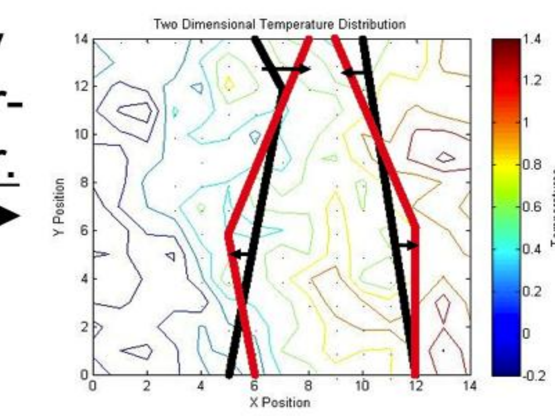
Regularization

Zoning boundaries change over time in response to changing temperatures. Since large boundary movements are expensive to fishermen, we can discourage them by adding another penalty term to the cost::



Movement penalty not considered. Boundaries have moved quite far.

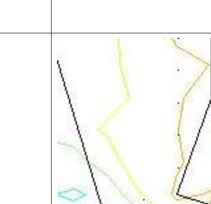
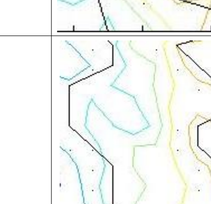
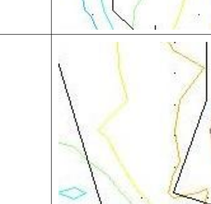
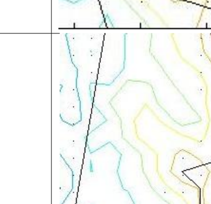
Movement penalty considered. Boundaries move negligibly.



$$\Gamma(\Lambda_1^t, \Lambda_2^t) = \sum_{i,j=1}^{n} p(s_{ij}^t) + \gamma \sum_{i=1}^{2} \|\Lambda_i^t - \Lambda_i^{t-1}\|$$

Methods of Optimization

Both the regularized and the unregularized problems involve combinatorial optimization. Since this tends to be computationally intensive, solving a "relaxed" problem is often necessary in practice [2]. The following chart compares several novel relaxation methods. The best method will be an effective mediation between computational complexity and managerial efficiency.

Method	Description	Complexity where n is one side of a square ocean, and m is the number of segments of each boundary	Estimated Time for a 20x20 Lattice	Estimated Time for a 100x100 Lattice	Sample Result
Complete Search	Calculates every possibility and chooses the one with the lowest penalty.	$\binom{n-2}{m-1} \cdot n^{m+1}$	1 segment: 2 minutes 2 segments: 18 minutes 3 segments: 3 hours	1 segment: 5 hours 2 segments: 2 days 3 segments: 17 days	
"Curves"	Optimizes once for every y-coordinate, but does not combine y-coordinates. Messy.	n^2	2 seconds	30 seconds	
"Segments"	Checks only segment combinations with endpoints on "Curves."	$n^2 + n(m-1)$	1 segment: 10 seconds 2 segments: 2 minutes 3 segments: 15 minutes	1 segment: 30 minutes 2 segments: 5 hours 3 segments: 2 days	
"Greedy"	Starts at the bottom and takes the segment with the least per-height penalty until it reaches the top.	Average: $n + n^2(n-1)$	5 seconds	1 hour	
Gradient Dissent	Disregarding any penalty during calculation, moves from the bottom along the gradient of the desired temperature until it reaches the top.	Dependent on complexity of the distribution	0-5 seconds	0-5 seconds	

Future Work

Questions we would like to consider include:
- How disastrously wrong can a very simple line allocation go?
- What happens if we penalize the number of line segments?
- Should we penalize lines for being "unnecessarily" long or short?

REFERENCES:
1. "Near real-time spatial management based on habitat predictions for a longline bycatch species," by A. Hobday and K. Hartmann. *Fisheries Management and Ecology*, 2006.
2. Numerical Optimization, by J. Nocedal and S. Wright. Springer, 2006.