

Policy Iteration Report

Question 3 – Write a Policy Iteration agent in PolicyIterationAgent.java by implementing the initRandomPolicy(), evaluatePolicy(), improvePolicy() & train() methods. The evaluatePolicy() method should evaluate the current policy (see your lecture notes), specified in the curPolicy field (which your initRandomPolicy() initialized). The current values for the current policy should be stored in the provided policyValues map. The improvePolicy() method performs the Policy improvement step, and updates curPolicy. **(6 points)**

Answer 3 –

Summary for Policy Iteration Agent Implementation

1) Functions Modified:

- initRandomPolicy()
- evaluatePolicy()
- improvePolicy()
- train()

2) Detailed Implementation Analysis:

(a) initRandomPolicy() Method

Purpose:

- Initialize a random policy for all valid game states
- Iterates through all states in the policy values
- For each state, selects a random move from the possible moves
- Ensures each state has a valid move assigned randomly

(b) evaluatePolicy() Method

Purpose:

- Evaluate the current policy's value function
- Iteratively updates state values until convergence
- Uses the Bellman equation to compute expected values
- Checks for convergence by comparing old and new state values
- Stops when the maximum change is less than the delta threshold

(c) improvePolicy() Method

Purpose:

- Improve the current policy by finding better moves
- Looks for moves that maximize expected value for each state
- Uses one-step look-ahead to find the best move
- Returns true if the policy was improved, false otherwise

(d) train() Method

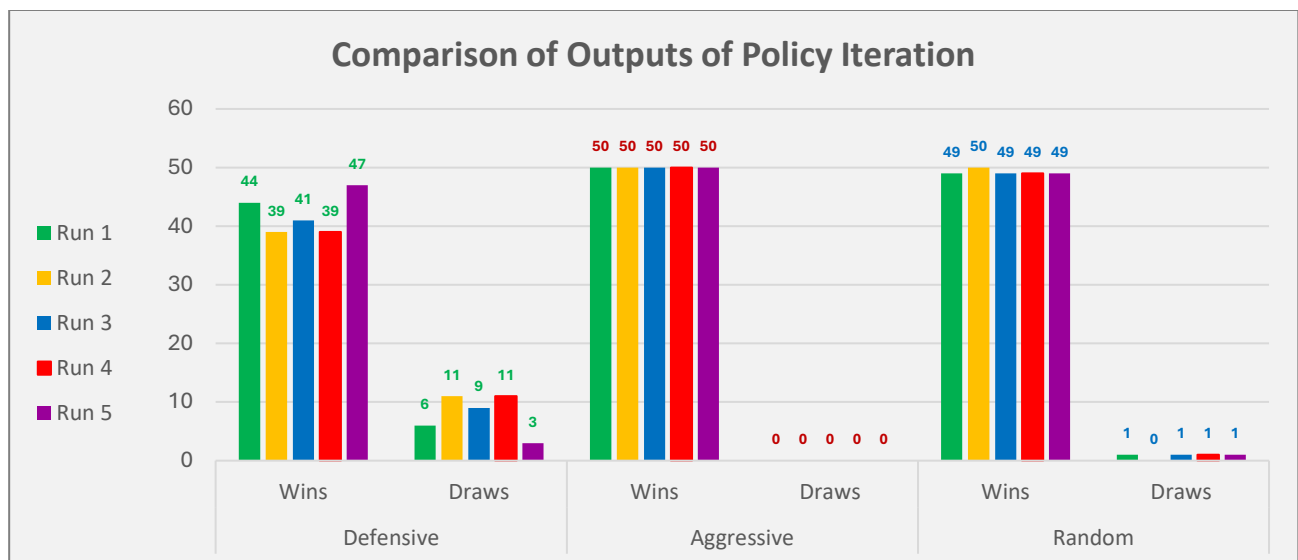
Purpose:

- Train the agent using Policy Iteration algorithm
- Alternates between policy evaluation and policy improvement
- Continues until the policy stabilizes (no further improvements)
- Sets the final policy for the agent

Question 4 – Test your Value Iteration Agent against each of the provided agents 50 times and report on the results – how many games they won, lost & drew against each of the other rule-based agents. The rule-based agents are random, aggressive, defensive. **(1 point)**

Answer 4 –

Iteration/ Agent	Defensive			Aggressive			Random		
	Wins	Losses	Draws	Wins	Losses	Draws	Wins	Losses	Draws
Run 1	44	0	6	50	0	0	49	0	1
Run 2	39	0	11	50	0	0	50	0	0
Run 3	41	0	9	50	0	0	49	0	1
Run 4	39	0	11	50	0	0	49	0	1
Run 5	47	0	3	50	0	0	49	0	1



The program was run for a total of 250 iterations split into 5 runs of 50 iterations each. As we can see from the tabular column, we obtain the best result during iteration 5 of the Value-Iteration Agent.

```

public class ValueIterationAgent implements ValueIterationAgentInterface {
    // 初始化
    public ValueIterationAgent(int n, int m, int k, int l, int r, int b, int g, int o, int p, int q, int s, int t, int u, int v, int w, int x, int y, int z, int aa, int ab, int ac, int ad, int ae, int af, int ag, int ah, int ai, int aj, int ak, int al, int am, int an, int ao, int ap, int aq, int ar, int as, int at, int au, int av, int aw, int ax, int ay, int az, int ba, int bb, int bc, int bd, int be, int bf, int bg, int bh, int bi, int bj, int bk, int bl, int bm, int bn, int bo, int bp, int bq, int br, int bs, int bt, int bu, int bv, int bw, int bx, int by, int bz, int ca, int cb, int cc, int cd, int ce, int cf, int cg, int ch, int ci, int cj, int ck, int cl, int cm, int cn, int co, int cp, int cq, int cr, int cs, int ct, int cu, int cv, int cw, int cx, int cy, int cz, int da, int db, int dc, int dd, int de, int df, int dg, int dh, int di, int dj, int dk, int dl, int dm, int dn, int do, int dp, int dq, int dr, int ds, int dt, int du, int dv, int dw, int dx, int dy, int dz, int ea, int eb, int ec, int ed, int ee, int ef, int eg, int eh, int ei, int ej, int ek, int el, int em, int en, int eo, int ep, int eq, int er, int es, int et, int eu, int ev, int ew, int ex, int ey, int ez, int fa, int fb, int fc, int fd, int fe, int ff, int fg, int fh, int fi, int fj, int fk, int fl, int fm, int fn, int fo, int fp, int fq, int fr, int fs, int ft, int fu, int fv, int fw, int fx, int fy, int fz, int ga, int gb, int gc, int gd, int ge, int gf, int gg, int gh, int gi, int gj, int gk, int gl, int gm, int gn, int go, int gp, int gq, int gr, int gs, int gt, int gu, int gv, int gw, int gx, int gy, int gz, int ha, int hb, int hc, int hd, int he, int hf, int hg, int hh, int hi, int hj, int hk, int hl, int hm, int hn, int ho, int hp, int hq, int hr, int hs, int ht, int hu, int hv, int hw, int hx, int hy, int hz, int ia, int ib, int ic, int id, int ie, int if, int ig, int ih, int ii, int ij, int ik, int il, int im, int in, int io, int ip, int iq, int ir, int is, int it, int iu, int iv, int iw, int ix, int iy, int iz, int ja, int jb, int jc, int jd, int je, int jf, int jg, int jh, int ji, int jj, int jk, int jl, int jm, int jn, int jo, int jp, int jq, int jr, int js, int jt, int ju, int jv, int jw, int jx, int jy, int jz, int ka, int kb, int kc, int kd, int ke, int kf, int kg, int kh, int ki, int kj, int kk, int kl, int km, int kn, int ko, int kp, int kq, int kr, int ks, int kt, int ku, int kv, int kw, int kx, int ky, int kz, int la, int lb, int lc, int ld, int le, int lf, int lg, int lh, int li, int lj, int lk, int ll, int lm, int ln, int lo, int lp, int lq, int lr, int ls, int lt, int lu, int lv, int lw, int lx, int ly, int lz, int ma, int mb, int mc, int md, int me, int mf, int mg, int mh, int mi, int mj, int mk, int ml, int mm, int mn, int mo, int mp, int mq, int mr, int ms, int mt, int mu, int mv, int mw, int mx, int my, int mz, int na, int nb, int nc, int nd, int ne, int nf, int ng, int nh, int ni, int nj, int nk, int nl, int nm, int nn, int no, int np, int nq, int nr, int ns, int nt, int nu, int nv, int nw, int nx, int ny, int nz, int oa, int ob, int oc, int od, int oe, int of, int og, int oh, int oi, int oj, int ok, int ol, int om, int on, int oo, int op, int oq, int or, int os, int ot, int ou, int ov, int ow, int ox, int oy, int oz, int pa, int pb, int pc, int pd, int pe, int pf, int pg, int ph, int pi, int pj, int pk, int pl, int pm, int pn, int po, int pp, int pq, int pr, int ps, int pt, int pu, int pv, int pw, int px, int py, int pz, int qa, int qb, int qc, int qd, int qe, int qf, int qg, int qh, int qi, int qj, int qk, int ql, int qm, int qn, int qo, int qp, int qq, int qr, int qs, int qt, int qu, int qv, int qw, int qx, int qy, int qz, int ra, int rb, int rc, int rd, int re, int rf, int rg, int rh, int ri, int rj, int rk, int rl, int rm, int rn, int ro, int rp, int rq, int rr, int rs, int rt, int ru, int rv, int rw, int rx, int ry, int rz, int sa, int sb, int sc, int sd, int se, int sf, int sg, int sh, int si, int sj, int sk, int sl, int sm, int sn, int so, int sp, int sq, int sr, int ss, int st, int su, int sv, int sw, int sx, int sy, int sz, int ta, int tb, int tc, int td, int te, int tf, int tg, int th, int ti, int tj, int tk, int tl, int tm, int tn, int to, int tp, int tq, int tr, int ts, int tu, int tv, int tw, int tx, int ty, int tz, int ua, int ub, int uc, int ud, int ue, int uf, int ug, int uh, int ui, int uj, int uk, int ul, int um, int un, int uo, int up, int uq, int ur, int us, int ut, int uu, int uv, int uw, int ux, int uy, int uz, int va, int vb, int vc, int vd, int ve, int vf, int vg, int vh, int vi, int vj, int vk, int vl, int vm, int vn, int vo, int vp, int vq, int vr, int vs, int vt, int vu, int vv, int vw, int vx, int vy, int vz, int wa, int wb, int wc, int wd, int we, int wf, int wg, int wh, int wi, int wj, int wk, int wl, int wm, int wn, int wo, int wp, int wq, int wr, int ws, int wt, int wu, int wv, int ww, int wx, int wy, int wz, int xa, int xb, int xc, int xd, int xe, int xf, int xg, int xh, int xi, int xj, int xk, int xl, int xm, int xn, int xo, int xp, int xq, int xr, int xs, int xt, int xu, int xv, int xw, int xx, int xy, int xz, int ya, int yb, int yc, int yd, int ye, int yf, int yg, int yh, int yi, int yj, int yk, int yl, int ym, int yn, int yo, int yp, int yq, int yr, int ys, int yt, int yu, int yv, int yw, int yx, int yy, int yz, int za, int zb, int zc, int zd, int ze, int zf, int zg, int zh, int zi, int zj, int zk, int zl, int zm, int zn, int zo, int zp, int zq, int zr, int zs, int zt, int zu, int zv, int zw, int zx, int zy, int zz);
    // ... (other code)
}

```